OSIMS – Online System Integrity Monitoring System
A Major Qualifying Project
Submitted to the Faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree in Bachelor of Science
in
Electrical and Computer Engineering
By

____
Michael Tasellari

____
Yuchang Zhang

Date:
Project Advisors:

____
Professor Bogdanov, Gene

ABSTRACT

This Major Qualifying Project (MQP) report discusses the research and development of Online System Integrity Monitoring System (OSIMS), a combined hardware and software monitoring network solution for industrial power systems where ensuring the system integrity is critical. OSIMS features infrared imaging in detecting flaws in the system such as, insulator wear-out, machines overload, and internal pressure and heat buildup. OSIMS software is a cross platform desktop application, featuring network and mapping, video streaming, image processing and continuous data plot.

ACKNOWLEDGEMENTS

We would like to thank Techsun Technology Inc. and Beijing Yanshang SINOPEC for their aid during the research phase. The feedback received helped us define the specifications of our product.

Additionally, we would also like to extend our sincerest thanks to our Major Qualifying Project advisor, Professor Gene Bogdanov for his constant support, patience, and guidance throughout the duration of this project.

EXECUTIVE SUMMARY

Electric failures in industrial power systems can lead to large scale destruction and personnel fatalities. The majority of these failures are due to undetected insulation wear-out and abnormal operation of machinery. Particularly in industries such as petroleum, the consequences could lead to explosions or hazardous chemical spills, which means ensuring system integrity is a highly critical task. The source of disastrous failures typically results from undetected flaws in the system. Without knowing the condition of the system, invisible machinery wear-out can turn into potential disasters.

After conducting background research, we discovered that heat is an efficient indicator of flaws and faults in the system. When electric machines encounter excessive stress or internal corruption that leads to overloading, the core will heat up and be reflected on the surface temperature. Similarly, when the insulator material becomes damaged and degraded to cause a leakage current, the surface will also heat up.

We then had a discussion at Techsun Technology Inc., a company that specializes in safety monitoring products for factories and gained more perspective of safety monitoring. We also had meetings with experts from SINOPEC Beijing refinery, from which we further defined the product and derived customer requirements.

We developed the Online System Integrity Monitoring System (OSIMS), a system combining hardware and a desktop application, to provide a safer working environment for factories and help businesses maintain profitability by monitoring and preventing electric failures. The OSIMS hardware features an 80x60 infrared camera resolution and remote Ethernet communication on a private network. The software features multiple corner monitoring, real-time system diagnostic plots, media plus raw data storage, and alarm triggering.

After we built the hardware prototype and developed alpha phase software, we conducted tests including thermal camera streaming verification, temperature precision and resolution, calibration, thermally reflective surface measurements and software functionality verification to prove compliance to the design specifications.

Specifications	**Expected**	**Achieved**
Frame Rate	8.7 [fps]	6 [fps]
Video Data Interface	SPI	✓
Control Interface	I^2C	Not utilized
IR Video Resolution	80 x 60	80 x 60
Ethernet Protocol	TCP/IP	✓
Power Source	Power over Ethernet	12V Adapter
Supply Voltage	7 – 12 [V]	12 [V]
Dual Camera System	N/A	N/A
Storage Media	Video	✓
Encapsulation	3D printed case	Onyx Material
PCB	Extension Board	✓
IR Window	Germanium (Ge)	✓
Accuracy	+/- 2 [°C]	+/- 2 [°C]
Precision	+/- 0.1 [°C]	+/- 0.11 [°C]
Standard Deviation	N/A	0.11 [°C]
Maximum Deviation	N/A	0.175 [°C]
Measurement Resolution	0.1 [°C]	0.11 [°C/step]
Avg Step Size	N/A	0.06 [°C/step]
Resolution	N/A	0.11 [°C/step]

The shortcomings of the alpha stage firmware and software are that some functions and algorithms are not optimized for faster and more stable operation. On the firmware side, the Ethernet protocol implementation introduced occasional delays during the transmission due to the large data throughput. OSIMS alpha stage software used a temporary solution to circumvent the difficulty in transferring data between the frontend and backend. In the future, we will keep improving on the alpha stage prototype, addressing issues encountered to improve run-time stability.

The OSIMS final prototype is shown below.

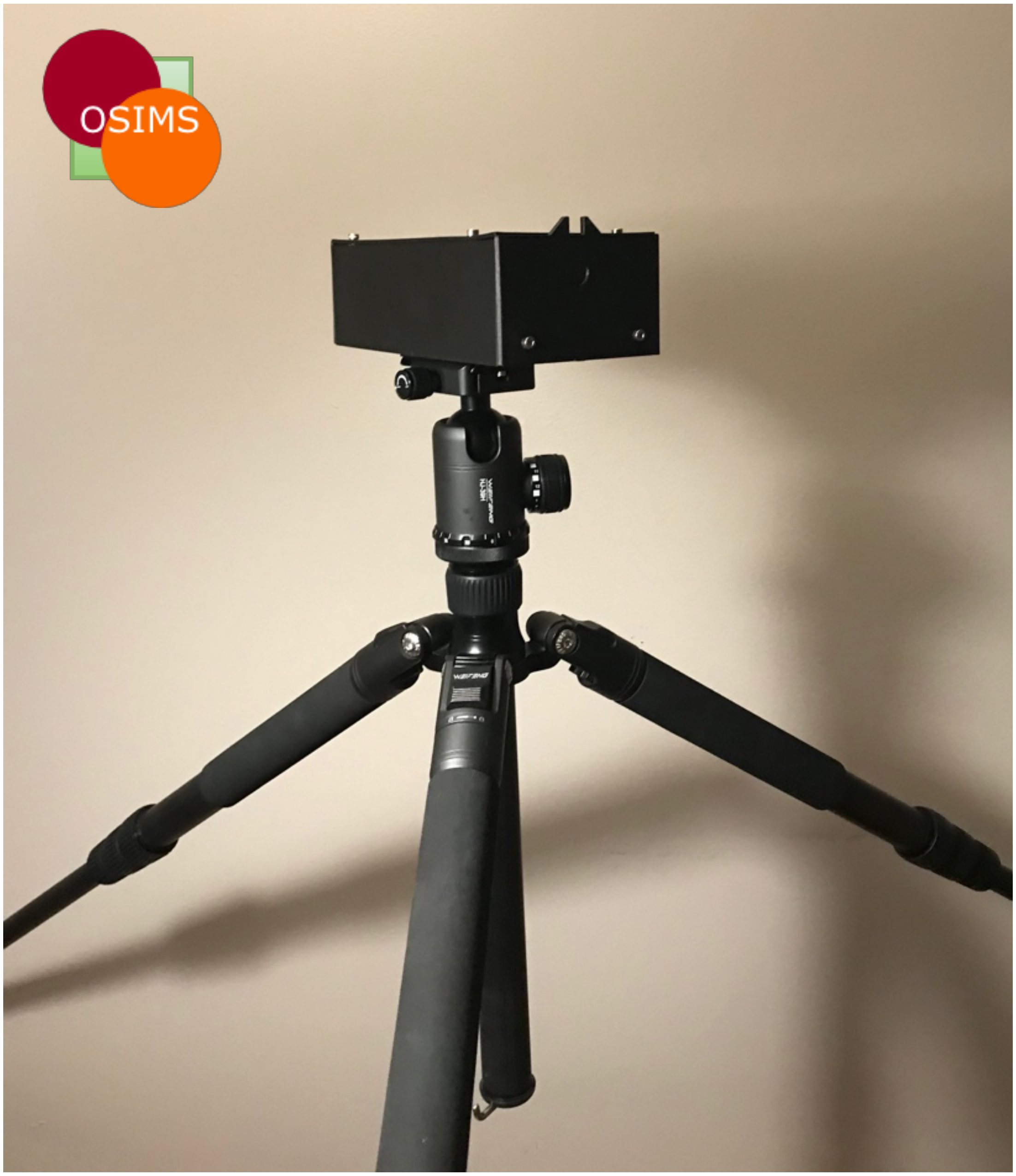


TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Industrial Failures

In recent years, the scale of destruction related to electric failures in industrial power systems is increasing as a result of modern factory productions becoming more automated and the electric systems of the factories becoming more complicated. An electric failure in a modern high-voltage power system can cause personnel fatalities as well as property loss on a large scale.

Dushanzi, China

In 2006, the Sinopec facility in Dushanzi, China, encountered a blackout, forcing the production line to shut down for 3 days and damaged several machineries, causing a direct property loss of 5 million RMB and an estimated production approximately 1.2 million USD loss [2].

Figure 1 Dushanzi Transmission Failure

The post-investigation found out that the blackout of the 35kV power system was caused by a transformer insulator wear out due to long-term regular usage and weather conditions. The excess current leakage eventually ignited the transformer and induced resonance and current surge in an adjacent line which was connected to several hydrogen compressors. Despite safety mechanisms, this accident caused a production shutdown and machinery damage.

Qingyang, China

In July 26th, 2015, rapid fire and explosions occurred in China Petroleum Qingyang refinery facility. Three employees were killed on the job [3]. This incident was due to a malfunction of a pressurized residual oil cooling device. The excess internal pressure caused the device to break down, igniting oil-transferring pipelines nearby, and caused a massive explosion. It was too late for the safety mechanism to take actions.

Figure 2 Refinery Explosion at Qingyang

Texas City, Texas

In March 23, 2005, an explosion occurred at a BP refinery in Texas City, Texas. It is the third largest refinery in the United States and one of the largest in the world, processing 433,000 barrels of crude oil per day and accounting for three percent of that nation's gasoline supply [4].

Figure 3 Texas BP Refinery Explosion

Over 100 were injured, and 15 were killed. The investigation indicated that several level indicators failed, leading to overfilling of a knockout drum, and light hydrocarbons concentrated at ground level throughout the area. A nearby running diesel truck set off the explosion. The catastrophic event was caused by undetected machinery failures.

All these dramatic incidents are caused by the wear-out of heavy-duty machineries due to unpredictable and "invisible" long-term effects, combined with a very large quantity of explosive chemicals.

1.2 Problem Statement

Potential failures in high voltage power systems can cause catastrophic events which result in worker fatalities, property damage and environmental pollution. These failures are due to undetected insulation wear-out and abnormal operation of machinery. In certain industries such as petroleum, the production involves explosive and hazardous chemicals, which means ensuring system integrity is a highly critical task. The electric failures typically result from undetected flaws in the system. Without knowing the condition of the system, invisible machinery wear-out can turn into potential disasters.

1.3 Our Proposal

Heat is an efficient indicator of flaws and faults in the system. When electric machines encounter excessive stress or internal corruption that leads to overloading, the core will heat up and be reflected on the surface temperature. Similarly, when the insulator material becomes damaged and degraded enough to cause a leakage current, the surface will also heat up.

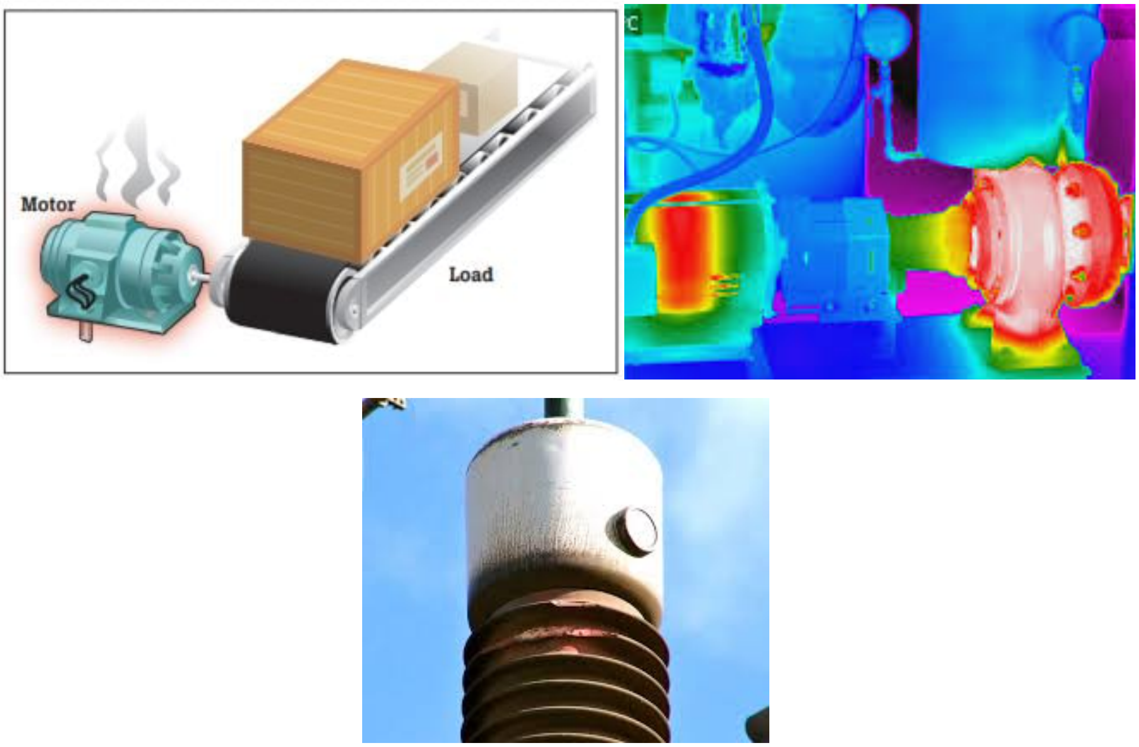


Figure 4 Machine Overload and Insulation Wear-out

The purpose of OSIMS is to provide a safer working environment for factories, to help businesses maintain profitability by monitoring and preventing electric failures. OSIMS provides continuous infrared video streaming and system diagnostic, which helps to identify potential flaws and faults in the industrial power system, including insulation wear-out, machinery internal corruption, machinery overload and overheating.

1.4 Literature Review

1.4.1 Heat Transfer Mechanisms

Thermal energy is related to the kinetic energy of molecules. The greater a material's temperature, the greater the thermal agitation of its constituent molecules. Heat gain or loss is caused by three primary mechanisms: (1) conduction, (2) convection, and (3) radiation [5].

Conduction involves heat transfer between two objects in contact. The regions with greater molecular kinetic energy will pass to regions with less molecular energy through direct collision, which refers as the "conduction" process. In metals, thermal energy is also carried by conduction-band electrons (the band of electron orbitals that electrons can jump up into from the valence band, when excited).

Convection occurs when a liquid or gas comes in contact with a material of a different temperature. This heat conduction causes the fluid to experience a volumetric expansion. The expanded fluid becomes buoyant and displaces, thus transferring heat by fluid motion in addition to conduction. Natural convection occurs when there is a density difference within the liquid due to temperature change (heating or cooling). Forced convection occurs when there is a pressure difference within the liquid or gas.

Radiation is the transfer of heat from one object to another due to electro-magnetic waves. The energy is carried by photons of light in the infrared and visible sections of the electromagnetic spectrum. When temperatures are not uniform, thermal energy is transferred from surfaces of higher temperatures to surfaces of lower temperatures.

Additionally, there are three other mechanisms that influence heat gain/loss by affecting insulation effectiveness: (1) Air infiltration, (2) air intrusion, and (3) moisture accumulation.

Air infiltration bypasses insulation. Heat is transferred due to gross flow of air generated from the air pressure difference between the exterior and interior. Air intrusion occurs when air enters the insulation from the exterior and remains in the interior. Moisture accumulation within materials reduces the insulation's capability of resisting heat flow, contributing to heat loss or gain.

Joule's Law

Joule heating can be described at a more microscopic level as the flow of electrons through an electrical path whose conductivity is limited. As flowing electrons collide with the atoms that comprise the molecules of the conductor, the electron's energy is transferred to the conductor's molecules in the form of heat.

The amount of heat (H) produced on a conductor, is proportional to the square of the amount of current (I) that is flowing through the conductor, when the electrical resistance of the conductor and the time of current flowing are constant [6]. The amount of heat produced is proportional to the electrical resistance (R) of the wire when the current in the wire and the time of current flowing are constant. The heat generated due to the flow of current is proportional to the time (t) of current flowing, when the electrical resistance and the amount of current are constant. All three conditions result to the following, as one can see its derivation from the Joule's law of power (P):

$$P = I^2 \cdot R \tag{4}$$

$$H = I^2 \cdot R \cdot t \tag{5}$$

The lower the resistance of a conductor, the smaller the voltage drop across it, resulting in less heating on the conductor for a given load. Considering the same amount of current flow, if higher resistance exists on the conductor, there is more voltage drop, therefore more heating is generated.

This introduces the importance of controlling or monitoring the energy flow in a system as well as the temperature change, in order to ensure efficiency and avoid catastrophic consequences, especially in oil refineries or industrial production facilities, where the power delivered in the system is in the order of mega-watts.

1.4.2 Industrial Temperature Monitoring

Many industrial systems rely on temperature measurements. Two important measurements involve heat flux and heat transfer. A change in temperature along with heat flux provide additional information on locating the source of heat. Heat flux (energy flux onto or through a surface, in [W/m^2]) is often detected earlier than temperature change, which contributes in a better process control and faster response to emergency situations [8].

Some applications of temperature measurement in industrial applications are as follows:

Table 1 Purposes behind Temperature Measurements in Industry

Process Monitoring	System Protection	System Energy Efficiency	Process Monitoring	Personnel Safety	Energy Efficiency
Creating a performance baseline	Detection of failures	Detection of wear of insulating layers	Detection of deposition of fouling	Measurement of heat stress by radiation	Studies of insulation / thermal resistance

1.4.3 Types of Temperature Sensors

Depending on the industrial application, different sensors are used to measure temperature. It is important to understand the characteristics of each sensor in order to determine the one that suits the application requirement. Some of the types of temperature sensors are as follows [9]:

Thermocouple Temperature Sensors consists of two wires made of different metals. The wires are joined at one end. As the temperature changes at the juncture, the thermocouple induces a change in Electromotive Force between the ends, which increases when the temperature rises.

Change-of-state Temperature Indicators comprise pellets, or liquid crystals which experience a change in appearance at certain temperatures. These are used in applications such as steam traps. These devices do not have good response time, hence no good response to transient temperature changes. In addition, the change of state in these devices is permanent, except in case of Liquid-Crystal Displays (LCDs).

Resistance Temperature Detectors (RTD) rely on change in resistance of a metal. The resistance rises linearly with temperature.

Thermistors depend on resistance change of a ceramic semiconductor, where the resistance drops non-linearly with increasing temperature.

Fluid-Expansion Temperature Sensors are represented by the household thermometer. There are two types of these devices, organic-liquid and mercury type. The key characteristics are that these devices do not pose explosion hazards, do not need electric power, and maintain stability during repeated usage.

Bimetallic Temperature Sensors capture the difference in the rate of thermal expansion between different metals. Two strips of different metals are joined together. When the device is heated, one side of the strip will expand more than the other. The bending as a result of the expansion is converted into a temperature reading by a pointer. These devices are not as precise as RTDs or thermocouples. However, these devices are portable and do not require a power supply.

Infrared (IR) Temperature Sensors are non-contacting. The devices measure temperature from the thermal radiation emitted by the material. The infrared sensor consists of a lens that focuses the infrared energy onto a detector. This energy is then converted into an electrical signal, which is displayed in the units of temperature.

Considering that the application of this project targets high power production facilitates or oil refineries temperature monitoring, infrared technology would be the optimal choice for this implementation. Common IR temperature sensors point to a specific point on the surface. However, in this application, wide-angle coverage and significant spatial resolution are required. Therefore, infrared cameras would be utilized.

1.4.4 IR Measurement Principle

Objects of different temperatures emit infrared light waves at different wavelengths, which is the fundamental principle of thermal imaging cameras. This effect is known as the blackbody radiation [10]. In general, the wavelength (λ) of the infrared radiation at peak intensity emitted from the object has the following relationship with its temperature (T): (b = 2.8977685×10^{-3} meter-Kelvin)

$$T = \frac{b}{\lambda_{max}} \tag{6}$$

At higher temperature, the object emits electromagnetic waves of shorter wavelengths, and vice versa, at lower temperature the object emits at longer wavelengths. In fact, the object will emit a bandwidth of electromagnetic waves at multiple wavelengths. Figure 5 shows the IR emission spectra at different temperatures.

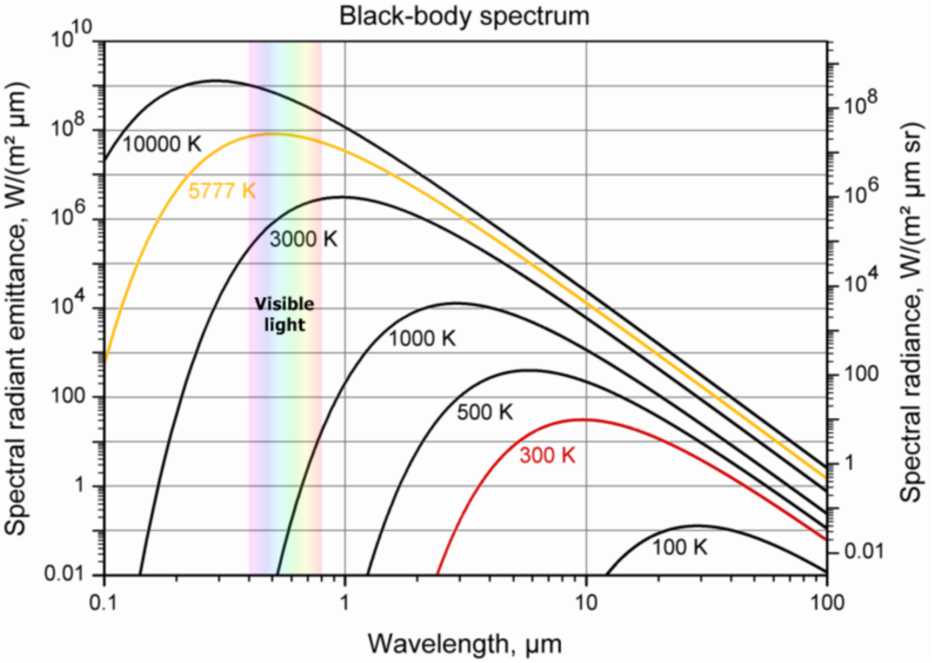


Figure 5 Blackbody Radiation Spectrum

Infrared energy is part of the electromagnetic spectrum in frequencies between visible light and radio waves as shown in Figure 6. It travels through space at the speed of light and can be reflected, refracted, absorbed and emitted. The wavelength range is between 0.7 micrometers to 1000 micrometers (microns) [11].

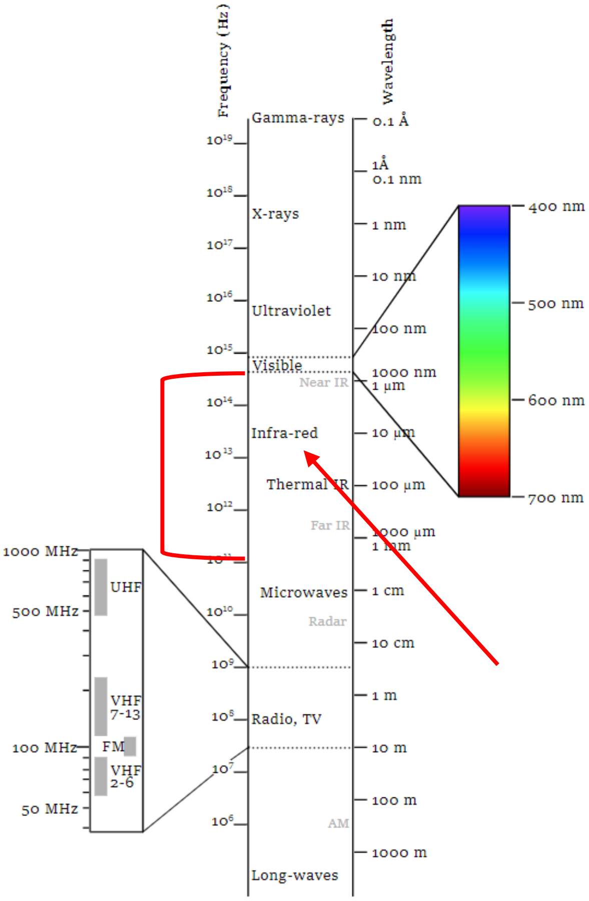


Figure 6 Electromagnetic Spectrum

In the case of opaque solid objects, part of the IR energy hitting the object's surface will be absorbed, and part will be reflected. A portion of the energy absorbed by the object will be re-emitted, and part will be reflected internally. The process is shown in detail in Figure 7.

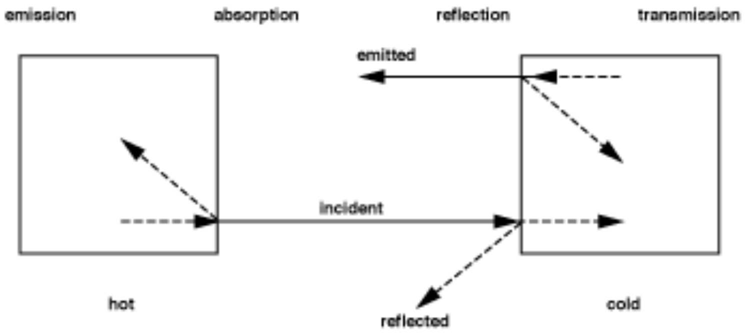


Figure 7 IR Radiation Energy Flow

These phenomena contribute to what is referred as the Emissivity of the object or material. Emissivity is defined as the ability of an object to emit infrared energy. Materials that do not reflect or transmit any IR energy are known as Blackbodies, which theoretically obtain a value of 1.0 emissivity. Emissivity varies with material types; therefore, IR radiation will be emitted with different intensities at a given temperature. This introduces Planck's law of blackbody radiation, which states that every object emits radiant energy, and the intensity of radiation is a function of the object's temperature [12].

$$B_\nu(\nu, T) = \frac{2h\nu^3}{c^2}\frac{1}{e^{h\nu/kT} - 1} \tag{7}$$

A real body emits only a fraction of the thermal energy emitted by a blackbody at the same temperature. If the emissivity is constant and independent of the wavelength, the body is a graybody.

The emissivity of real objects is not constant nor independent of the wavelength, thus, they cannot be considered graybodies. The emissivity of a material is a function of its molecular structure and surface characteristics. It is not affected by its surface color, unless the source of the color is a radically different substance to the main body of the material. In relation to the surface, a highly polished surface reflects more IR energy, thus affecting emissivity (i.e. highly polished stainless steel has a much lower emissivity than the same piece with a rough surface).

However, it is usually assumed that for short wavelength intervals, the emissivity can be considered constant, treating the real object as graybodies. Thus, although the emissivity of real objects is dependent on wavelength, they are still treated as graybodies by averaging the emissivity through short intervals, in which the infrared sensor working principle relies on. This average is possible because the emissivity is a slow-varying function of wavelength for solid objects, but not for gases or liquids. This introduces the Stefan-Boltzmann formula for graybody radiators, which is also graphically represented below.

$$W = \varepsilon \cdot \sigma \cdot T^{4} \tag{8}$$

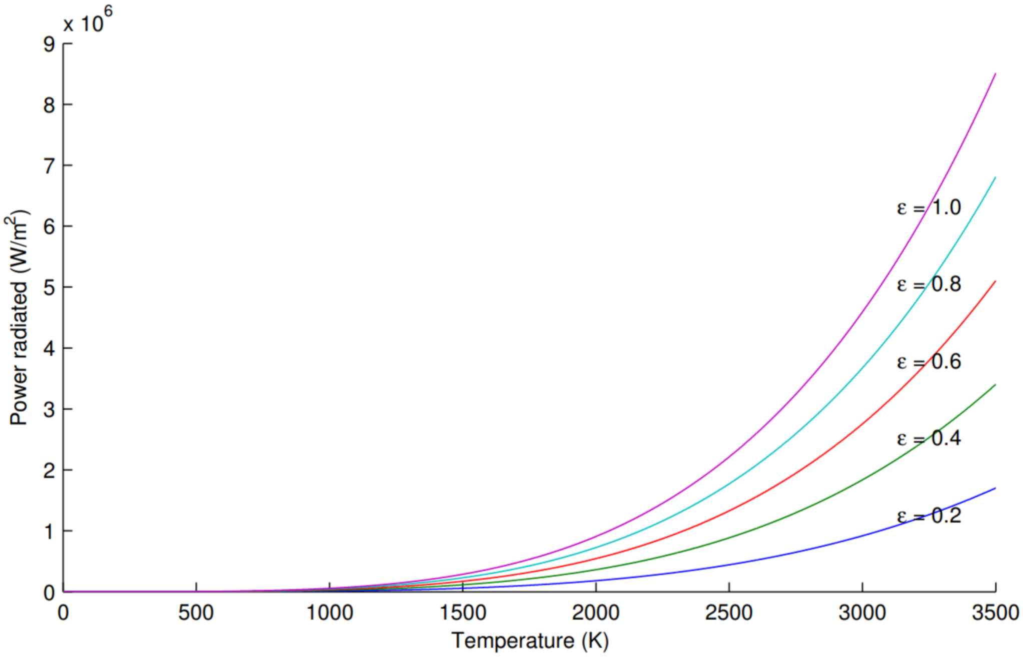


Figure 8 Power radiated by a Graybody with different Emissivity

If all the radiation energy falling on an object is absorbed, the absorptivity (*α)* is one. At a steady temperature, all the energy absorbed must be emitted, thus the emissivity (*ε)* of such body is one. Therefore, there is a direct linear relationship between the absorptivity and emissivity.

$$\varepsilon = \alpha \qquad (9)$$

According to Kirchhoff's law, the emissivity and absorptivity of any material are equal at any specified temperature and wavelength. Graybodies emit only a portion of the thermal energy emitted by an equivalent blackbody. Therefore, the emissivity in these bodes is always less than one, and the reflectivity (the infrared energy that is bounced off the target, which indicates the temperature of incident radiation from the target's surroundings), greater than zero.

1.4.5 Utilizing IR Cameras for Temperature Measurement

An IR camera is a non-contact device that detects IR energy or heat and converts it into an electronic signal, which is processed to produce a thermal image on a video monitor and perform temperature calculations. The heat sensed by the IR camera can be quantified and measured, allowing to not only monitor thermal activity, but also evaluate and identify the severity of heat-related problems [13].

However, not all the radiation received comes from the target object. External parameters, such as the surrounding objects or the atmosphere, contribute on the overall radiation detected. The total radiation received by the camera (W_{tot}) is a product of three sources: the emission of the target object (E_{obj}), the emission of the surroundings, reflected by the object (E_{refl}), and the emission of the atmosphere (E_{atm}) [14].

$$W_{tot} = E_{obj} + E_{refl} + E_{atm} \tag{10}$$

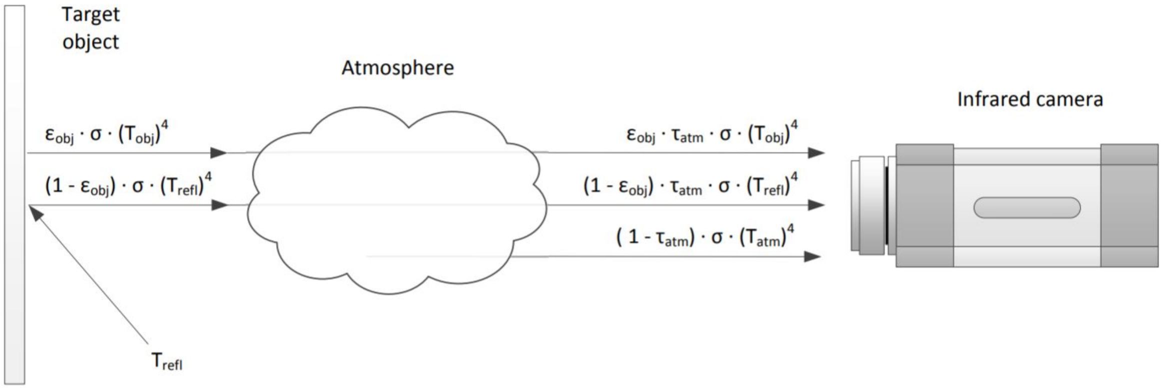

Target object
Atmosphere
Infrared camera
$\varepsilon_{obj} \cdot \sigma \cdot (T_{obj})^4$
$(1 - \varepsilon_{obj}) \cdot \sigma \cdot (T_{refl})^4$
$\varepsilon_{obj} \cdot \tau_{atm} \cdot \sigma \cdot (T_{obj})^4$
$(1 - \varepsilon_{obj}) \cdot \tau_{atm} \cdot \sigma \cdot (T_{refl})^4$
$(1 - \tau_{atm}) \cdot \sigma \cdot (T_{atm})^4$
T_{refl}

Figure 9 Radiation captured by the IR Camera

Not all radiation emitted by the target object is received by the camera; as a function of the transmittance of the atmosphere (τ_{atm}), a portion is absorbed by the atmosphere. Therefore, the emission of the target object can be expressed as follow:

$$E_{obj} = \varepsilon_{obj} \cdot \tau_{atm} \cdot \sigma \cdot (T_{obj})^{4} \tag{11}$$

Graybodies have a reflectivity greater than zero, thus the infrared radiation emitted by the surroundings is reflected. The reflectivity can be calculated by the emissivity, where part of this reflected radiation is also absorbed by the atmosphere.

$$E_{refl} = \rho_{obj} \cdot \tau_{atm} \cdot \sigma \cdot (T_{refl})^{4} = (1 - \varepsilon_{obj}) \cdot \tau_{atm} \cdot \sigma \cdot (T_{refl})^{4} \tag{12}$$

The other factor is the emission of infrared radiation from the atmosphere, which can be expressed as follow:

$$E_{atm} = \varepsilon_{atm} \cdot \sigma \cdot (T_{atm})^{4} = (1 - \tau_{atm}) \cdot \sigma \cdot (T_{atm})^{4} \tag{13}$$

Combining all the equations, the temperature of the object can be derived:

$$T_{obj} = \sqrt[4]{\frac{W_{tot} - (1 - \varepsilon_{obj}) \cdot \tau_{atm} \cdot \sigma \cdot (T_{refl})^4 - (1 - \tau_{atm}) \cdot \sigma \cdot (T_{atm})^4}{\varepsilon_{obj} \cdot \tau_{atm} \cdot \sigma}} \tag{14}$$

The of the parameters needed to calculate the temperature are: emissivity of the object, the reflected temperature, the transmittance of the atmosphere, and the temperature of the atmosphere.

The transmittance of the atmosphere can be estimated by using the distance from the object to the camera and the relative humidity. For the expected distances, the transmittance is close to one. The temperature of the atmosphere can be acquired using a common thermometer. However, as the emittance of the atmosphere is close to zero, this parameter itself has little influence on the temperature measurement. Contrarily, the emissivity of the object and the reflected temperature have a significant influence on the temperature measurement and must be recorded precisely for an accurate result.

Another important factor when doing remote temperature measurements is spot size ratio, which is the ability of an infrared camera to measure an object from a given distance. The processing software functions by averaging a certain number of pixels together in order to calculate the actual temperature as shown in Figure 10.

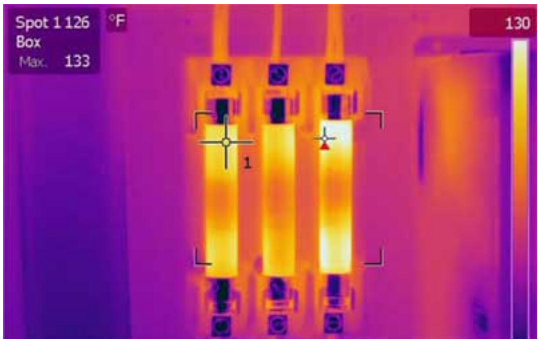


Figure 10 Thermal Camera Image

As shown in the image, "Spot 1" refers to the small circle that represents the number of pixels that are averaged together to create a temperature reading. Spot size is usually represented as a ratio (i.e. 20':1" spot size ratio can measure a 1 in object from 20 feet away). By moving closer or further from the target, the spot size will correspondingly decrease or increase.

Pixel count and optics play an important role in determining the spot size ratio of an infrared system. An example with three different resolutions will be considered for demonstrating the significance of pixel resolution in IR cameras. The larger the pixel count, the denser the measurement spot, resulting in more accurate temperature readings of small or narrow objects. Below are images from three different IR cameras with 160x120, 320x240 and 640x480 pixel resolution.

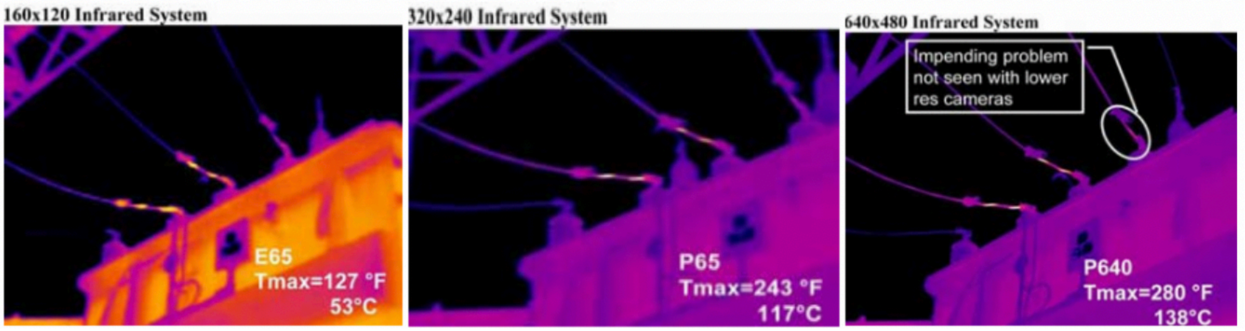


Figure 11 160x120, 320x240, and 640x480 IR systems temperature measurement

As the spot size ratio increases, a smaller point is measured, thus a more precise reading of the hot spot is obtained. A telephoto lens would improve the spot size ratio of an infrared camera by narrowing down the field of view. This "tightens" the pixels into a narrower field of view, thus resulting in more pixels on a measurement spot. It is important to refer to these key factors prior to implementing an IR camera or system in order to ensure accuracy in measurement for the given application.

2 COSTUMER REQUIREMENTS

In the winter of 2018, we paid an onsite visit to the SINOPEC facility in Beijing, China. SINOPEC is one of the largest petroleum and gas corporation in the world. The purposes of this visit were to find out the customer requirements and limitations of safety monitoring systems, and to familiarize ourselves with the field of applications.

The SINOPEC Beijing facility is a large industrial complex with mixed structures (indoor / outdoor). Some electric machines are in a controlled ambient environment, whereas others might be exposed to mixed weather conditions. In Figure 12, on the left there is a 35kV to 6kV step down transformer exposed outdoors. On the right is 35kV power line switch placed indoors. After speaking with the director of engineering, the customer requirements in the following section were derived.

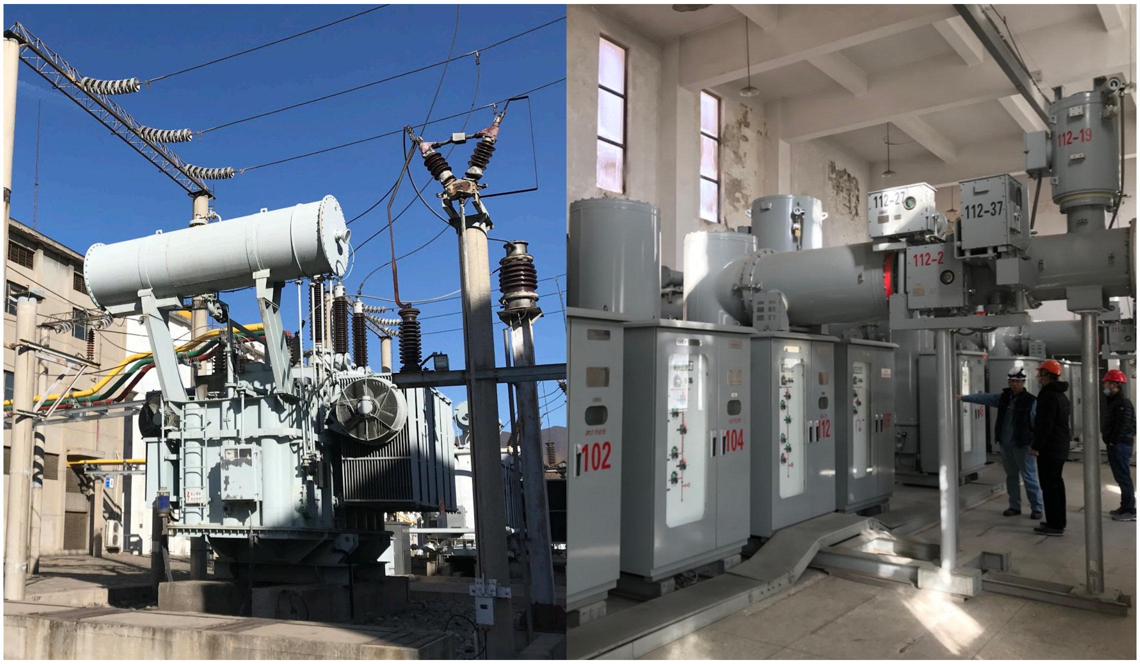


Figure 12 SINOPEC Beijing Facility

Safety Reliability

Our customer has very strict safety regulations for hazard control. Therefore the monitoring equipment:

- Does not interfere with the operation of the system.
- Does not cause fire or explosions.
- The installation does not impose danger to work place safety.

Long Distance and Mixed Weather Conditions

The machineries and power system of SINOPEC facility cover a very large area. Some monitoring points and the control room can be remotely located. This include both indoor and outdoor monitoring:

- OSIMS has to achieve long-distance and fast communication.
- Reduce the installation difficulty over long distance.
- Weather and water proof to account for exposure to mixed weather conditions.

Continuous Accurate and Precise Monitoring

The SINOPEC facility requires continuous monitoring of the system that has multiple electric machineries:

- Continuous real-time monitoring.
- Multiple corner monitoring.
- Software run-time stability for very long operating time.
- The change in temperature is more important than the actual temperature, therefore a high precision and a medium accuracy are required.
- Come up with a reflective surface monitoring solution. This is a requirement because some machineries have shiny reflective metal surfaces. For example, the surface of the power switch shown in Figure 12 is a thermally reflective (low emissivity) material.

Integrated Sensing Solution

Our customer prefers having the option of integrating thermal imaging with other sensing technologies. Such as humidity sensing and Partial Leakage Current Sensing (PLCS). Thermal Imaging (IR) technology is capable of detecting insulation wear-out, and abnormality within the system. PLCS is another preferred sensor in the industry for determining insulation weak spots.

Monitoring and Diagnostic Tool

The customer requires software for viewing the real-time data shared on the local network. The software has to have some functionalities to indicate system failures, and give comprehensive safety diagnostics with some data processing capability. More importantly, it has to trigger a warning when detecting significant rise in temperature.

3 COMPETITIVE MARKET ANALYSIS

Currently, FLIR Systems is the world's largest commercial company specializing in the design and production of thermal imaging cameras, components and imaging sensors. FLIR’s applications are: Government and Defense, Marine, Industrial, Public Safety, Security, Home and Outdoor. FLIR systems major product lines include: Surveillance (military systems), Instruments, Security, OEM & Emerging Markets, Marine and Detection. There are three products that are taken into consideration: FLIR A310, FLIR A310 EX, and FLIR AX8.

FLIR A310 camera comes with an 18 mm standard lens providing a 25 25°×19° field of view and 320×240 infrared resolution. The module has the ability to stream video over Ethernet to view live images on a PC, supports both communication and power over the Ethernet cable, and can be controlled remotely over the Web and TCP/IP protocol. The standard features include built-in analysis functions like spot area measurement, and temperature difference. Alarms can be set to go off as a function of a specific analysis, internal temperature or digital input. The camera automatically sends analysis results, infrared images and alarm triggering notification. Ethernet/IP and Modbus TCP allow the unit to share analyses and alarm result with a programmable logic controller (PLC)

FLIR A310EX features several digital I/O channels and sensors for temperature, humidity and pressure. Among other functions, the I/O channels enable the user to control the camera and the internal heater remotely. The purpose of the heater is to effectively prevent fogging and freezing of the protection window. Image access is provided through an integrated web interface or Modbus TCP/IP. The module is equipped with two fiber optic and two Ethernet ports, which enables a flexible network integration. The infrared resolution is 320x240 at 7-8Hz frame rate.

FLIR AX8 is a more compact approach to condition monitoring. The size is smaller compared to the other options, as a result of sacrificed features and capabilities. However, it provides a Multi-Spectral Dynamic Imaging, combining both visible and infrared camera video data to create a more detailed representation of the surroundings. The module offers an 80 x 60 infrared resolution with a field of view 48° x 37°. Also, this product provides the data over Ethernet on a web-based platform for data analysis. The table below shows a more detailed specification of each product:

Table 2 Table of Competitive Market Value Analysis

	Features	FLIR A310	FLIR A310 EX	FLIR AX8
	IR resolution	320x240 pixels	320x240 pixels	80x60
	Built-in Camera	No	No	640x480
	Zoom	1-8x	1-8x	Fixed
Connections & Communications	Ethernet	Yes	Yes	Yes
	Ethernet image streaming	16-bit 320x240 @4.5 Hz	16-bit 320×240 pixels@ 7-8 Hz	640x480
	Ethernet type	100 Mbps	100 Mbps	100 Mbps
	Storage media	Built-in memory	Built-in	Built-in
Imaging, Optical, and Measurement Analysis	Emissivity correction	0.01 to 1.0	0.01 to 1.0	0.01 to 1.0
	Op. Temp Range	-15°C to + 50°C	–20°C to +40°C	0°C to+50°C
	Field of view (FOV)	25° × 18.8°	25° × 18.8°	48° × 37°
	Image frequency	9 Hz	30 Hz	9Hz
	Spatial resolution	1.36 mrad	1.36 mrad	N/A
	Accuracy	±2°C	±2°C	±2°C
	Object temperature range	–20 to +120°C 0 to +350°C	–20 to +120°C 0 to +350°C	–10°C to +150°C
	Thermal sensitivity	< 0.05°C @ +30°C	< 0.05°C @ +30°C	< 0.10°C @ +30°C
Price		$8,995	$17,995	$995

Based on the customer requirements and the competitive market analysis, key features necessary for our applications can be derived. The main differences between the products mentioned above are related to the quality or resolution of the infrared video. The majority of the other features is similar. All products share power over Ethernet (PoE) capability, web interface, built-in storage, accuracy of ±2°C, etc.

Our primary goal is to achieve similar specifications at a lower cost of manufacturing. As seen above, the cheapest product is listed at $995. The main competitor for OSIMS would be FLIR AX8. This module provides an 80 x 60 infrared resolution and a 640x480 visible resolution. The streaming frame rate is around 9Hz, and features emissivity correction variable from 0.1 to 1.0. The web interface supports built-in analysis functions like spot area measurement, temperature difference and alarm. In addition, the module is rated IP67 to withstand different weather conditions.

Considering the current market solutions and our customer requirements, the next section will go over the target product specifications, within the scope of this major qualifying project.

4 DESIGN APPROACH

4.1 Product Specification

Thermal Imaging

- **Infrared Imaging:** The unit should support infrared streaming at a frame rate below 9 Hz to comply with US export restrictions.
- **Precision**: For high precision, set specification at +/- 0.1 [°C].
- **Accuracy:** +/- 2 [°C] (value derived from competitive market analysis).
- **Resolution:** 80x60.

Ethernet Communication

- **Protocol**: TCP/IP.
- **Power over Ethernet (PoE)**: The unit should utilize PoE technology to reduce the time and expense of having electrical power cabling installed. Installation and distribution of network connections are much simpler and effective .
- **Speed**: ≥ 100 [Mbps].

Installation Requirement

- **Exterior Integrity**: The case of the sensor unit must be insulated, weather proof and robust. A 3D printed case will be used for the scope of this MQP.
- **Physical Stability**: The integrity of the case/mount has to guarantee long-term physical stability of the product (does not fall/bend). The exterior structure and material need to be tested under certain stress and torque. The case has to be water proof to meet IP67 standard for the final product (does not apply within the scope of this MQP).

Software

- **Cross-Platform Development**: The software needs to be compatible for various operating systems including Windows, Linux and MAC OS
- **Safety Diagnostic and Alert**: The software must provide hot corner detection, data logs, alarm triggering, and video and raw data storage capability.

More detailed specifications are listed in Table 3

Table 3 List of Product Specifications

Categories	**Specifications**
Frame Rate	< 9Hz
Video Data Interface	SPI
Control Interface	I^2C
IR Video Resolution	80 x 60
Ethernet Protocol	TCP/IP
Power Source	Power over Ethernet
Dual Camera System	640x320 camera
Storage Media	.avi video data
Encapsulation	IP67-Rated
Precision	+/- 0.1 [°C]
Accuracy	+/- 2 [°C]
Measurement Resolution	+/- 0.1 [°C]
Resolution	0.1 °C
Software Operating System	Cross-Platform
Functionalities	Corner Temperature Plot; Multiple Corner Monitoring; Alarm Triggering

4.2 Design Options

4.2.1 Hardware

Microcontroller

Based on the product specifications, an analysis will be conducted to determine the appropriate components for this project. Five categories were used to evaluate the selection of a microcontroller:

- Performance
- Complexity
- Programming Tools
- Available Resources
- Price

The microcontroller should support three communication protocols; I^{2}C, SPI and Ethernet. Also, the performance of the processor plays an important role for handling big amount of data and fast transmission. Next, the microcontroller should be equipped and configured to operate in their corresponding programming environment. The configuration process should be feasible and user friendly. In addition, in a prototype stage, an evaluation board or the microcontroller should be popular such that there are sufficient available resources for reference. Finally, considering that during this stage the budget is limited, price is another factor that should be taken into consideration.

Two microcontrollers were put to comparison: NUCLEO-F767ZI by ST Microelectronics, and EK-TM4C1294XL by Texas Instruments.

NUCLEO-F76ZI runs on an ARM®32-bit Cortex®-M7 processor with a maximum CPU frequency of 216 MHz. It has 2MB Flash and 512 KB SRAM. In addition, it supports up to 4 I^{2}C channels and 6 SPI channels. The evaluation board has an on-board ST-LINK/V2-1 debugger/programmer with SWD connecter which is used for developing the firmware on the chip. It supports Ethernet 10/100Mbps. A third-party IDE based on Eclipse is available for free that incorporates a GCC C/C++ compiler and GDB-based debugger. In addition, another programming tool called STM32CubeMx is available to use for configuring the processor. This is a graphical software configuration tool that allows the generation of C initialization code using graphical wizards, which makes it ideal for starting on a project. The company has provided a package that contains the abstraction layer (HAL) and the low-layer (LL) APIs, plus a consistent set of middleware components (RTOS, USB, FAT file system, Graphics and TCP/IP). However, the documentation of the initialization code and the test examples provided lack details, making the code challenging to troubleshoot.

The EK-TM4C1294XL runs on a 32-Bit ARM® Cortex®-M4F processor with a maximum CPU frequency of 120 MHz. It has an on-bard USB port for power and programming/debugging. The board supports I^2C, SPI and Ethernet communication. TI has developed TivaWare which is software that provides drivers for all the peripheral devices supplied in the design. The microcontroller can be programmed through Texas Instruments Code Composer Studio IDE, which includes an optimizing C/C++ compiler, source code editor, project build environment, debugger and a profiler.

The following table shows the score given by the team for each corresponding category to both evaluation boards. As a result, NUCLEO-F767ZI was chosen for the development of this prototype.

The range is from 1-3, where 1 indicates BAD and 3 indicates GOOD.

Table 4 Microcontroller Evaluation

	Performance	**Complexity**	**Programming Tools**	**Available Resources**	**Price**	**Score**
NUCLEO-F76ZI	3	3	3	2	3	14
EK-TM4C1294XL	2	2	3	3	3	13

Thermal Camera

The second most important module is the thermal camera. Based on the specification derived, the thermal camera should support at least an infrared resolution of 80 x 60. It is important that the camera core comes integrated on a breakout board for installation feasibility. Surprisingly, the only available camera cores are produced by FLIR and are the most advanced micro camera cores currently in the market. Companies like GroupGets or Sparkfun have already designed a breakout board contains a socket for the Lepton, power supplies, a 25 MHz Crystal Oscillator and a 100-mil header for use in breadboard or wiring to any host system. The header consists of 8 pins that provide direct I^2C and SPI communication as well as power to the camera core. Three lepton cores were considered for this application and ranked on the following categories:

- Feasibility
- Resolution
- Price

FLIR Lepton 3.0 and 3.5 share the same infrared resolution of 160 x 120 while the Lepton 2.5 provides an 80 x 60 resolution. All of them are compatible and supported by the same breakout board. However, Lepton 2.5 and 3.5 are radiometric. A radiometric thermal camera measures the temperature of a surface by interpreting the intensity of an infrared signal reaching the camera. The benefits include: improved accuracy, typically on the order of 5 % in high-gain state, digital data linear in scene temperature, and temperature precision. The following table shows the score given by the team. As a result, Lepton 2.5 breakout board camera was chosen for the development of this prototype. The radiometric feature and the price were the determining factors in this module.

The range is from 1-3, where 1 indicates BAD and 3 indicates GOOD.

Table 5 Thermal Camera Evaluation

	Feasibility	**Resolution**	**Price**	**Score**
Lepton 2.5	3	2	2	7
Lepton 3.0	2	3	1	6
Lepton 3.5	3	3	1	6

Infrared Window

An infrared window let pass infrared light with desired wavelengths and reflects lights of other wavelength. The specific choice of a window is usually driven by the application, environment, wavelength and cost considerations. There are numerous types of lens materials that can be used in IR windows. For example, a mid-wave research and development applications with high ambient temperature requirements may use materials that would be unsuitable for long-wave condition monitoring of industrial applications. Below there is a table that shows the transmission range for different IR window materials used in the industry.

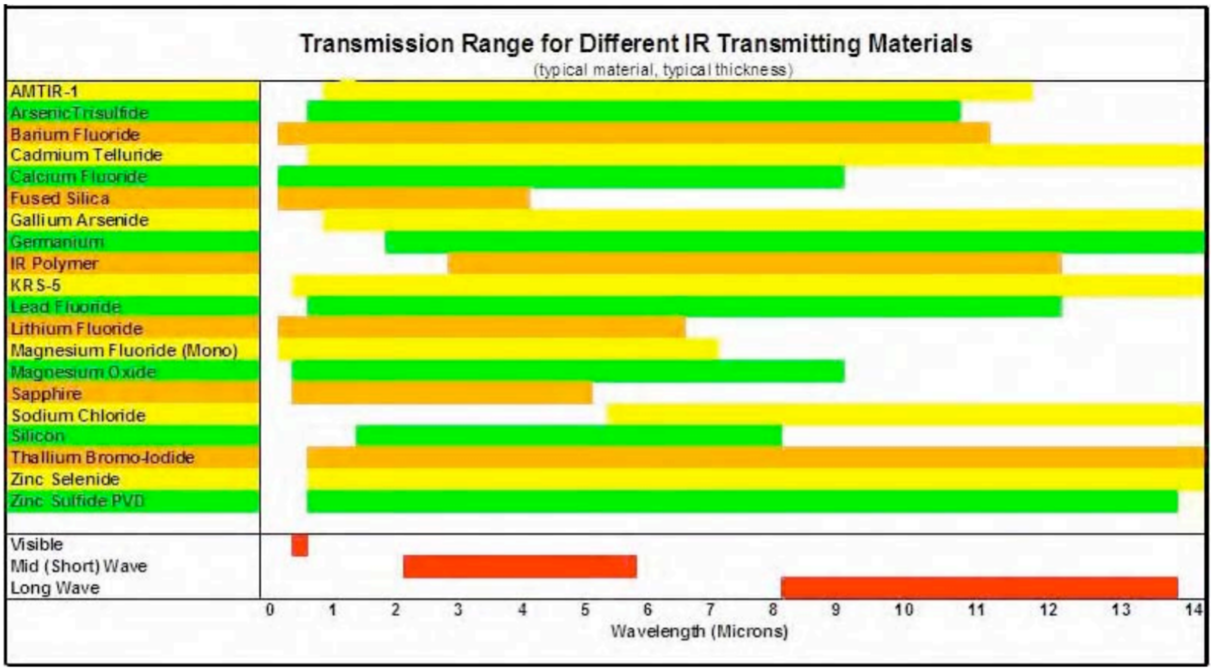


Figure 13 Transmission Range for Different IR Transmitting Materials

An important attribute is the index of refraction, which is the ratio of the speed of light in a vacuum to the speed of light within a given material. It can be referred as a “bending” effect of light as it enters a high index medium from a low index medium as shown below.

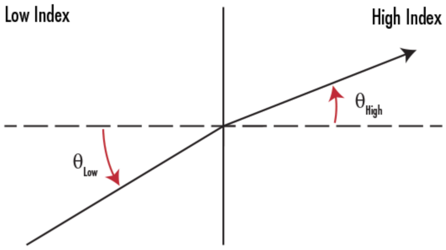
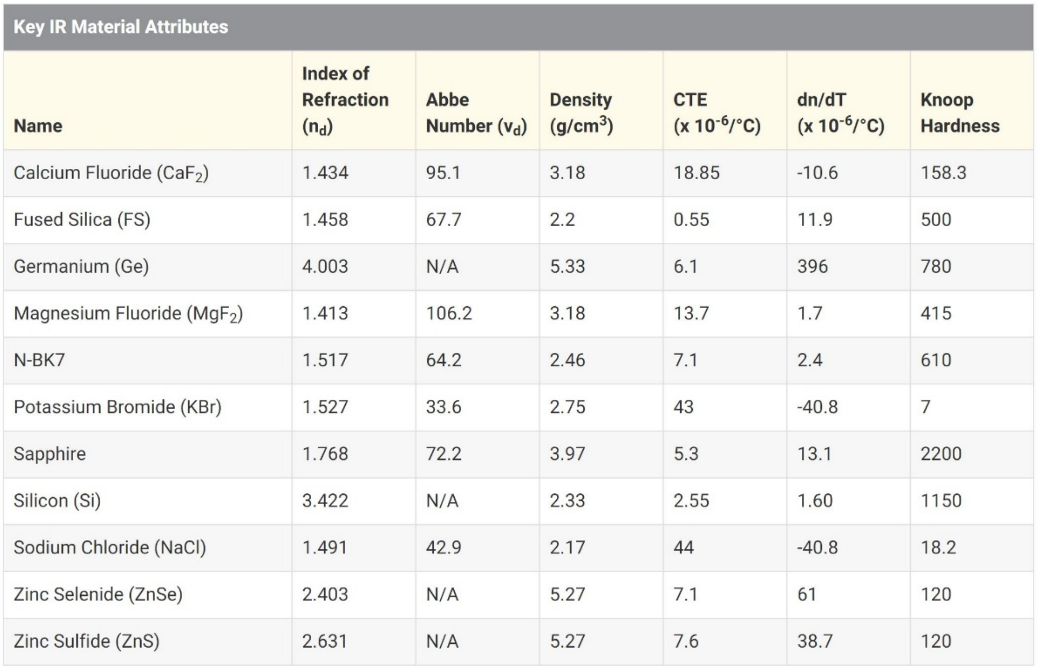


Figure 14 Light Refraction from a Low Index to a High Index Medium

Furthermore, another characteristic of the window is the dispersion, which is a measure of how much the index of refraction a material changes with respect to wavelength. This is given by the Abbe number (vd). The index of refraction of a medium varies as the temperature changes. The index gradient (dn/dT) can be troublesome when operating in unstable environments. Also, the density of the material plays an important role regarding the long term durability.



Key IR Material Attributes

Name	Index of Refraction (n_d)	Abbe Number (v_d)	Density (g/cm^3)	CTE (x 10^{-6}/°C)	dn/dT (x 10^{-6}/°C)	Knoop Hardness
Calcium Fluoride (CaF_2)	1.434	95.1	3.18	18.85	-10.6	158.3
Fused Silica (FS)	1.458	67.7	2.2	0.55	11.9	500
Germanium (Ge)	4.003	N/A	5.33	6.1	396	780
Magnesium Fluoride (MgF_2)	1.413	106.2	3.18	13.7	1.7	415
N-BK7	1.517	64.2	2.46	7.1	2.4	610
Potassium Bromide (KBr)	1.527	33.6	2.75	43	-40.8	7
Sapphire	1.768	72.2	3.97	5.3	13.1	2200
Silicon (Si)	3.422	N/A	2.33	2.55	1.60	1150
Sodium Chloride (NaCl)	1.491	42.9	2.17	44	-40.8	18.2
Zinc Selenide (ZnSe)	2.403	N/A	5.27	7.1	61	120
Zinc Sulfide (ZnS)	2.631	N/A	5.27	7.6	38.7	120

Figure 15 Key IR Material Attributes

The foremost attribute defining any material is transmission. Transmission is a measure of throughput and is given as a percentage of the incident light, which is shown below for different infrared window materials.

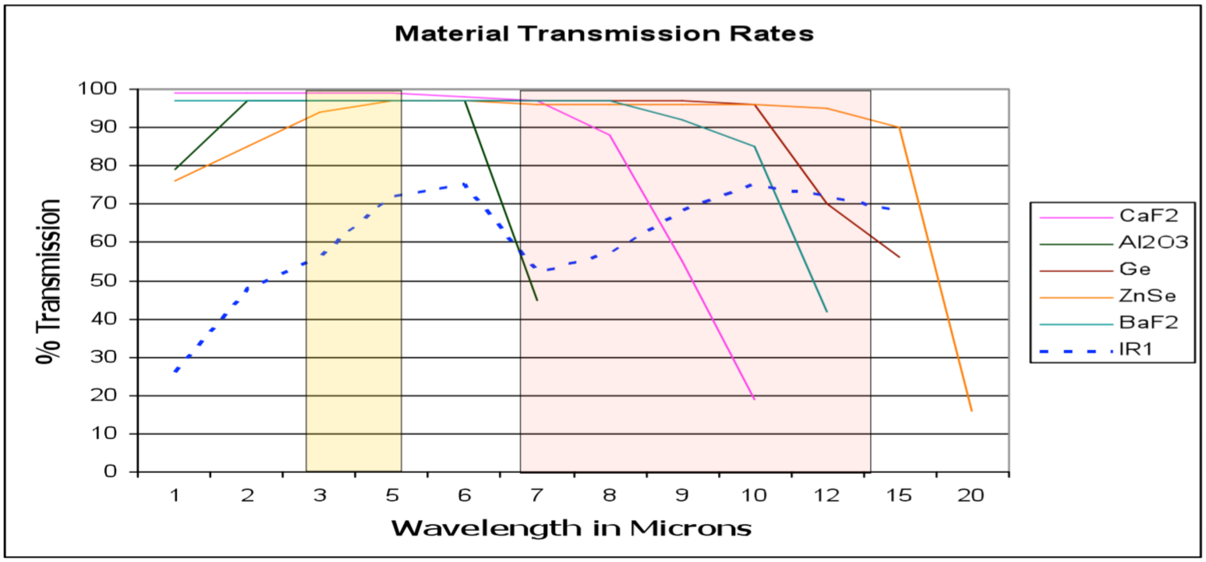


Figure 16 Transmission of Different Window Materials

As shown above, the selection of the windows directly depends on the application. Considering the cost of these windows and the budget limitations, a used germanium windows (12.7 mm diameter, 1 mm thick) was elected for this application as shown in Figure 17.

Germanium is very hard, high density material – characteristics that are advantageous where robustness is needed. The high index of refraction necessitates that good anti-reflection coating be used. More importantly, the germanium window has a good transmittance from 2μm to 12 μm. The window has a broad band anti-reflective (BBAR) coating for 8-12μm long-wavelength infrared (LWIR) thermal imaging / thermography applications.

Figure 17 Germanium Window

4.2.2 Software

Development Tools and Environment-- Frontend

There are several options for designing the frontend. Html/CSS/JS is a common language used for web based applications, it has low difficulty and can be supported in all operating systems because it will be in the form of a webpage. C++ has the highest difficulty because it is a low level language that can be powerful in terms of optimizing memory management and algorithm. However, it imposes higher development challenge for the same reason. C# is an enhanced, second-generation version of the C language, one of the earliest back-end programming languages. C# is a general-purpose, object-oriented version specifically developed by Microsoft for the .NET Framework. It is less difficult compare to C++, however still limited to Windows only applications. Electron.js is an html/CSS/JS cross-platform wrapper, turning simple webpage design into desktop applications. It has the least difficulty in development, and supports the most operating systems. Table 4 lists the comparison among all options. The criteria are equally weighed including:

- Operating system (3 for supporting cross-platform, 1 for single OS)
- Difficulty (3 for low difficulty, 2 for medium, 1 for high)
- Optimization (3 for high optimization, 2 for medium, 1 for low)
- Software or Web based (3 for Software, 1 for Web based)

Having the highest rating, Electron is chosen as the frontend development tool.

Table 6 Frontend Development Tool Options

	Html/CSS/JS	**C++**	**C#**	**Electron**
Operating System	Mac, Win, Linux 3	MAC, Win, Linux 3	Win 1	MAC, Win, Linux 3
Difficulty	Low 3	High 1	Medium 2	Low 3
Optimization	Low 1	High 3	High 3	Medium 2
Software/Web	Web 1	Software 3	Software 3	Software 3
Total	8	10	9	**11**

Backend

The options for backend developments can be any of the programming languages, three most commonly used languages were listed below: C++, Python, MATLAB. The criteria taken into consideration are scripting language (can be used by a webpage as a script), and difficulty for implementation, whether it complies with Electron frontend and the originality of the completed work. Scripting language statements can be alternatively executed by human operator one by one, which greatly improves the debugging and development feasibility. Flexibility indicate if the code can be executed independently from its programing environment and still have the libraries necessary for execution.

Table 5 shows the comparison among the options. Criteria include:

- Whether it is a Scripting language (3 for Yes, 1 for No)
- Difficulty (3 for low difficulty, 2 for medium, 1 for high)
- Works with Electron (3 for Yes, 1 for No)
- Originality (3 for High, 2 for Medium, 1 for Low)

Python was rated 11, therefore chosen for back end development. Additionally, MATLAB Python plug-in will be used because there are existing codes for plotting data.

Table 7 Backend Programming Language Options

	C++	**Python**	**MATLAB**
Scripting	NO 1	Yes 3	Yes 3
Difficulty	High 1	Medium 2	Low 3
Works with Electron	Yes 3	Yes 3	No 1
Flexibility	High 3	High 3	Low 1
Total	8	**11**	8

In summary, Figure 18 shows a complete architecture of the software design. The frontend framework is handled by Electron. The backend is developed in Python and MATLAB. For the development environment, Eclipse IDE is chosen. It supports various coding languages and scripting capability for power debugging. Last but not least, GitHub as our project repository.

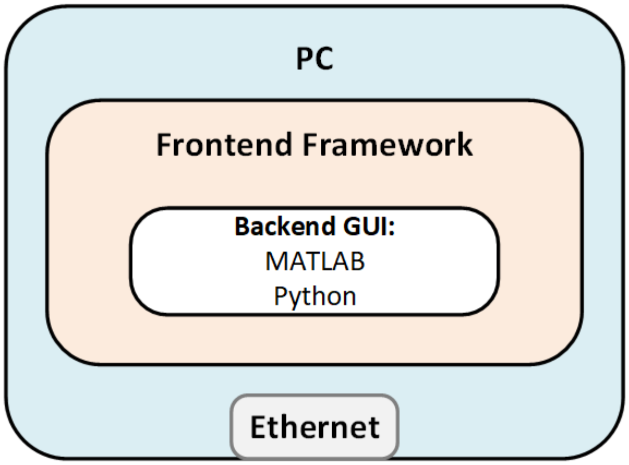


Figure 18 Software Structure

Table 8 Software Development Tools and Environment

Frontend	Backend	IDE	Repository
Electron	Python/MATLAB	Eclipse	GitHub

5 PRODUCT DEVELOPMENT

5.1 High Level Block Diagram

The first stage prototype of an OSIMS unit will consist of a microcontroller and a thermal camera core breakout board. The camera core supports CCI/TWI and SPI interface. CCI/TWI is similar to I^2C standard and is used for configuring and calibrating the camera, while SPI is used for video processing. The microcontroller will support I^2C, SPI and Ethernet communication protocols. There will be an assigned local IP address to each OSIMS unit, which will be connected via Ethernet to a router on a private network. The platform will support thermal video streaming over Ethernet to a local personal computer. There will be dedicated OSIMS software for data acquisition and visualization. The block diagram is shown below:

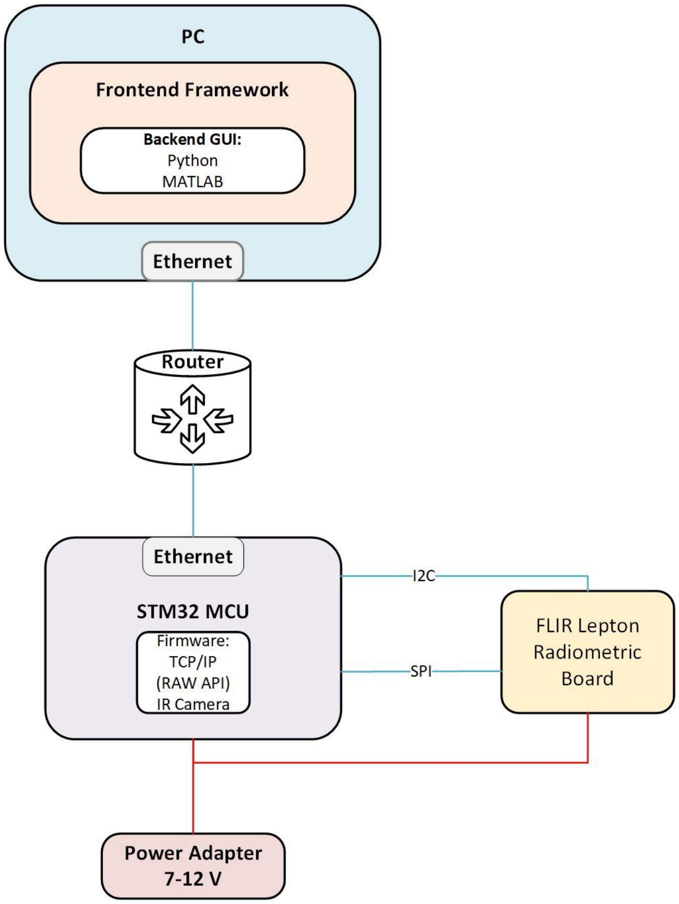


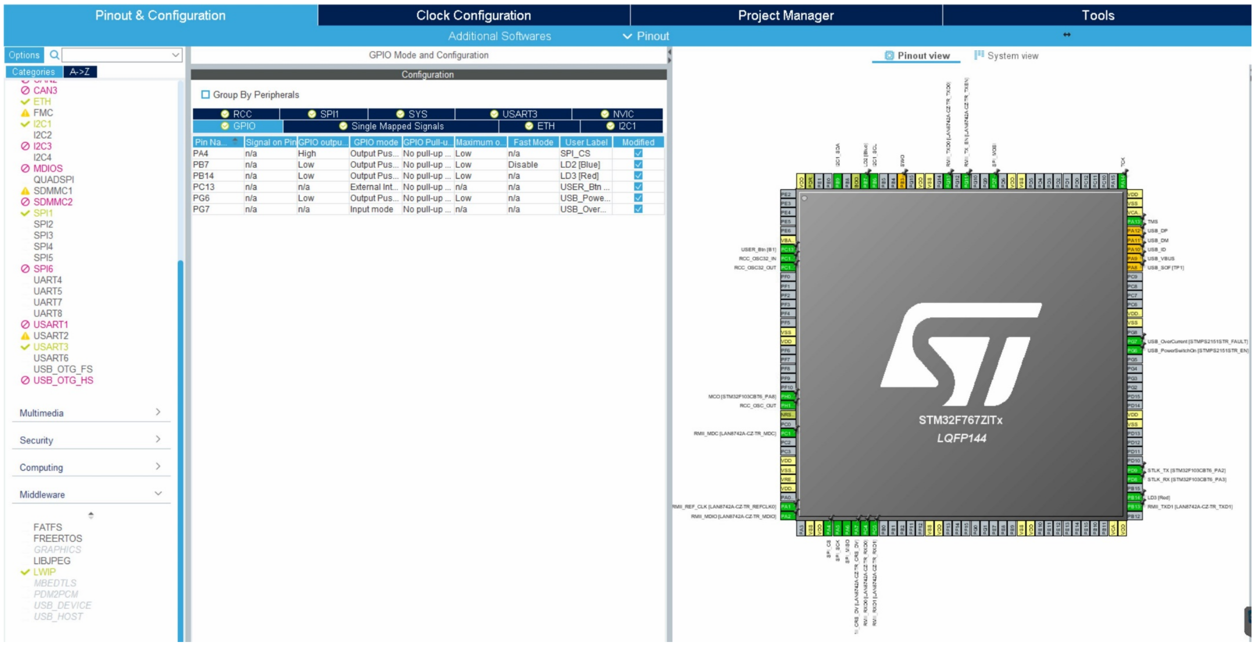
Figure 19 OSIMS Prototype Block Diagram

5.2 Microcontroller Configuration

The microcontroller is a 32-Bit ARM® Cortex®-M7 embedded evaluation board manufactured by STMicroelectronics. ST has partnered with a third-party company to develop SW4STM32, GCC-Based IDE used for building the firmware. Also, ST has provided STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards. It also embeds comprehensive packages composed of the hardware abstraction layer (HAL) and the low-layer (LL) APIs, plus middleware components (such as RTOS, USB, TCP/IP, and graphics). The tool incorporates pinout-conflict solver, a clock-tree setting helper, a power-consumption calculator, and an utility performing MCU peripheral configuration (GPIO, USART,) and middleware stacks (USB, TCP/IP).

In this design, the STM32CubeMX is used to configure the following:

- Connectivity: I2C, SPI, and ETH (Ethernet)
- Middleware stack: LWIP TCP/IP



Pin Na...	Signal on Pin	GPIO outpu...	GPIO mode	GPIO Pull-u...	Maximum o...	Fast Mode	User Label	Modified
PA4	n/a	High	Output Pus...	No pull-up ...	Low	n/a	SPI_CS	☑
PB7	n/a	Low	Output Pus...	No pull-up ...	Low	Disable	LD2 [Blue]	☑
PB14	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	LD3 [Red]	☑
PC13	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	USER_Btn ...	☑
PG6	n/a	Low	Output Pus...	No pull-up ...	Low	n/a	USB_Powe...	☑
PG7	n/a	n/a	Input mode	No pull-up ...	n/a	n/a	USB_Over...	☑

Figure 20 STM32CubeMX Configuration

Then, the internal clock frequency of the processor was configured to 216 MHz for optimal performance. There are two peripheral clock busses: APB1 and APB2. The configured SPI1 channel is connected to the APB2 peripheral clock bus, which is pre-scaled by a minimum factor of ½. This results in a bus frequency of 108MHz for the SPI. The clock configuration is shown below:

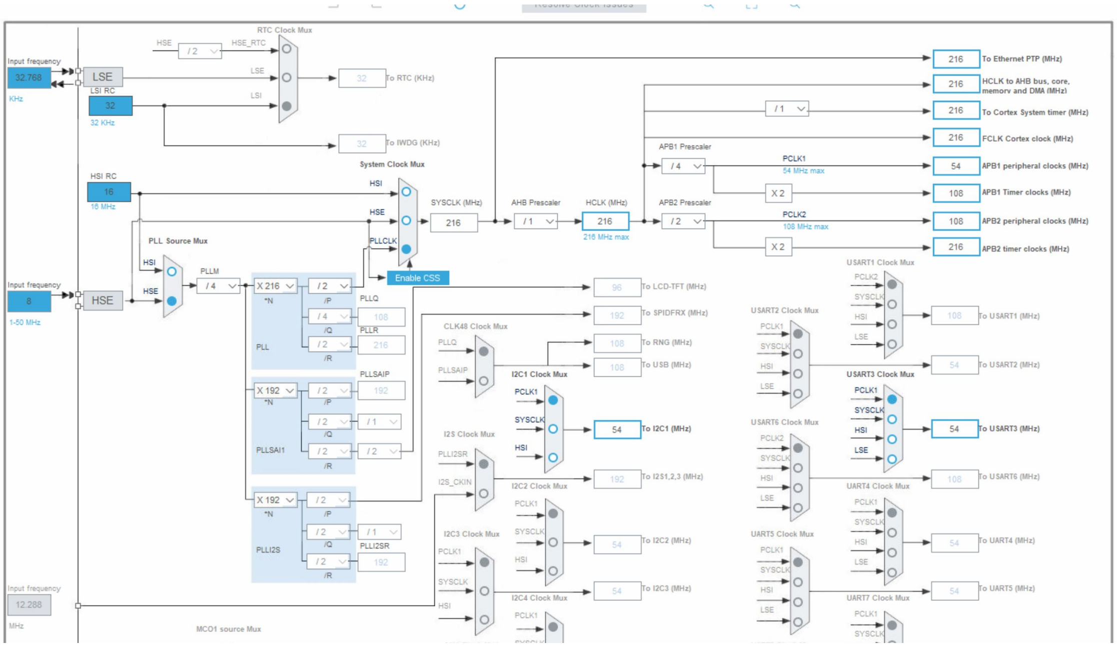


Figure 21 STM32CubeMX Configuration

Thermal Camera Configuration Requirements

The SPI protocol is packet-based with no embedded timing signals and no requirement for flow control. The master initiates the data transfer and controls the clock speed. The data can be pulled from the camera within a specified frequency range as shown below:

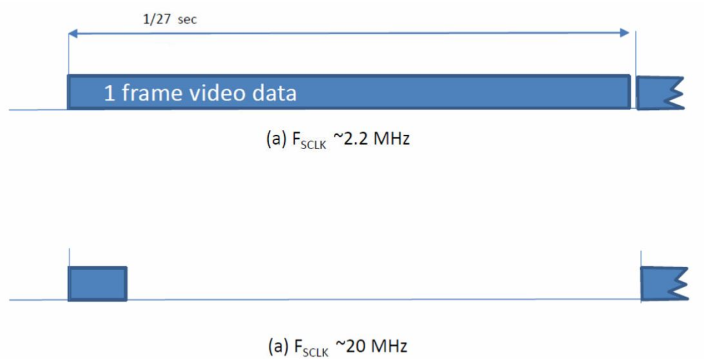


Figure 22 Thermal Camera SPI Clock Range

The maximum clock rate is 20 MHz. Thus, for this application the APB2 peripheral bus is pre-scaled by a factor of $\frac{1}{8}$, resulting in a clock speed of 13.5 MHz. Regarding the GPIOs, the camera only utilizes SCK, CS and MISO as shown below.

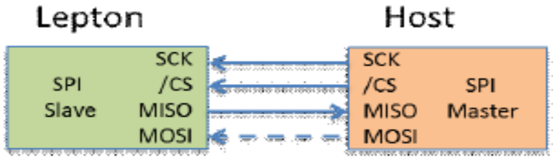


Figure 23 Thermal Camera SPI Configuration

The MOSI line can be either connected to a GPIO or to GND because the camera core is restricted to a single master and single slave type of operation. However, four GPIOs are configured:

- PA4 – SPI1_SSEL (active LOW)
- PA5 – SPI1_SCK (set HIGH when IDLE)
- PA6 – SPI1_MISO
- PA7 – SPI1_MOSI (set LOW)

The camera core uses SPI Mode 3: Clock Polarity (CPOL) = 1, Clock Phase (CPHA) = 1

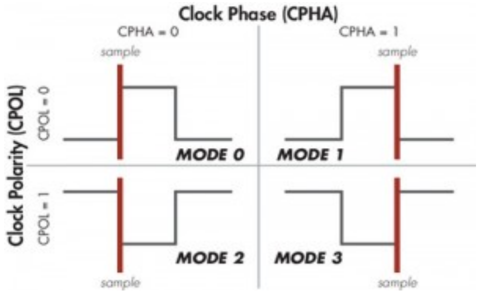


Figure 24 SPI Modes

The data is set up by Lepton on the falling edge of SCK and should be sampled by the host on the rising edge. The data transfer is the most significant byte first, in big-endian order. The CPOL parameter controls the steady state value of the clock when no data is being transferred. This parameter affects both master and slave modes. If CPOL is low, the SCK pin has a low-level idle state. If CPOL is high, the SCK pin has a high-level idle state. The CPHA parameter is configured as 2EDGE (or 1), the second edge on the SCK pin (rising edge if CPOL is high, falling edge if CPOL is low) is the Mbs capture strobe. Data are latched on the occurrence of the second clock transition. Data output one half-cycle before the first falling edge of SCK and on subsequent rising edges. Input data is latched on the falling edge of SCK. The table below summarizes the SPI configuration for the camera:

Table 9 SPI Specifications

Master Clock	216 MHz
APB2 Clock (peripheral clock	108 MHz
SPI Pre-scaler	8
SPI Clock Rate	13.5 MHz
SPI Mode	Full-Duplex Master
Clock Polarity	High (1)
Clock Phase	High (1)

Lastly, the I2C1 channel is configured as Fast Mode, at 400 KHz frequency. Two GPIOs PB6 and PB9 were configured as HIGH, for clock line (I2C1_SCL) and data line (I2C1_SCDA) respectively.

TCP/IP Ethernet Configuration Requirements

A lightweight IP TCP/IP middleware stack has been chosen for the Ethernet protocol. The Dynamic Host Configuration Protocol (DHCP) is disabled to hardwire the IP address for the network, and the board itself. The following IP were assigned for the TCP/IP:

- IP ADDRESS IP: 192.168.001.066
- NETWORK ADDRESS: 255.255.255.000
- GATEWAY_ADDRESS: 192.168.001.001

5.3 Thermal Camera Implementation

The camera core comes with a pre-configured calibration from the factory. These settings can be adjusted or changed through the CCI/TWI (or I^2C) interface. In this application, the settings are the following (default):

Flat Field Correction (FFC) State: Despite the output image being highly uniform from the factory calibration, the drift effects over long periods of time degrade uniformity, resulting in an imagery which appears grainier and blotchy. In factory settings, the camera is set to automatic

FFC, performing it during startup, after 3 minutes (default) has elapsed since the last FFC, and if the camera temperature has changed by more than 1.5 degrees Celsius since the last FFC.

Gain State: It is configured to operate in a high-gain state, which provides lower noise equivalent differential temperature and lower intra-scene range.

Telemetry Mode: Telemetry disabled by default. There are 60 video packets per frame and the packet length is 164 bytes.

Radiometry Mode: Radiometry enabled, TLinear enabled. The radiometry mode affects the transfer function between the incident flux and pixel output. For applications in which temperature measurement is required, radiometry must be enabled to access the related calibration and software features, such as TLinear. Radiometry mode with TLinear enabled changes the pixel output from representing scene flux in 14-bit digital counts to representing scene temperature values in Kelvin. With TLinear mode enabled with a resolution of 0.01 °C, a pixel value of 30000 signifies that the pixel is measuring 26.85 °C. The radiometric accuracy over the operational temperature range is within ±5°C. Additionally, the radiometry enabled mode performs internal signal level adjustments to ensure that the output data is independent of the camera's own temperature.

Automatic Gain Control (AGC) Mode: AGC is a dynamic range adjustment of the infrared sensor data to satisfy a particular display system, which by default is left disabled. This is appropriate for dealing with 14-bit raw data, which is the output video format selected for this application.

Video Output Format: Raw 14 (default). This mode is appropriate for viewing 14-bit raw data, where the first two bits of each pixel's two-byte word are set to 0.

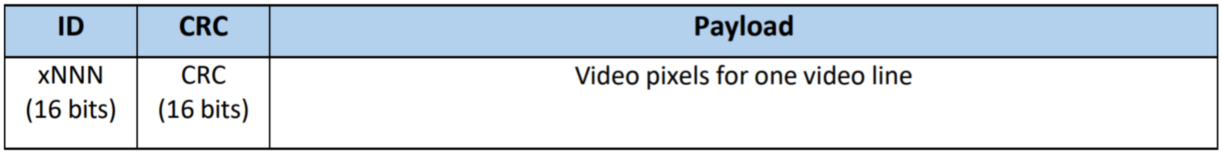
SPI Protocol

Packet: The protocol is based on a single standardized packet, which is the minimum "transaction" between the master and slave. Each video packet contains data for a single video line or telemetry line. Additionally, the SPI protocol of the camera core includes discard packets that are provided when no video packets are available.

Frame: A frame is defined as a continuous sequence of packets consisting of full frame's worth of pixel data

Stream: A stream is defined as continuous sequence of frames.

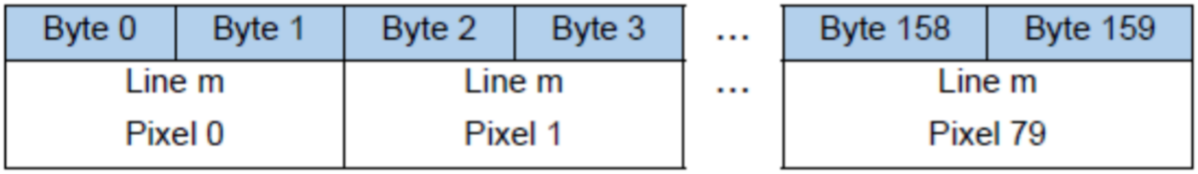
Each packet contains a 4-byte header (2-byte ID and 2-byte CRC, which are set to 0 in this particular configuration) followed by a 160-byte payload. The ID field is a 12-bit packet number because the leading 4 bits are reserved and are not part of the packet number. The packet numbering restarts at zero on each new frame. This helps to count the video lines received by the camera core. The CRC is calculated over the entire packet, however there is no requirement for the host to verify the CRC. The video packet is shown below:



ID	CRC	Payload
xNNN (16 bits)	CRC (16 bits)	Video pixels for one video line

Figure 25 Video Packet

The camera core transmits discard packets until it has a new frame from its imaging pipeline. These packets can be discarded by checking the ID field, which is always xFxx. For Raw14 Mode, the payload is illustrated as the following:



Byte 0	Byte 1	Byte 2	Byte 3	…	Byte 158	Byte 159
Line m Pixel 0		Line m Pixel 1		…	Line m Pixel 79	

Figure 26 One Video Line per 160-Byte Payload

The frame rate of the stream of packets is just below 27 Hz. However, the rate of unique frames is just below 9Hz to comply with US export restrictions. For each unique 80x60 frame, there are two duplicates that follow the stream. There is an internal 32-bit counter provided in the telemetry lines to check for unique frames, however this was not necessary in this stage of the prototype.

Thermal Image Acquisition Algorithm

The algorithm starts by pulling low the CS pin to enable the SPI transfer. Then, the line is read from the SPI bus at 13.5 MHz speed. Each video line's ID field is examined to identify discard packets and determine the packet or line number.

The line number is stored in order to compare it with the next line to ensure proper synchronization. Every consecutive video line is checked and copied to a buffer, which is responsible of stacking up a 2-D array to form a frame. Once, all 60 rows are captured and stored, the frame is stored in a First-In-First-Out (FIFO) data structure, and the line number is reset back to zero to start sampling the next frame. After the FIFO is filled, a lightweight TCP/IP protocol is used for sending the frames over Ethernet to the router. The figure below shows the data flow from the camera to the router:

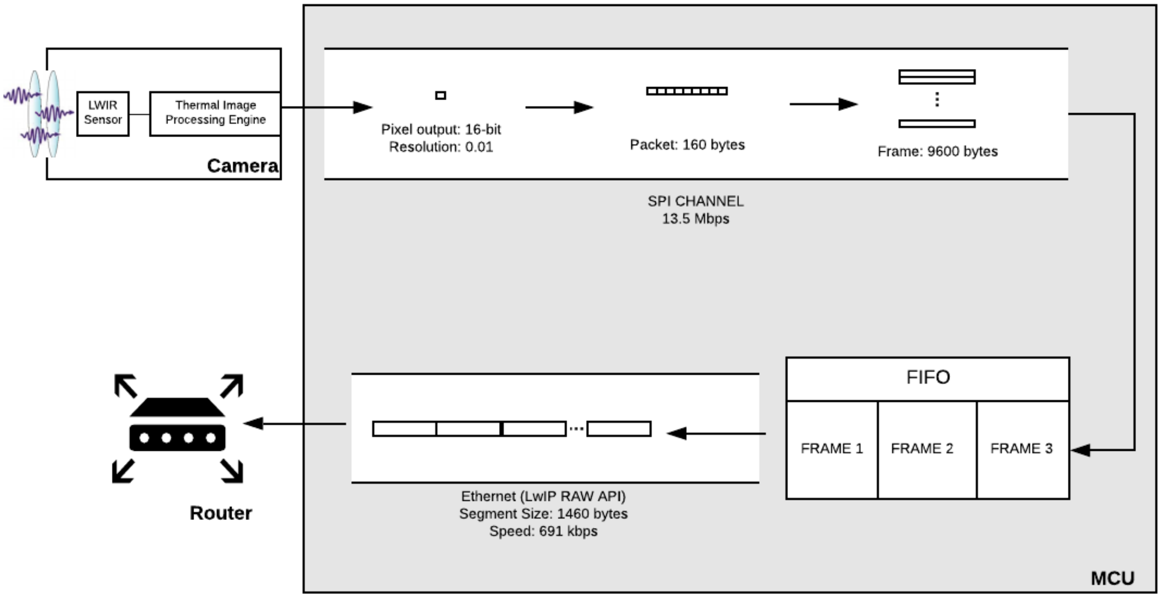


Figure 27 Data Flow

There are three main violations that can result in a loss of synchronization:

- Once a packet starts, it must be completely clocked out within 3-line periods.
- Provided that SPI clock rate is appropriately selected, and that /CS is not de-asserted during the packet transfer, an intra-packet timeout is an unexpected event.
- Failing to read out all packets for a given frame before the next frame is available.
- Failing to read out all available frames.

Thus, the CS is de-asserted and idle SCK for at least five frame periods, equivalent to 185 ms to satisfy the camera specification. This step ensures a timeout of the SPI interface, which puts the camera in the proper state to re-establish synchronization.

5.4 Ethernet Protocol Communication

LwIP TCP/IP stack has been used for the Ethernet communication from the microcontroller to the router. LwIP TCP/IP implementation is used to reduce RAM usage while keeping a full-scale TCP/IP stack, which makes it suitable for use in embedded systems. LwIP has three application programming interfaces (APIs), out of which Raw API is used for this project.

Raw API enables the development of applications using event callbacks. It provides the best performance and optimized code size but adds some complexity to the development of the application. When initializing the application, it is necessary to register callback functions to different events (such as TCP_sent, TCP_error…). The callback functions are called from the LwIP core layer when the corresponding event occurs.

For the first stage of the prototype, the standalone mode operation model has been used, where the software is continuously polling to check if a packet has been received. When the packet is received, it is copied from the Ethernet driver buffers into the LwIP buffers. Once copied, it is handled to the LwIP stack for processing. The operation model is shown below:

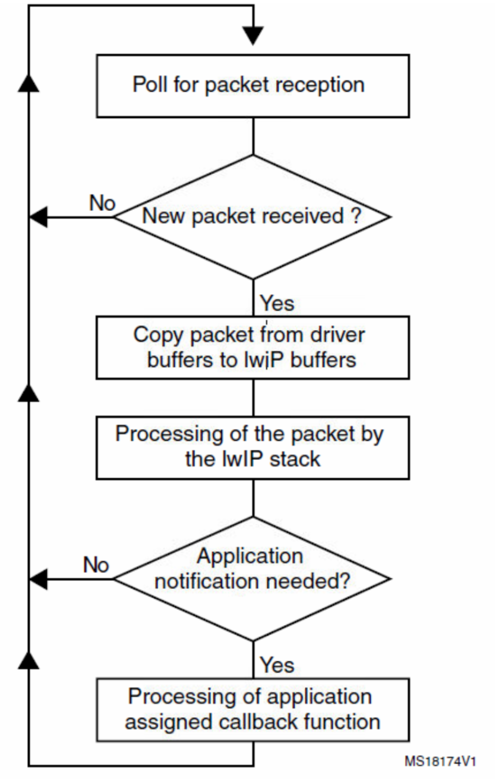

Poll for packet reception
No
New packet received ?
Yes
Copy packet from driver buffers to lwiP buffers
Processing of the packet by the lwIP stack
No
Application notification needed?
Yes
Processing of application assigned callback function
MS18174V1

Figure 28 LwIP Standalone Operation Model

For the TCP/IP protocol, the following common callback functions are assigned:

- Callback for incoming TCP connection event, assigned by TCP_accept API call
- Callback for incoming TCP data packet event, assigned by TCP_recev API call
- Callback for signaling successful data transmission, assigned by TCP_sent API call
- Callback for signaling TCP error (after a TCP abort event), assigned by TCP_err API call
- Periodic callback for polling the application (2 Hz), assigned by TCP_poll API call

It is important to call the TCP_send function in every callback to ensure the data or the frame in this case is forced to be sent. Additional configurations were implemented on the LwIP lwipots.h file, in order to support the streaming of the thermal video, which are shown below:

Table 10 lwipots.h Configuration

The size of the heap memory.	MEM_SIZE	(40*1024)
Maximum segment size	TCP_MSS	(1460)
TCP window size	TCP_WND	4 * TCP_MSS
Sender buffer space	TCP_SND_BUF	65535
The maximum number of bytes that TCP_write may allocate ahead of time	TCP_OVERSIZE	TCP_MSS

The above parameters played an important role in tuning the TCP/IP communication protocol to achieve a maximum throughput.

TCP_MSS: The maximum TCP segment size. Depending on the application, this value can be adjusted up to a maximum 1460 bytes.

MEM_SIZE: If the application will send a lot of data that needs to be copied, this should be set high depending on the memory specification of the computer.

TCP_WND: If the memory allows it. it is recommended that this parameter is set as high as possible. For every active connection, the full window must be buffered until it is acknowledged by the remote side, affecting the overall transmission time.

TCP_SND_BUF: It represent the sender buffer space in bytes. For maximum throughput, this value should be set as TCP_WND, however in this application the performance was better when the maximum value was used.

TCP_OVERSIZE: The maximum number of bytes that tcp_write can allocate ahead of time to create shorter pbufs(data structure for packets) chains for transmission.

A camera frame is equivalent to 9600 bytes and is transmitted at a rate of 8.6 Hz. The TCP/IP maximum segment size is 1460 bytes. Thus, it will split the data and send it piece-by-piece and not continue with the next frame until all segments are delivered. The 8.6 Hz frame rate means that there are 8.6 or 9 frames transmitted each second, equivalent to 691 kbps throughput.

5.5 PC Software

5.5.1 Backend

Data Stream

The backend is written in Python. Python module “socket” is used to establish network connection to the hardware and to transmit and receive data. To recapitulate the Ethernet streaming mentioned previously, OSIMS module continuously streams thermal images of size 80X60 (4800 pixels) at a frame rate of 9 fps over Ethernet with TCP/IP. The data is in the form of a byte stream. In a single image, there are 4800 pixels with each pixel containing a two-byte integer representing its temperature, therefore, each image contains 9600 bytes of data.

The first step is to retrieve the data, in order to do so, OSIMS software establishes a Socket connection to the OSIMS hardware via IP address: 192.168.001.066. Then the software continuously receives data in the form of a binary buffer. The bytes in the buffer are then converted to a list of floats representing temperatures of each pixel within a frame as shown in Figure 29:

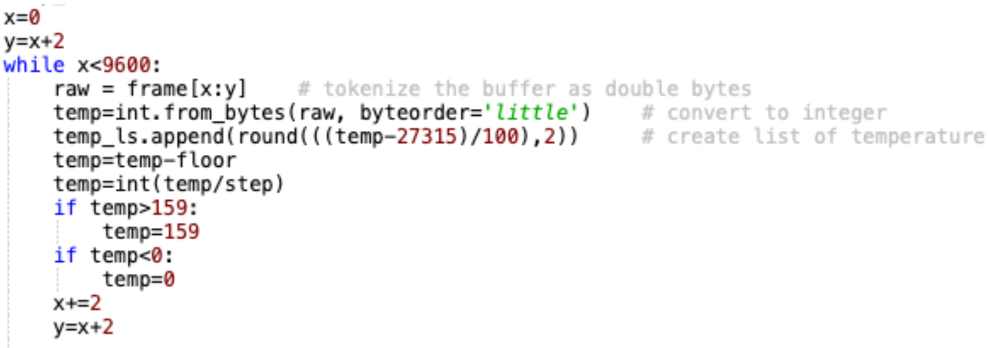
- The 9600 bytes buffer is tokenized to 4800 double-bytes, each double-byte contains temperature of one pixel in little-endian byte order.
- Each of the 4800 double-byte is then converted to integer with little-endian.

- Each double-byte integer represents temperature in Kelvin multiplied by 100. To convert to temperature in Celsius, we must subtract 27315 from it and divide by 100:

$$Integer = K*100 \qquad K = C + 273.15 \qquad C = (Integer - 27315)/100$$

- Then, the 4800 integers are rearranged as a list of temperatures.

The steps listed above are repeated nine times per second for live temperature streaming.



```
x=0
y=x+2
while x<9600:
    raw = frame[x:y]    # tokenize the buffer as double bytes
    temp=int.from_bytes(raw, byteorder='little')    # convert to integer
    temp_ls.append(round(((temp-27315)/100),2))     # create list of temperature
    temp=temp-floor
    temp=int(temp/step)
    if temp>159:
        temp=159
    if temp<0:
        temp=0
    x+=2
    y=x+2
```

Figure 29 Binary Buffer to Float List Conversion

Image and Video Streaming

The list of temperatures is then used to construct the infrared image. Intuitively, a higher temperature pixel is assigned to a brighter color, whereas a lower temperature pixel is assigned to a colder color. OSIMS uses the palette shown in Figure 30 to convert temperature into infrared images. The palette has 160 steps that starts from dark purple (cold) to glowing yellow (hot).

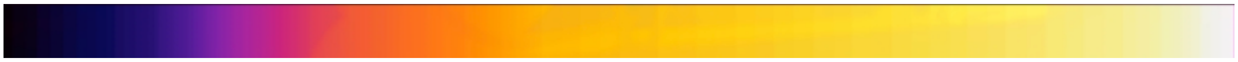
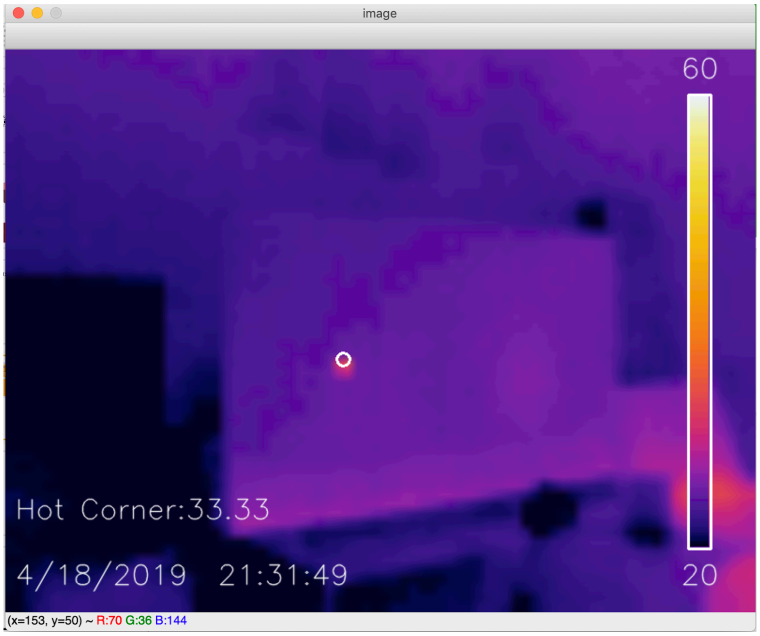


Figure 30 Infrared Image Color Palette

The range of temperature determines the temperature that can be visualized, in addition it also determines the visual resolution, which is the smallest temperature variation that can be reflected by the palette. The tradeoff is that larger temperature range will give a lower resolution, on the contrary a smaller range sacrifices the maximum and minimum temperature There are three ways to configure the temperature range: “Default”, “Auto” and “Manual”. By default, OSIMS will set the range to (20°C ~ 60°C), this gives users an easy configuration to optimize OSIMS for measuring objects in the ambient temperature range, such as human body (approx. 35°C), electric motor(approx. 45°C). “Auto” configuration allows the finest temperature resolution in a given frame. If configured as “Auto”, OSIMS will first obtain a temperature array, determines the hottest and coldest temperatures within the array, then matches the 160-sized palette to reflect temperatures within that range. “Manual” configuration awaits for user’s inputs. This gives user the advantage of manually configuring OSIMS for their application.

The backend IR imaging and video streaming functionalities are implemented with OpenCV for Python. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library [15]. It has libraries for opening image, converting to video and image processing. The array of temperatures is turned into a rgb image and displayed with the OpenCV image module. The image is then updated at 9 fps and converted into avi format video for archiving using OpenCV videowrite module.

Figure 31 shows an IR image of an ambient room with “Default” configuration. The temperature range of 20°C to 60°C can efficiently reflect the heat distribution of objects at ambient temperatures. Note that the hot spot identified at the center of the image is actually a result of thermal reflection of a hot object on the surface of the TV screen.



Hot Corner:33.33

4/18/2019 21:31:49

(x=153, y=50) ~ R:70 G:36 B:144

Figure 31 Infrared Image by OSIMS

Continuous System Integrity Monitoring

OSIMS software has three features to monitor system integrity:

- Hot Corner Plot
- Pixel Plot
- Warning

Hot corner plot identifies the hottest temperature inside the frame and plot the temperature continuously in time domain. This functionality is achieved by measuring the maximum temperature of each frame. The process is repeated every time the image gets updated to create a list of hot corner temperatures. Figure 32 shows the interface of the Hot Corner Plot. This feature allows the user to monitor the hot corner behavior of the targeted system in real time.

Pixel plot has features similar to those of the hot corner plot. It plots temperatures of a fixed pixel over time. This feature gives user accessibility to a certain point of interest, other than only knowing the hottest temperature inside the frame. This feature is necessary in the application of multiple points monitoring, meaning if more than one corner in the frame are to be measured, OSIMS software can add one or multiple pixel plots to keep track of all points of interest and plot their temperatures vs time in separate windows. Figure 33 shows an example corner plot window at pixel (14,50).

Warning: OSIMS software has warning feature to setup one temperature threshold. If any pixel inside the frame exceeds the threshold, the alarm will be triggered to alert that temperature has exceeded the critical value.

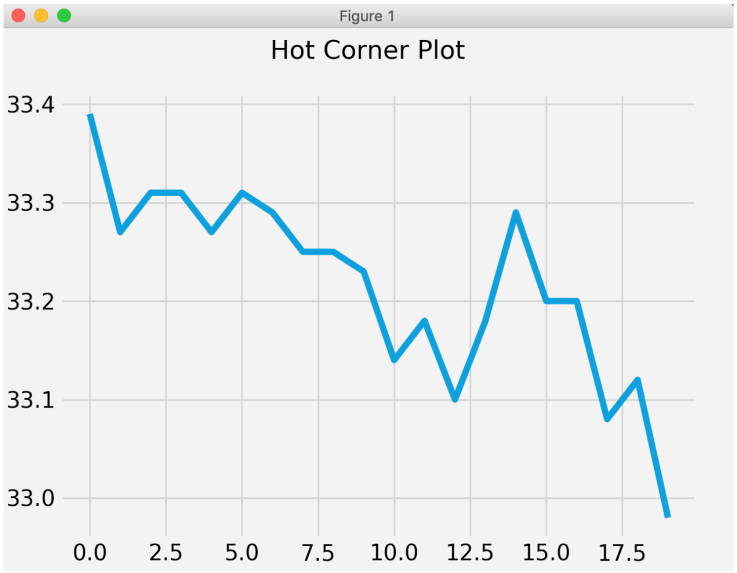


Figure 32 Hot Corner Continuous Plot

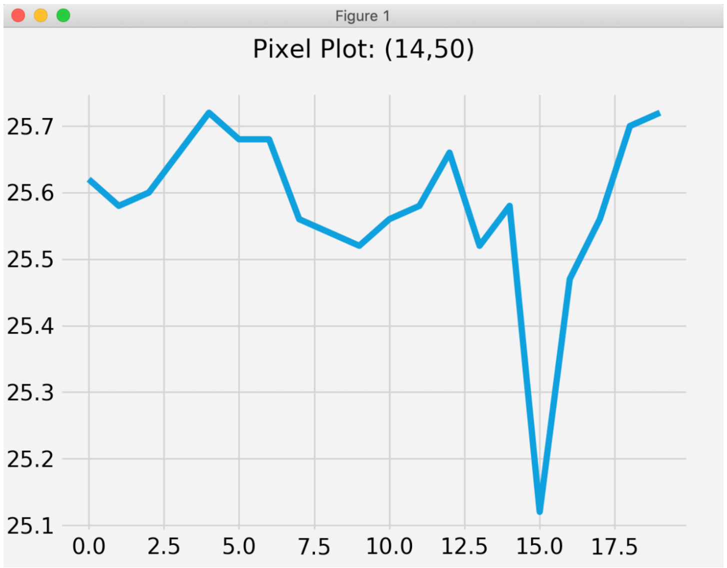


Figure 33 Pixel Continuous Plot

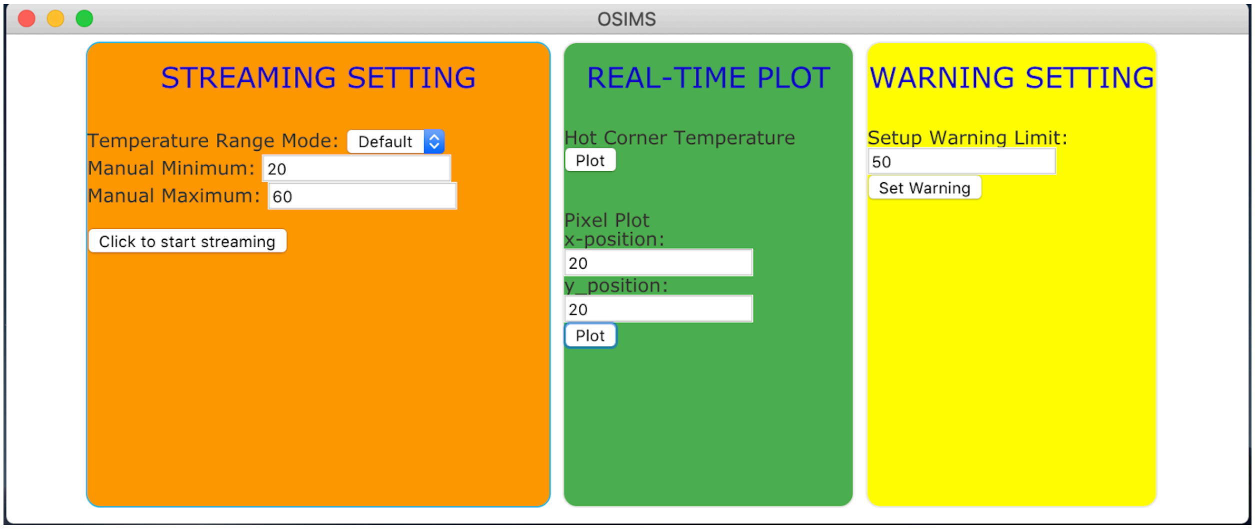
Data Archive: OSIMS stores temperatures of the frame at every second in the form of csv files. This data can be used as training data for machine learning in the future. It is also useful for post-accident investigation. This is done by creating a fileout() function with Python library csv. At the end of the loop after the temperature list of one single frame has been received, this function gets called, writing data to the designated file and, only then, proceeds to the top of the loop.

5.5.2 Frontend

User Interface

Electron is a wrapper of html/CSS and JavaScript. That means, Electron provides a cross-platform engine to display a GUI written as an html webpage. Html is used to define the layout of the webpage, and the contents within. The layout means a webpage is divided into blocks for better structuring the elements within the GUI, including the positioning and sizing of the blocks. CSS is a language describing the style of html. CSS is used to set the fonts, colors, alignment and background colors of the html elements. JavaScript can read and write to the fields and interact with the elements within an html webpage.

Figure 34 shows the main user interface of OSIMS software which has three sub windows: streaming setting, real-time plot and warning setting, each corresponds to the backend functionalities mentioned in previous chapter. The sizes and positioning of the three sub windows are defined by html. Contents including texts, input boxes, drop box and buttons are elements of html. The vivid colors, fonts and alignments are defined by CSS. Last but not least, the input boxes, drop boxes and buttons can be directly accessed by JavaScript.



Temperature Range Mode: Default
Manual Minimum: 20
Manual Maximum: 60

x-position: 20
y_position: 20

Setup Warning Limit: 50

Figure 34 OSIMS Software Main User Interface

Frontend to Backend Structure

Figure 35 shows the block diagram of how the software frontend interacts with the Python backend. As mentioned previously, the html button will trigger a JavaScript function, and JS can also read data from html directly. JavaScript can then connect to a Python backend via an NPM module "Python-Shell", a module of JavaScript that acts as a wrapper to allow interaction with a Python script.

Figure 36 shows the elements in html that receive user input. The button underlined in red initiates the JavaScript linker function "streaming()".

Figure 37 shows the JavaScript linker function "streaming()". It retrieves the values of the elements inside the HTML, packs the values and send the package of data as system arguments to Python.

The Python script can read the user inputs from the package data inside the sys. To do so:

```
import sys
args = sys.argv
val_1 = args[1]                # args[0] reserved for system directory
val_2 = int(args[2])
val_3 = int(args[3])
```

Note that in Figure 35, it is easy for JavaScript to send data to Python through standard system arguments (sys.stdin / sys.args), however it is difficult (not impossible, but difficult) to send data in the other direction. Therefore, a system data buffer was created to solve this problem. This binary file is only written by Python and read by JS [16].

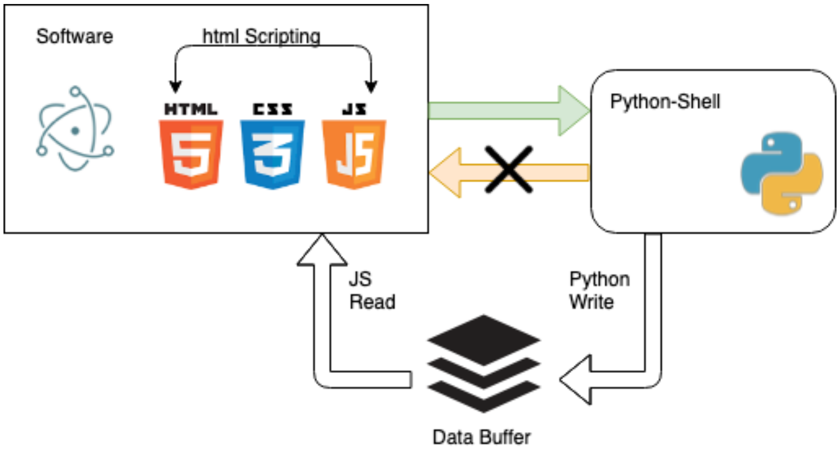
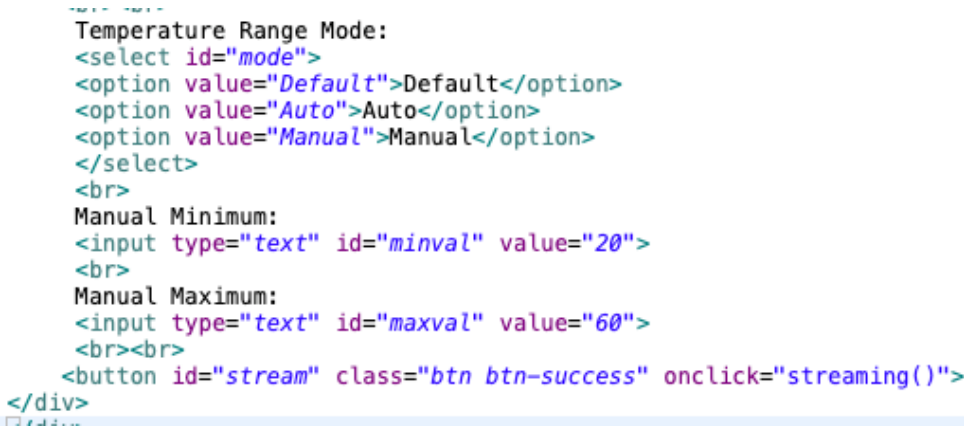
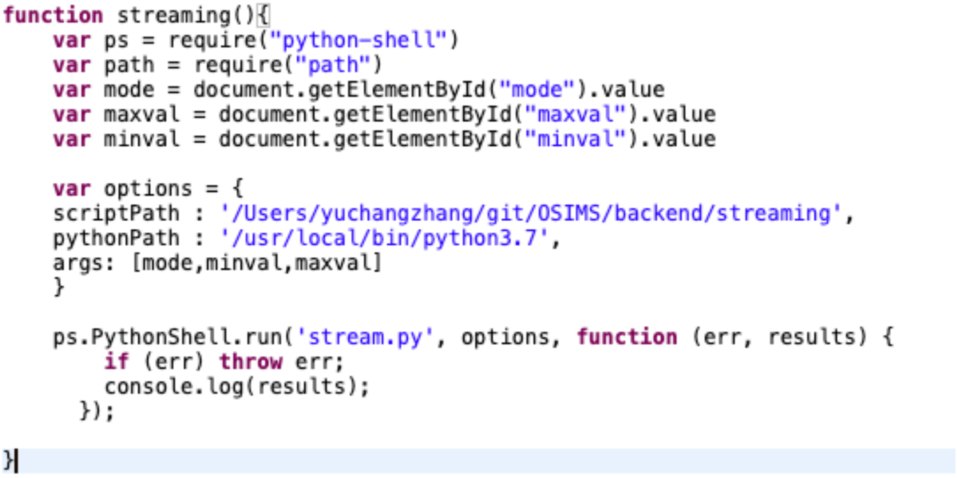


Figure 35 Frontend and Backend Block Diagram



```
     Temperature Range Mode:
     <select id="mode">
     <option value="Default">Default</option>
     <option value="Auto">Auto</option>
     <option value="Manual">Manual</option>
     </select>
     <br>
     Manual Minimum:
     <input type="text" id="minval" value="20">
     <br>
     Manual Maximum:
     <input type="text" id="maxval" value="60">
     <br><br>
    <button id="stream" class="btn btn-success" onclick="streaming()">
</div>
```

Figure 36 HTML User Input Elements



```
function streaming(){
    var ps = require("python-shell")
    var path = require("path")
    var mode = document.getElementById("mode").value
    var maxval = document.getElementById("maxval").value
    var minval = document.getElementById("minval").value

    var options = {
    scriptPath : '/Users/yuchangzhang/git/OSIMS/backend/streaming',
    pythonPath : '/usr/local/bin/python3.7',
    args: [mode,minval,maxval]
    }

    ps.PythonShell.run('stream.py', options, function (err, results) {
        if (err) throw err;
        console.log(results);
      });

}
```

Figure 37 JavaScript Linker Function

Communication Between Python Processes

As discussed in section 6.4.1, the steaming and real-time plots are separate Python scripts and therefore separate Python processes. This creates a problem because the separate Python processes are completely isolated from each other. Hence, it is not possible to send and receive data between them.

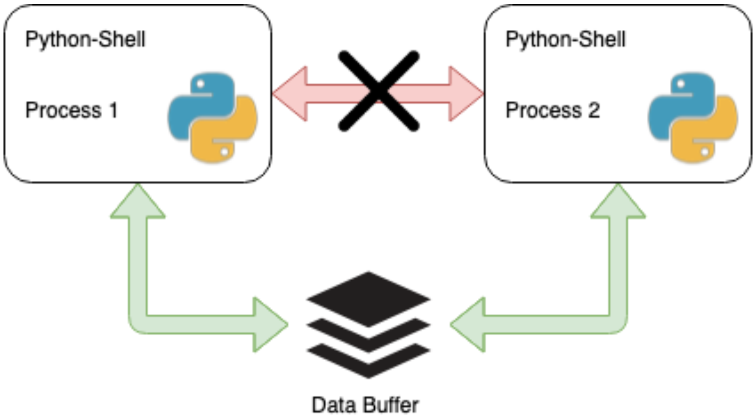


Figure 38 Data Path between Python Processes

To get around this problem, a data buffer solution was introduced that can be accessed and manipulated by multiple Python processes at the same time. However, it is a rather temporary solution that needs to be readdressed in the next stage. For example, if Python process 1 is the video streaming, whereas process 2 is the hot corner plot, process 1 will convert buffers received from Ethernet to a list of temperatures and stream live video, however the image and temperature data are only available to itself. Process 2 wants to access the list of temperatures in order to perform the hot-corner plot. Process 1 opens a file and continuously writes a real-time data log of the newest temperature frames to that file. Process 2 opens the same file for read-only access to prevent conflicts. Then, process 2 is able to retrieve the same temperature data for real-time plotting. This process can also be reversed in direction if needed.

5.6 Hardware Integration

5.6.1 Expansion Printed Circuit Board

In order to support the connectivity between the microcontroller and the thermal breakout board, an expansion PCB was designed for signal and power routing among the modules. The PCB is a two-layer board, where the first layer contains the signal routings and connectors, while the bottom layer is a ground plane. A 12 V, 1A power adapter is used to power the board through a barrel connecter. The end-to-end connections are shown below:

Table 11 List of Power and Signals between Microcontroller and Camera Breakout Board

	Microcontroller		Thermal Camera Breakout Board	
	Signal	Pin	Pin	Signal
Power	5 V	CN8-9	J2-6	3V-5V
	Ground	CN8-11	J2-5	Ground
I2C	I2C1_SDA	CN7-4	J2-7	SDA
	I2C1_SCL	CN10-13	J2-8	SCL
SPI	SPI1_SSEL	CN7-17	J2-1	SPI_CS
	SPI1_SCLK	CN7-10	J2-4	SPI_CLK
	SPI1_MISO	CN7-12	J2-3	SPI_MISO
	SPI1_MOSI	CN7-14	J2-2	SPI_MOSI

The microcontroller evaluation board as well as the thermal camera breakout board are placed on top of the expansion board, enabling direct connection to the designated pins. The schematic, PCB and 3D view of the expansion board is shown in following Figures.

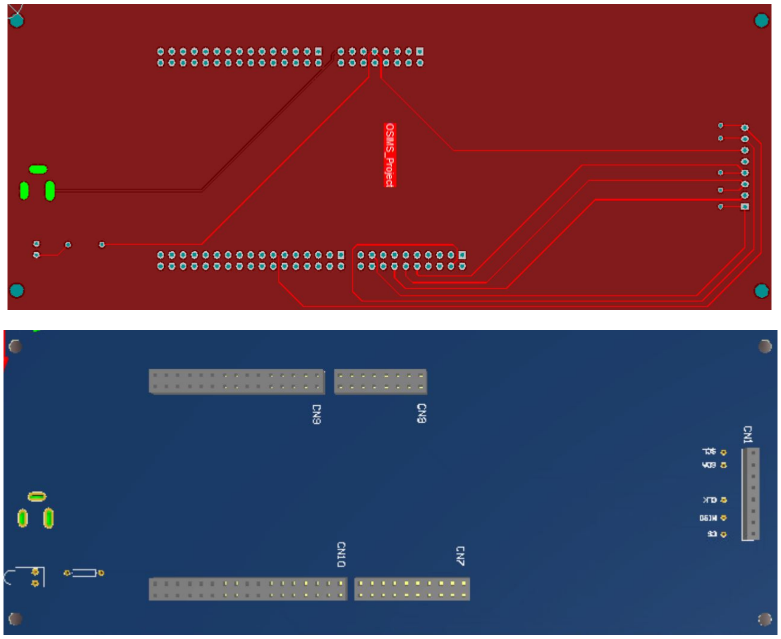


Figure 39 OSIMS Expansion PCB and 3D View

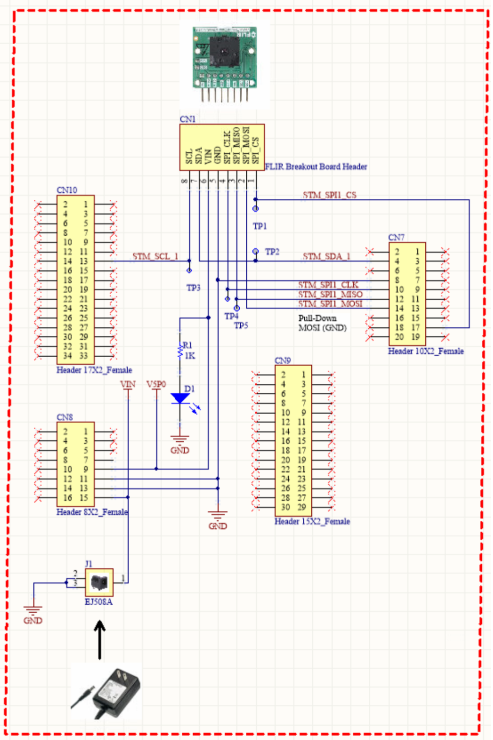


Figure 40 OSIMS Expansion PCB Schematic

5.6.2 Case

The prototype case consists of an Onyx 3D printed material, which is based on a tough nylon, but also provides parts with stiffness greater than any pure thermoplastic material available for professional 3D printers. The case isolates the internal circuitry from the outside environment and hosts the germanium window from the inside. The case dimensions are 177 mm x 77 mm x51mm. The dimensions to support the evaluation board that was used in the prototype stage. However, in the future, when all components are integrated within one board, the size will be significantly smaller.

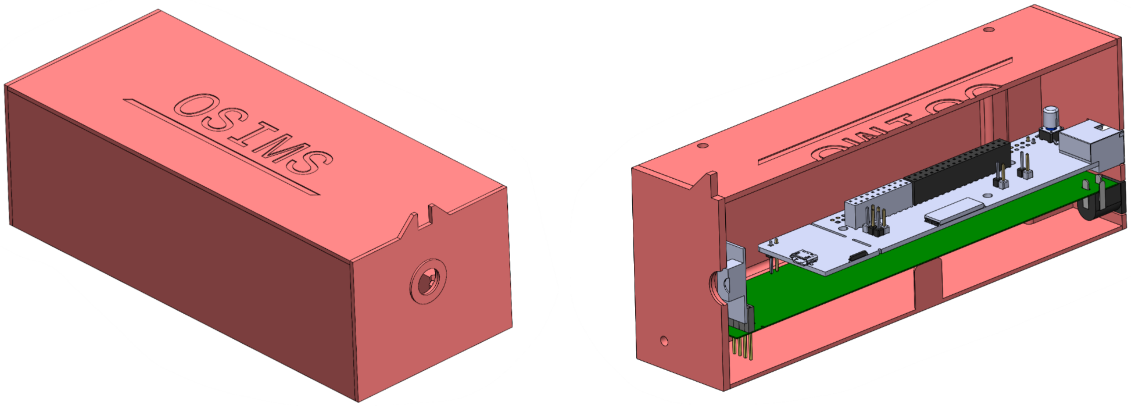


Figure 41 OSIMS 3D Model

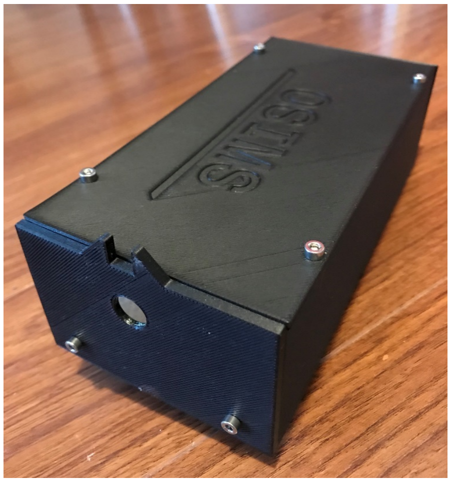


Figure 42 OSIMS 3D Case

5.7 System Application

On a system level, OSISMS will be placed at certain locations to monitor several machineries or equipment of a specific zone. All these will be connected through a local router over Ethernet, on a private network. The software will allow the operator to continuously monitor all units simultaneously and be cautious regarding any risen temperature that can potentially affect the integrity of the system. The system implementation is shown below:

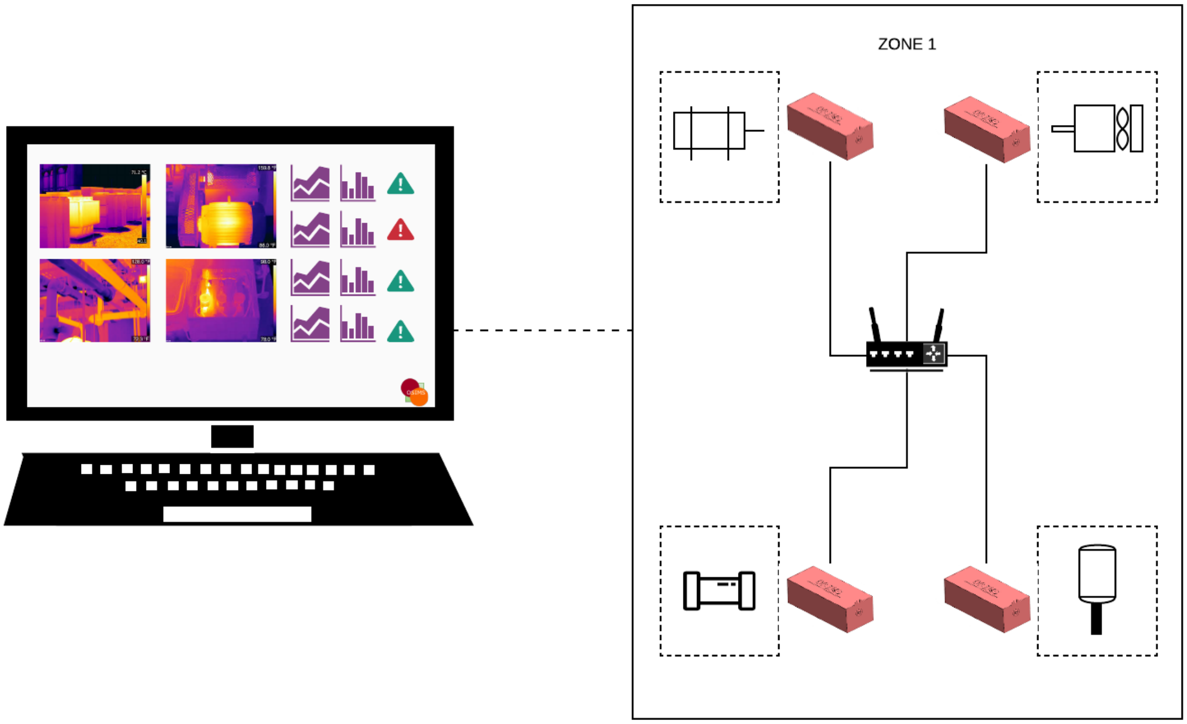


Figure 43 OSIMS System Level Implementation

6 TESTING AND RESULTS

This chapter discusses testing and results of OSIMS hardware prototype and OSIMS alpha software. The tests listed below are carried out to verify the product specification for the prototype, as well as to prove software functionalities:

- Thermal Camera Streaming Verification
- Temperature Precision and Resolution
- Calibration
- Thermally Reflective Surface Measurement
- Multiple Corner Monitoring

The test bench consists of three heating elements mounted on a wood surface, as shown in Figure 44. The top left is an electric heater; top right is a simulation of an electric motor; whereas bottom right simulates a pipeline with internal heat. Wood was chosen to be the bottom panel for the test bench, because wood is a good thermal insulator, additionally, it is not thermally reflective. Therefore, it prevents heat interference among heating elements and does not reflect the thermal image of the surroundings.

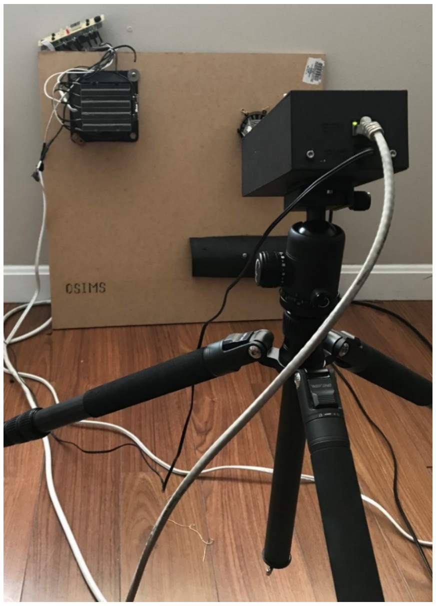


Figure 44 OSIMS Test bench Setup

6.1 Thermal Camera Streaming Verification

The microcontroller enables the power to the thermal camera as well as establishes the medium of communication among each other. The thermal camera utilizes the SPI protocol to output the video data. The specification of the camera states that the frame rate is normally just below 27 Hz. However, the rate of unique frames is just below 9 Hz (8.6 Hz) to comply with US export restrictions. The microcontroller initiates all transaction by pulling low the CS pin and controls the clock speed. One frame video data takes 1/27 seconds, which is equivalent to a clock frequency of 2.2 MHz. The maximum tolerable clock rate by the camera is 20 MHz, therefore a 13.5 MHz speed has been configured for this application. The clock rate was confirmed via oscilloscope in Figure 45.

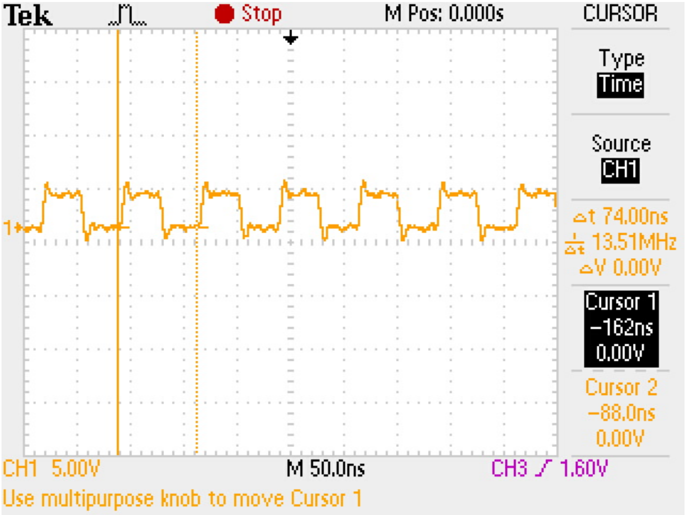


Figure 45 SPI Clock Rate

The minimum clock rate is a function of the number of bits of data per frame that needs to be retrieved. Having a clock rate of 13.5 MHz results in a 74 nS period. The SPI protocol is configured with a 16-bit data transfer between the master and the slave. So, it takes 74 nS to sample each bit in the transmission or 1.184µS for 16 bits (2 bytes), as shown below.

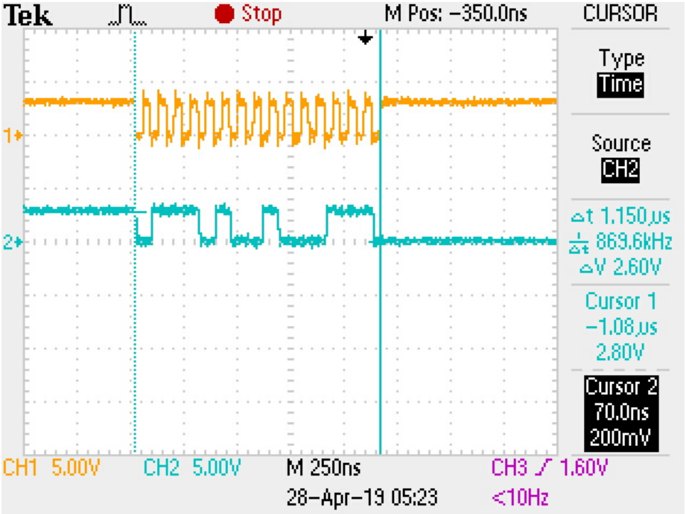


Figure 46 SPI Sampling 16 bits

There are 60 video packets for each frame, where one video line is 164 bytes or 1312 bits. According to the datasheet, the throughput is approximately 25.9 frames per second. As mentioned previously, the video data from the camera is retrieved from SPI and stored in a 2-D array, which is after processed in a FIFO data structure. The TCP/IP LwIP protocol accesses the frame stored in the FIFO and handles the transmission over the Ethernet. It was expected that during this process, delays could be encountered because of the software polling approach.

To test the streaming capability, two LEDs were used to monitor the rate of the main loop execution as well as the time it took for the frame to be sent over Ethernet. This would help visualize the transmission as well as determine it by placing two probes on the anode side of the LEDs. A LOW-to-HIGH represent a full execution of that processes mentioned above. In Figure 47, channel 2 represent the number of frames sent per second, and channel 3 shows the amount of executions of the main loop.

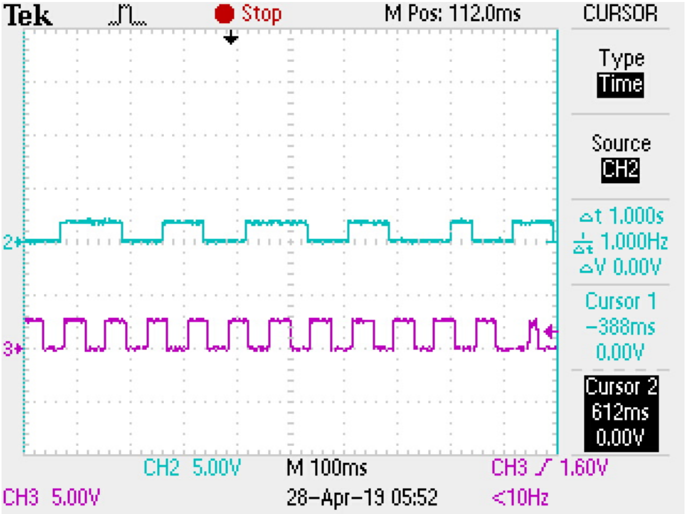


Figure 47 Frame Streaming Rate vs Main Loop Execution Rate

As shown above, the actual streaming is around 6 frames per second as opposed to 8.6 frames per second expected from the datasheet. It can be noticed that some of the HIGH states are not symmetrical, which indicates a delay in the transmission of the frame over the Ethernet. The delay is directly related to the throughput of the TCP/IP protocol, considering the amount of data transmitted and the buffering performed to enable such process, as well as the polling algorithm which is performed twice a second. However, this application requires the module to be stationary, and is not subject to any movement. Capturing solely 6 frames out of 8.6 is insignificant, but definitely an aspect for future improvement.

Table 12 Thermal Camera Test Verification Results

Specifications	Expected	Measured
SPI Clock Rate	13.5 MHz	13.51 MHz
SPI Period	74 nS	74 nS
Bit Sampling (min SPI transfer, 16 bits)	74 ns/bit = 1.184µS/16bits	1.184µS
Streaming Frame Rate	8.6 Hz	6Hz

6.2 Temperature Precision and Resolution

6.2.1 Standard Deviation

Precision is an important aspect of OSIMS, because in order to provide reliable monitoring of the system it depends on high precision in data measurements. An inconsistency in temperature readout can lead to misinterpretation of the safety measure of the system. For example, if the temperature suddenly rises above a significant margin, it will trip the system to give false alarm.

Test Method

To calculate the precision of OSIMS' measurements over time:

1. Selected a test object that has a low and steady temperature close to ambient. Therefore, the surface temperature should be fairly consistent over a small interval of time (for example 20 seconds).
2. Took multiple temperature measurements at an equally spaced time interval. Here, eighteen data points were taken at a one second interval.
3. Calculated standard deviation of data set.
4. Calculated maximum deviation of data set.

Figure 48 shows OSIMS' measurements of test unit one on the test bench described previously. For this test purpose, test unit one was idle and had a near ambient temperature of approximately 27.3 degrees. Eighteen data points were measured in an 18 second interval.

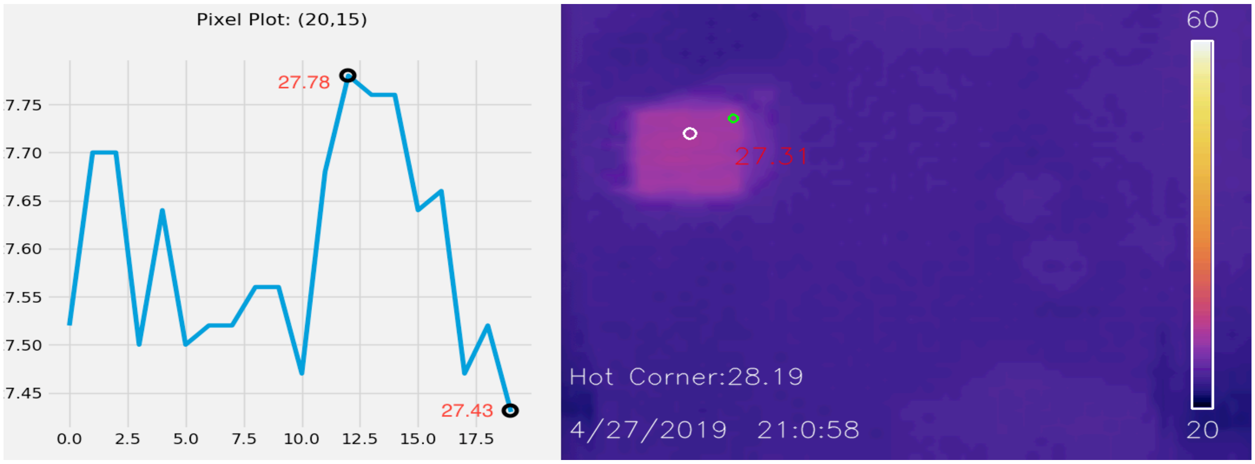


Figure 48 Temperature Deviation of Fixed Temperature Object

Table 8 shows 18 data points of test unit one measured by OSIMS. The equation for standard deviation is shown below. First, calculate the mean of the data set:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_x)^2}{n-1}} \qquad \mu_x = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\mu = (27.52 + 27.7 + \ldots + 27.43) / 18 \approx 27.603$$

Then substitute µ = 27.603 and N = 18 into the equation and obtain:

Standard Deviation (SD) = 0.11 °C

Calculating the maximum deviation helps to understand the worst-case precision, which is the maximum offset from average values in the data set:

Maximum Deviation (MD) = (27.78 – 27.43) / 2 = 0.175 °C

Table 13 Standard Deviation Data Set

Data Point	Value
0	27.52
1	27.7
2	27.7
3	27.5
4	27.64
5	27.5
6	27.52
7	27.62
8	27.46
9	27.68
10	27.78
11	27.76
12	27.76
13	27.64
14	27.66
15	27.47
16	27.52
17	27.43

6.2.2 Measurement Resolution

The measurement resolution is the minimum variation in temperature that can be realized by OSIMS. In other words, it reflects the minimum ΔT per Δt. Therefore, resolution is proportional to the sampling rate. OSIMS samples temperature at a rate of once per second, which means the resolution is in the unit of [°C/step]. Very importantly, note that the precision can also affect the resolution, because the deviation in precision is similar to the concept of "noise". If the resolution is smaller than the standard deviation, it means that the measurement is corrupted by noise.

Test Method

To verify the measurement resolution:

1. First, test unit one was turned on until it reached 100 °C and then turned off to let it cool. The test unit showed an approximately linear decay in temperature.
2. Recorded various data points along the temperature decay.
3. Calculated total decay, and then divided by time steps to determine average step size.
4. Compare step size with standard deviation. If step size is smaller than the standard deviation, then the resolution has the value of the standard deviation.

Figure 49 shows OSIMS measuring the temperature of a test unit one which has a steady linear decay over time, and also plotting the hot corner vs time with a sampling rate of 1 Hz.

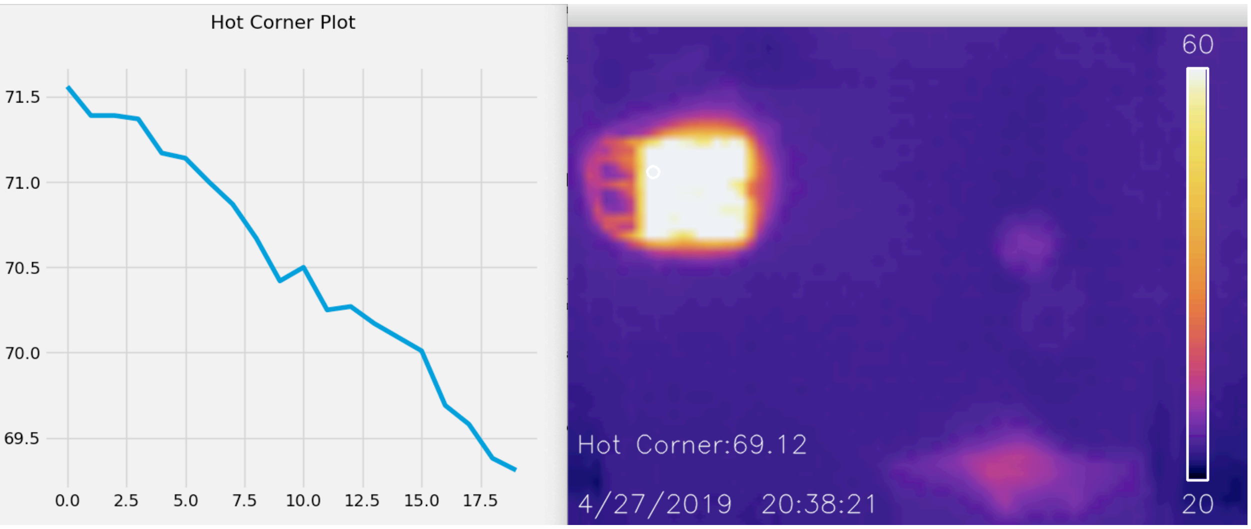


Figure 49 Linear Temperature Decay

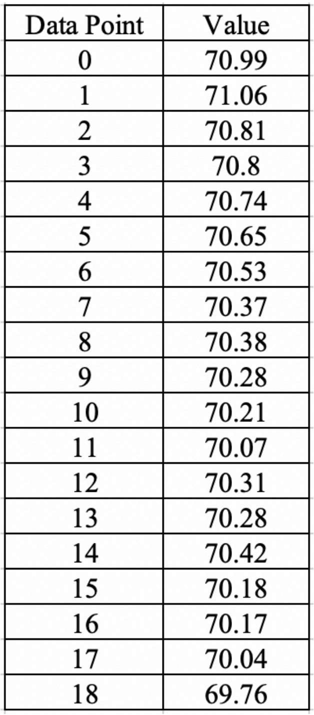
Table 12 shows 18 data points 18 seconds interval, indicating temperature dropping from 70.99 °C to 69.76 °C over 18 seconds:

Average step size = (70.99 – 69.76) / 18 = 0.06 [°C/s]

Average step size: 0.06 [°C/step] < Precision: +/- 0.11 [°C]

The calculated step size is smaller compared to standard deviation. It is impossible to have a fine resolution that is smaller than the noise amplitude. Therefore, the temperature resolution of OSIMS has the value of the standard deviation: 0.11 [°C/step].

Table 14 Resolution Data Set



Data Point	Value
0	70.99
1	71.06
2	70.81
3	70.8
4	70.74
5	70.65
6	70.53
7	70.37
8	70.38
9	70.28
10	70.21
11	70.07
12	70.31
13	70.28
14	70.42
15	70.18
16	70.17
17	70.04
18	69.76

6.2.3 Precision and Resolution Results

Table 13 shows the results obtained from precision and resolution tests. The calculated standard deviation closely agrees with the precision specification of +/- 0.1 °C. The measurement resolution also has a value of 0.11 °C, which meets the design spec of 0.1 °C.

Table 15 Temperature Precision and Resolution Test Results

Specification	Expected	Measured
Precision	+/- 0.1 [°C]	+/- 0.11 [°C]
• Standard Deviation (SD)	N/A	0.11 [°C]
• Maximum Deviation (MD)	N/A	+/- 0.175 [°C]
Measurement Resolution	0.1 °C	0.11 [°C/step]
• Avg Step Size	N/A	0.06 [°C/step]
• Resolution	N/A	0.11 [°C/step] (SD)

6.3 Calibration

6.3.1 Window Effect

An important component in the design is the infrared window, which is designed to allow infrared radiation transmission. A germanium window has been chosen due to its good broadband infrared transmission range as well as its transmissivity properties. The purpose of this test is to study the effect of the window on the temperature readings. Three tests were performed on three different objects. The first test is done using a third-party software, MATLAB, where the same corner temperature is measured with and without the window. The results are shown below:

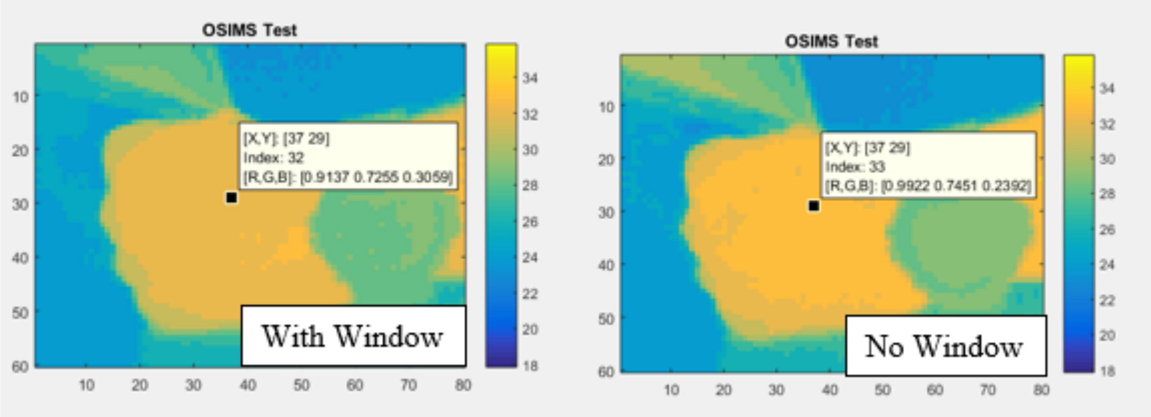


Figure 50 MATLAB Window Effect Readings, Human Hand

As seen above, the corner of interest is positioned at [X.Y] = [37,29]. The measurements are performed quickly considering the fluctuation of the human body temperature. MATLAB readings showed that there is a 1°C difference between having a window and no window. The MATLAB algorithm is performing a full rounding on the temperature readings, indicating that a reading of 32.5°C would result to a 33°C reading.

For the second and third tests, OSIMS software was used to conduct the measurements. The second object is an electric griddle where the temperature scale is set at a fixed level. The measurements were performed after the temperature reached a constant level as shown in Figure 51. The dip in the Figure was a distortion caused by the installation of the window. The temperature readings were not affected, staying consistent at 181.13°C.

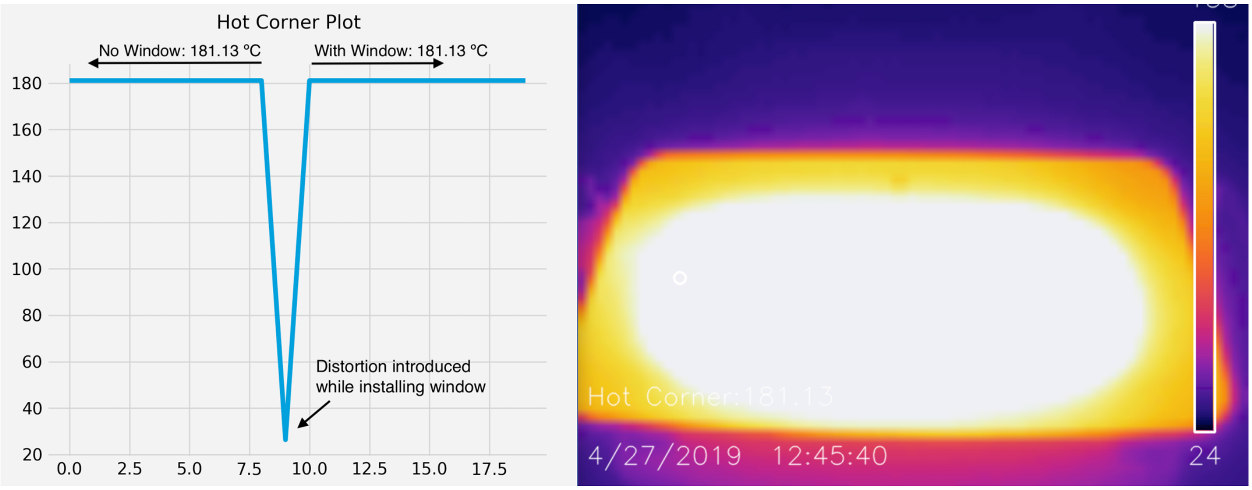


Figure 51 Window Effect, Electric Griddle

For the third test, a fixed corner on a human body was considered. Knowing that the human body fluctuates in temperature, the measurement was performed quickly. As shown below in the graph, the temperature offset introduced by the germanium window was 0.58 °C.

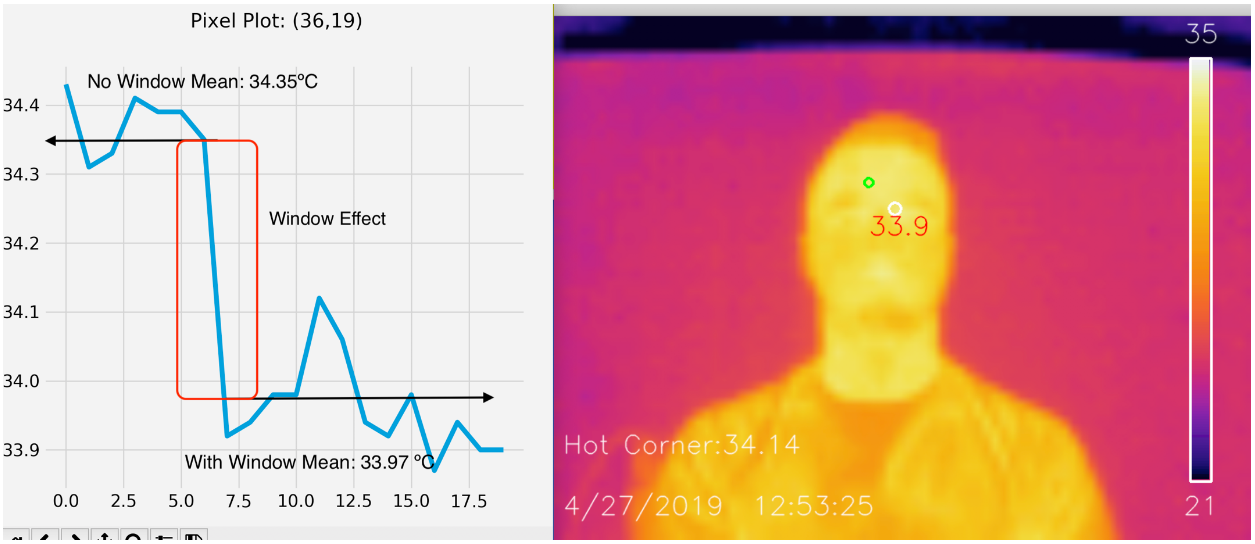


Figure 52 Window Effect Test, Human Body

6.3.2 Temperature Offset

The method used to calibrate the temperature offset is by comparing OSIMS' measurements to that of a commercial off-the-shelf (COTS) product, then calculating the difference temperature offset. An infrared thermometer was chosen, with a measurement range of -50 °C to 750 °C. It has a resolution of 0.1 °C and accuracy of +/- 2 °C.

Table 16 Infrared Thermometer Specification

Division	0.1 ℃ or 0.1°F
Accuracy	≥100 °C ± 2 % ≤100 °C ± 2 °C

Figure 53 Infrared Thermometer

To perform the temperature offset measurement, two objects were used to compare the difference in temperature measurements: a hot water boiler at approximately 80 °C, and human body. Figure 54 is the experiment with human body temperature. OSIMS measured 34.12 °C, whereas the COTS product measured 33.3 °C. In Figure 55 OSIMS measured temperature of the center point on the surface to be 80.85 °C, the COTS product measured 83.0 °C respectively. Two other objects which were not shown in the figures but also measured in the test were cold milk and ambient water mug.

In conclusion, the temperature readings of OSIMS were compared to a COTS product, and the difference between two measurements are shown in Table 15. Note that the differences are close to the listed accuracy of the COTS product of +/- 2 °C, which means the difference can be caused by the accuracy margin of COTS product. Therefore, the measured temperature offset of OSIMS is approximately zero.

Table 17 Temperature Offset Measurements

	OSIMS	COTS	Offset
Human Body	34.12 °C	33.3 °C	0.82 °C
80 °C Surface	80.85 °C	83.0 °C	-2.15 °C
Cold Milk	11.6 °C	10.9 °C	0.7 °C
Water Mug	25.12 °C	23.2 °C	1.92

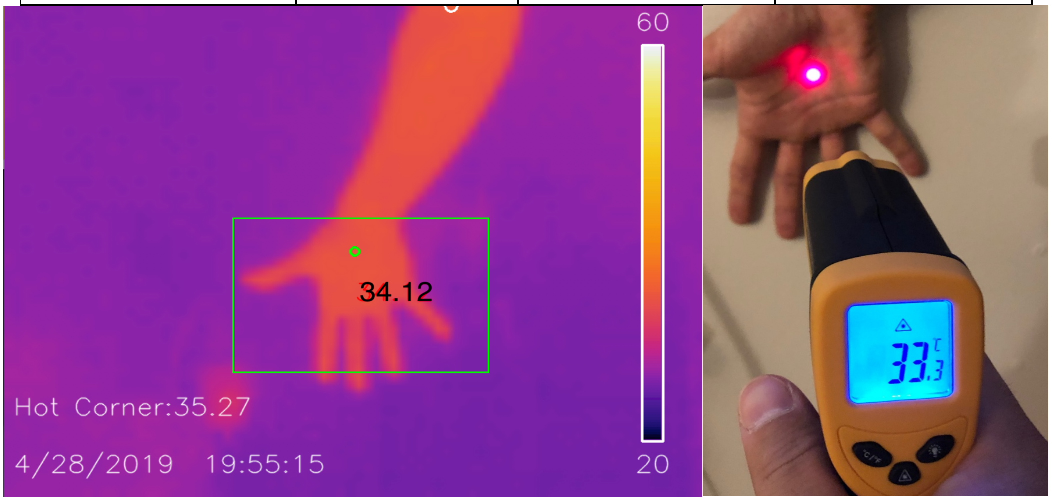


Figure 54 Accuracy Measurement with Human Body Temperature

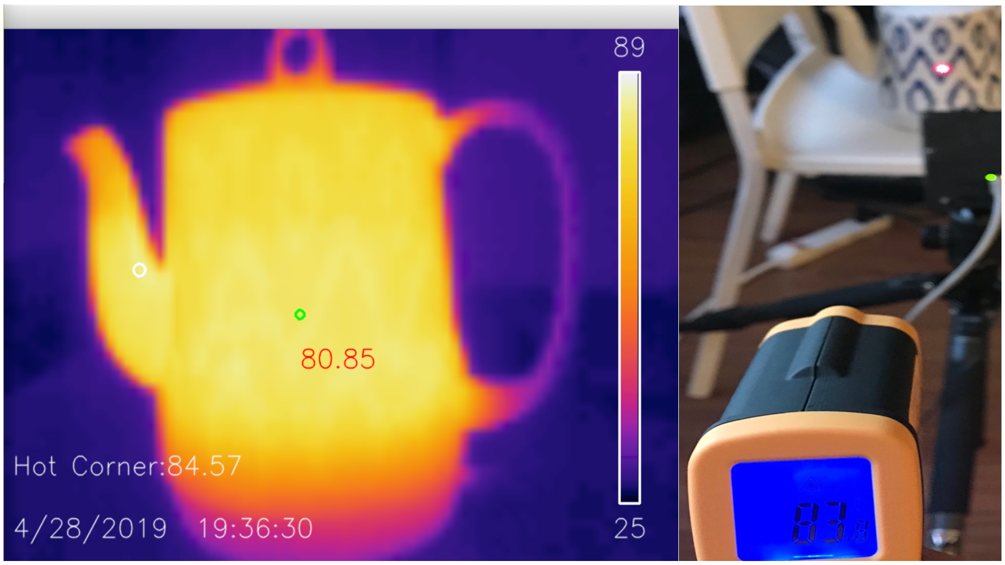


Figure 55 Accuracy Measurement with Water Boiler

6.3.3 Calibration Test Results

The total temperature calibration has to account for both the window effect and the temperature offset comparing to other product. The window effect introduced an offset of -0.58 °C to temperature measurement.

Recall in Table 15 that the offsets between OSIMS and COTS are: 0.82, -2.15, 0.7, 1.92, with mean value: 0.32. Those values proved that OSIMS gave very close measurements to the COTS products. Therefore, it is safe to assume that OSIMS has zero measurement offset, as well as an accuracy of +/- 2 [°C].

In conclusion, the temperature of OSIMS is calibrated by adding 0.58 °C from the window effect.

6.4 Thermally Reflective Surface Measurements

An important factor to consider while conducting infrared measurements is thermal reflection. As mentioned previously, emissivity describes the efficiency of a surface at radiating energy in a defined waveband at a given temperature. Most shiny metals have very poor infrared radiation, whereas non-metal surfaces are more efficient emitters. Another characteristic of metal surface is thermal reflection, which means the infrared energy radiated from near sources is reflected on metal surface, resulting in thermal distortion from the surroundings. These effects were demonstrated below. A hot coal was placed adjacent to an aluminum foil surface, which was reflected by the surface, which introduced reflection distortion to the measurements.

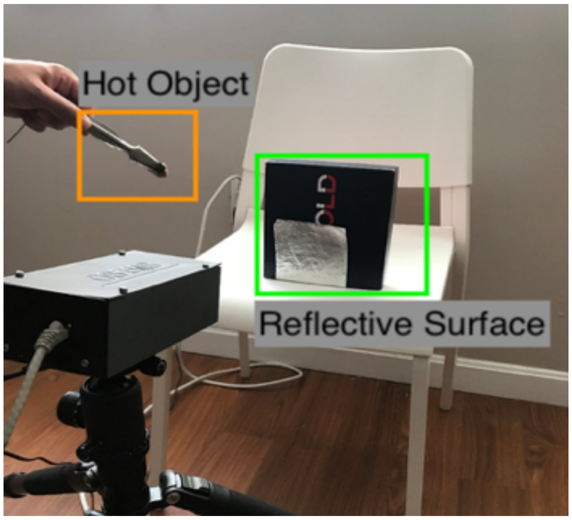


Figure 56 Metal Surface Reflectivity

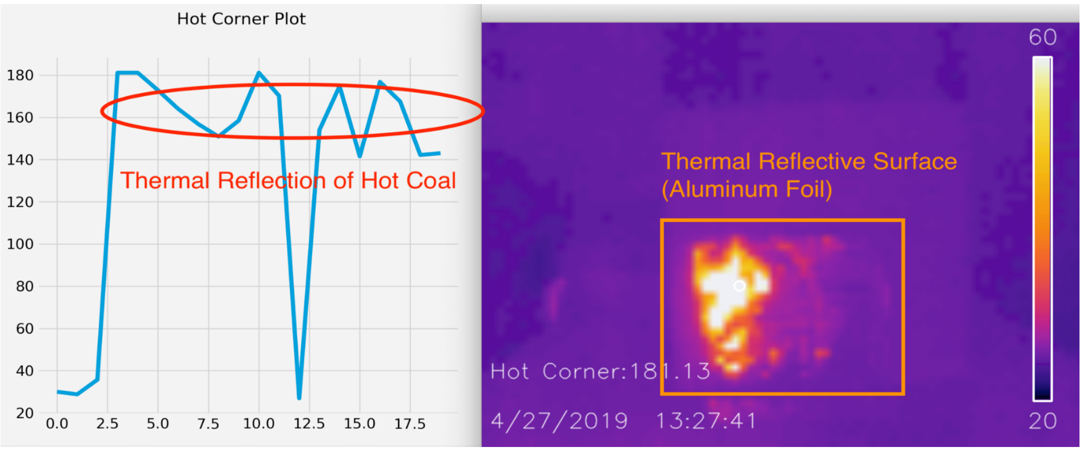


Figure 57 Thermal Reflection of Metal Surface

Emissivity affects the temperature readings significantly. Surfaces such as iron, steel or aluminum have an emissivity of 0.74, 0.69, and 0.04 respectively. Thus, these factors should be taken into consideration while performing calibration. However, another approach for acquiring good measurements on these surfaces is to apply white paint (emissivity 0.94) or electric tape (emissivity 0.95). This approach was put to test with a glass object wrapped in aluminum foil, containing warm water, where one side of the glass was coated with white marker. The measurements are shown below.

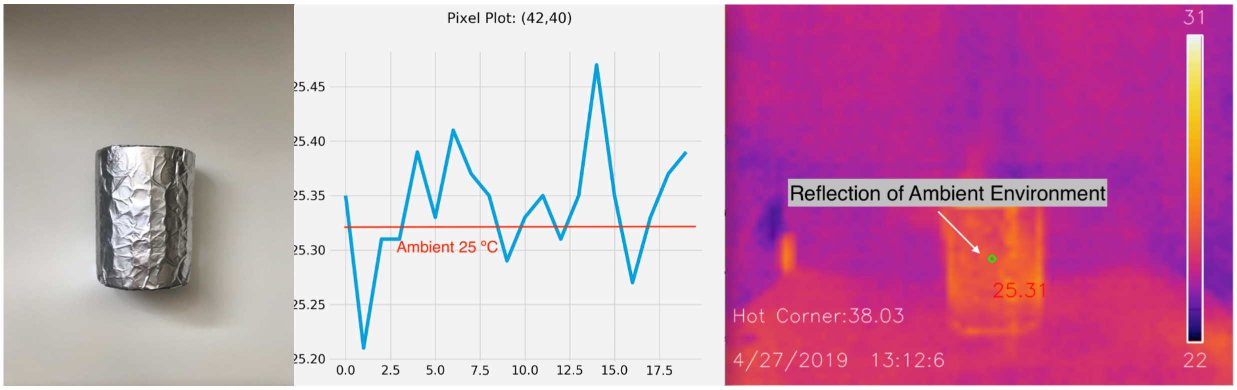


Figure 58 Temperature Measurement of Uncoated Surface with Reflection

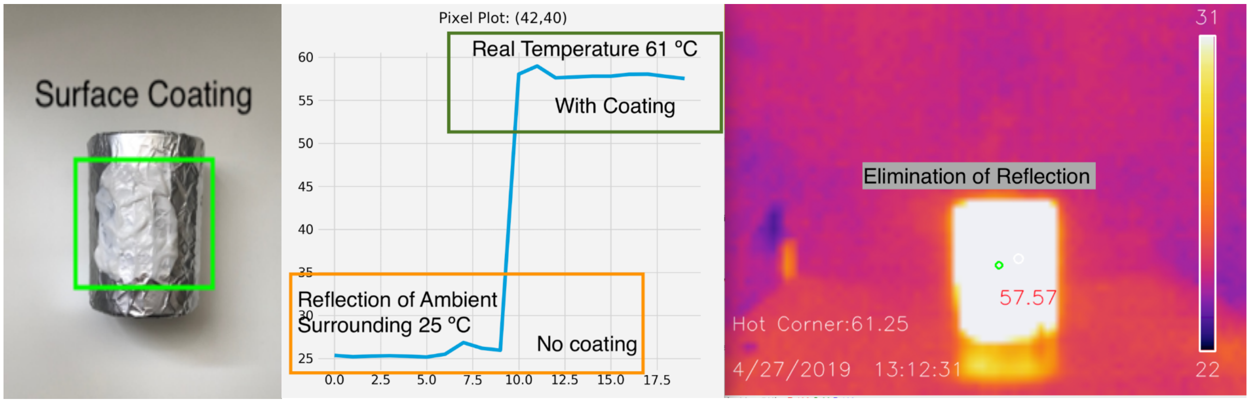


Figure 59 Temperature Measurement of Coated Surface

As expected, the measurement indicated that the temperature reading on the aluminum foil surface was simply radiation caused by the ambient environment (25.31°C). This was expected because of the poor emissivity of 0.04 of the aluminum foil. However, after applying white coating on the surface, the thermal camera was able to depict the actual body temperature. The shift in temperature between uncoated and coated was 32.26 °C. In addition, a DC motor was put to test with and without white coating. The results are shown below.

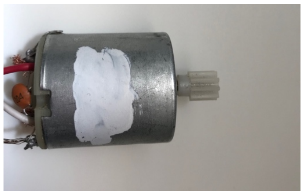


Figure 60 Electric Motor Partly Coated

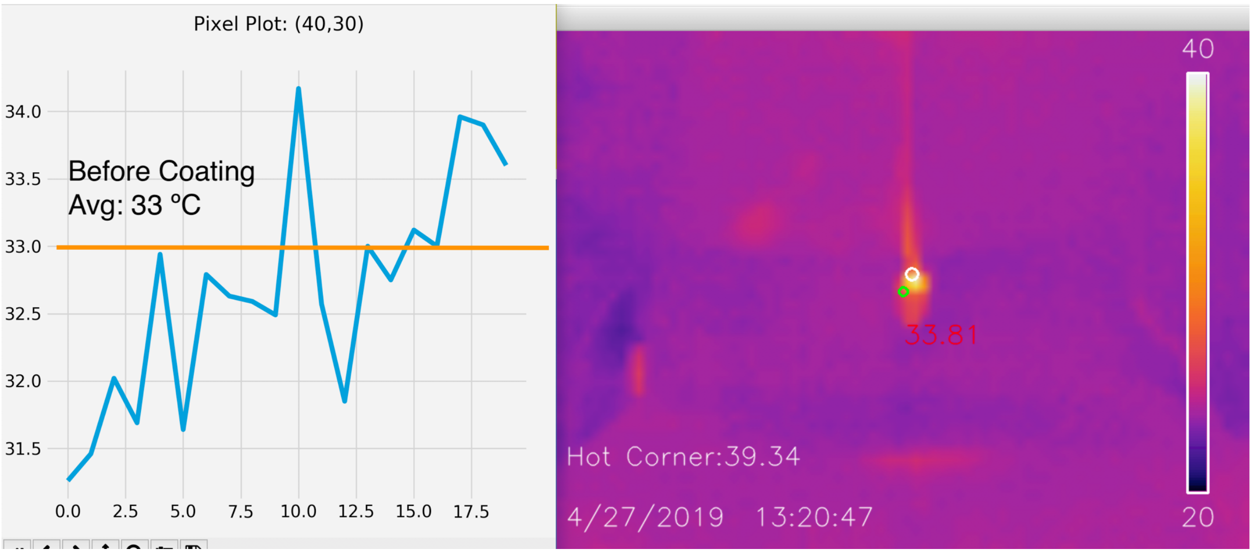


Figure 61 Electric Motor with Partial Reflective Surface

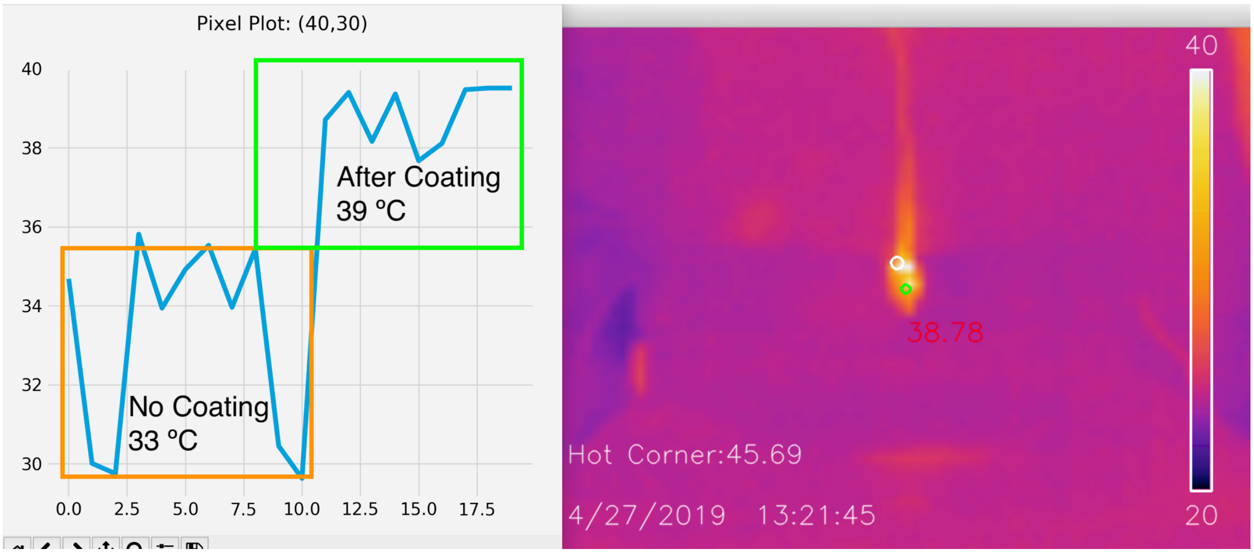


Figure 62 Electric Motor with Coated Surface

Before applying the white coating, the DC motor was reading an average temperature of 33 °C. Once the white coating was applied, the temperature reading increased by 6 °C. The whiteout coating effectively increased the emissivity of the motor surface.

The purpose of these tests is to demonstrate the difficulties in measuring thermally reflective surfaces and to introduce possible solutions to overcome it. In a real system application, each OSIMS unit will be calibrated for the emissivity of the targeted surface material. An appropriate coating will be applied to any reflective material.

6.5 Software Functionality Verification

6.5.1 Multiple Corner Monitoring

To verify the multiple corner monitoring functionality of OSIMS software, OSIMS was configured to plot temperatures of three test corners shown in Figure 63 simultaneously. To recapitulate, test unit one on the top left is a simulation of a transformer; top right is a simulation of an electric motor; bottom right simulates pipeline with internal heat. The test procedures were:

- Turned on and heat up test units individually.
- Enabled streaming in software and added three pixels each corresponds to one test unit.
- Set warning threshold temperature at 80 °C.
- Observed and verified Hot Corner Plot.
- Observed and verified Pixel Plot.
- Observed and verified Warning.

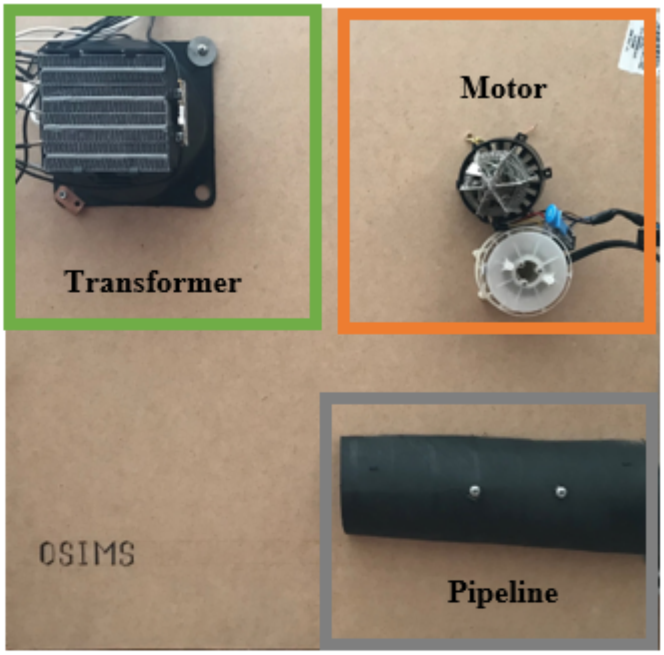
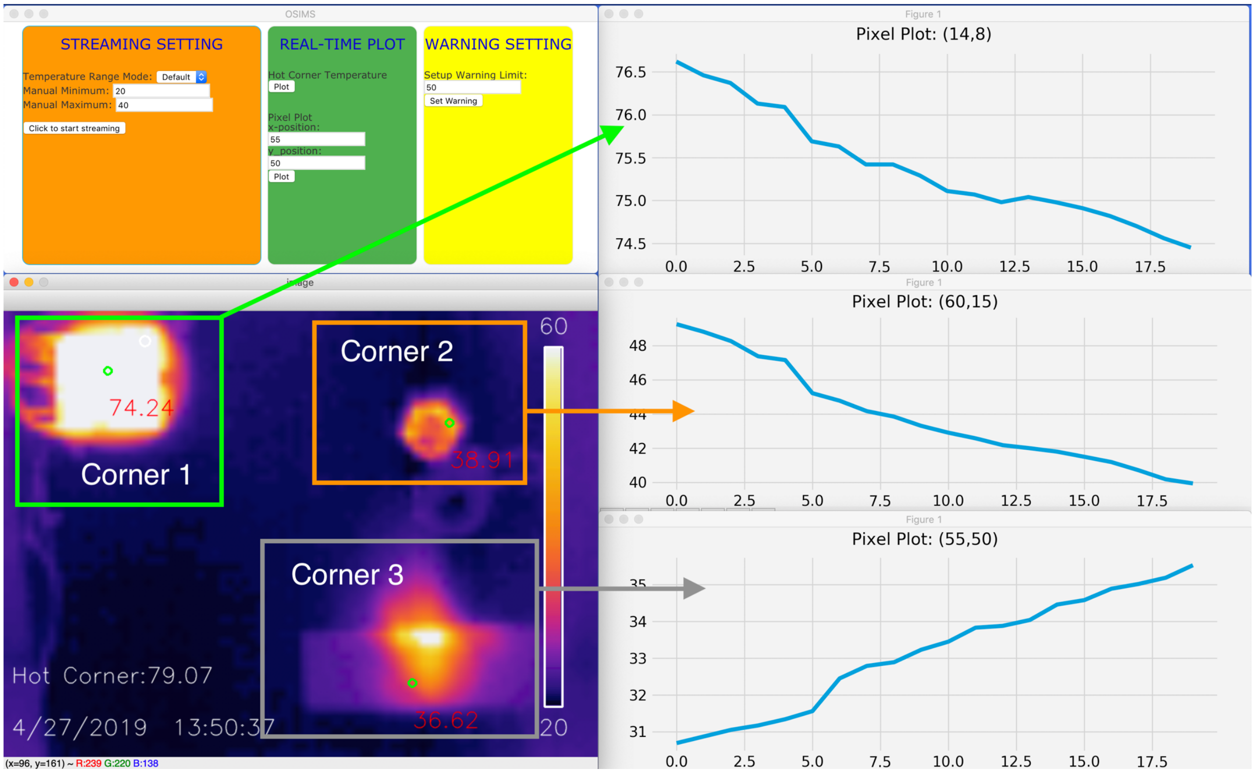


Figure 63 Multiple Corner Test Bench

The top left is the user interface of OSIMS. For the purpose of this test, the streaming was configured by default showing a temperature range of 20 °C ~ 60 °C. Then three pixel plots were manually added to the test units. The test was conducted for over one hour, which sufficiently proved the run-time stability of the alpha phase software. During the test, three test corners were heated up above the warning threshold individually and all succeeded in triggering the alarm. However, lags were observed in video streaming which caused worst-case one second delay in plotting, and data loss.



Temperature Range Mode: Default
Manual Minimum: 20
Manual Maximum: 40

Pixel Plot
x-position: 55
y_position: 50

Setup Warning Limit: 50

Figure 64 Multiple Corner Verification

6.5.2 Data Log

Recall that OSIMS software is designed to store a data log in two separate formats: video avi format, and cvs data format. The avi video gives the user the capability of storing visual feedbacks of historical data, whereas the cvs data can be used for future software development such as machine learning, and data processing. The video is stored locally at a defined directory, Figure 65 shows the data archived of .avi and .cvs files uncorrupted. Figure 66 shows the csv data structure. For each row, the first cell is the time, followed by temperatures of 4800 pixels.

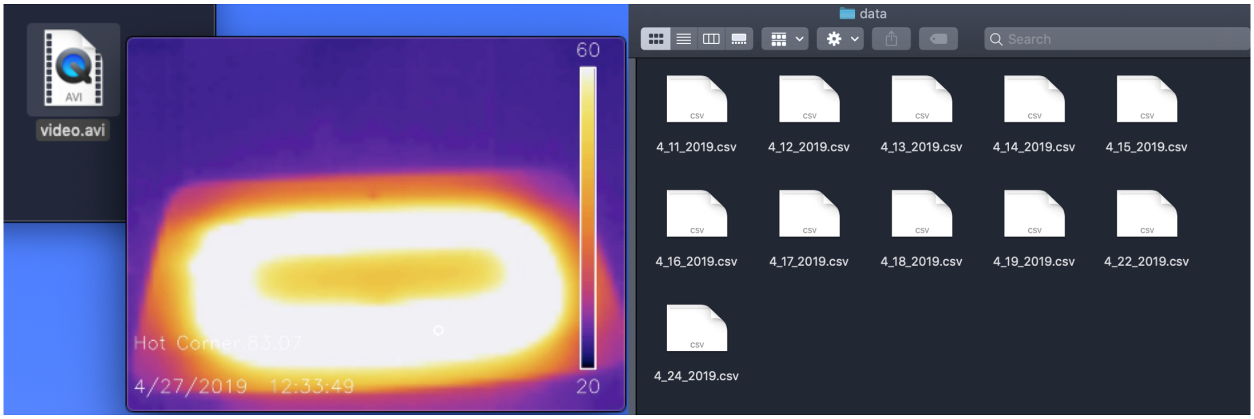
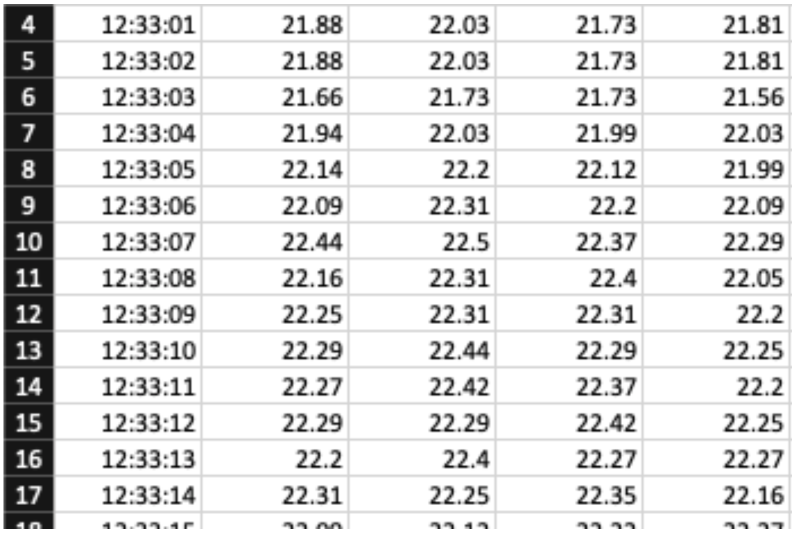


Figure 65 Data Archive



4	12:33:01	21.88	22.03	21.73	21.81
5	12:33:02	21.88	22.03	21.73	21.81
6	12:33:03	21.66	21.73	21.73	21.56
7	12:33:04	21.94	22.03	21.99	22.03
8	12:33:05	22.14	22.2	22.12	21.99
9	12:33:06	22.09	22.31	22.2	22.09
10	12:33:07	22.44	22.5	22.37	22.29
11	12:33:08	22.16	22.31	22.4	22.05
12	12:33:09	22.25	22.31	22.31	22.2
13	12:33:10	22.29	22.44	22.29	22.25
14	12:33:11	22.27	22.42	22.37	22.2
15	12:33:12	22.29	22.29	22.42	22.25
16	12:33:13	22.2	22.4	22.27	22.27
17	12:33:14	22.31	22.25	22.35	22.16

Figure 66 CSV Format Data Log

6.5.3 Summary

Overall, all functionalities of OSIMS alpha stage software were verified. The thermal video streaming was verified to be stable and spontaneous. The real-time data plot features met with design spec; the warning feature gave a timely response when temperature reached the critical threshold. The multiple corner monitoring capability was proved in this test, as well as software run-time stability without crashing for 1+ hour. The shortcomings such as the lag and delay observed in streaming are yet to be investigated further, and improved in the beta stage.

6.6 Benchmark

This section summarizes the milestones achieved within the scope of this MQP, and the future work improvements on the current design.

Table 18 OSIMS Design Milestones and Future Work

Specifications	**Expected**	**Achieved**	**Future Work**
Frame Rate	8.7 [fps]	6 [fps]	8.7 [fps]
Video Data Interface	SPI	✓	SPI
Control Interface	I^2C	Not utilized	I^2C
IR Video Resolution	80 x 60	80 x 60	160 x 120
Ethernet Protocol	TCP/IP	✓	
Power Source	Power over Ethernet	12V Adapter	PoE
Supply Voltage	7 – 12 [V]	12 [V]	
Dual Camera System	N/A	N/A	✓
Storage Media	Video	✓	
Encapsulation	3D printed case	Onyx Material	IP67
PCB	Extension Board	✓	
System Design	Eval + Modules	✓	Chip level
IR Window	Germanium (Ge)	✓	ZnSe
Accuracy	+/- 2 [°C]	+/- 2 [°C]	
Precision	+/- 0.1 [°C]	+/- 0.11 [°C]	
Standard Deviation	N/A	0.11 [°C]	
Maximum Deviation	N/A	0.175 [°C]	
Measurement Resolution	0.1 [°C]	0.11 [°C/step]	
Avg Step Size	N/A	0.06 [°C/step]	
Resolution	N/A	0.11 [°C/step]	

7 DISCUSSION

7.1 Difficulties

There were several challenges faced throughout the development stages. When the SPI interface was first configured, the clock line introduced problems for the synchronization due to its high impedance state of the pin. Despite configuring the pin at HIGH state through STM32CubeMX, it was necessary to hardcode the pin to a HIGH state on the firmware during the initialization stage.

LwIP introduced critical challenges on the implementation of the Raw API. Considering the amount of data that is being transferred over a second, the TCP/IP protocol occasionally introduces a delay of 1 to 2 seconds during streaming. This could be due to the memory allocation or buffering while preparing for the transmission. Also, having the software continuously polling does not provide the fastest medium of communication. Another approach could be RTOS by using Netconn or Socket API.

To recapitulate the OSIMS software frontend and backend connection mentioned in 5.5.2, it is difficult for Python to send data to JS. Therefore, an alternative solution of creating a system data buffer was introduced, which was the easier but less correct option. However, if Python writes critical data to the buffer, it can be problematic if a Python process crashes and disrupt the system data buffer, or if system data is deleted by a human operator by accident. In future work, the beta phase of OSIMS can implement the correct method of data transfer from Python to Electron.

In Chapter 5.5.2 another data transfer challenge was mentioned, that Python subprocesses could not transfer data between each other. This challenge was circumvented by creating a data log buffer that is accessible to both Python subprocesses. However, the downside is that there will be a time delay for the data received by different processes. One possible alternative solution that can be implemented in the future is integrating subprocesses together to form one subprocess.

7.2 Future Work

The first prototype development stage of OSIMS achieved the following:

- Three boards are used for the prototype design: microcontroller development board, thermal camera breakout board, and the expansion PCB. Both Ethernet and power cabling were necessary for supporting the hardware.
- Real-time thermal imaging and streaming over the Ethernet using TCP/IP LwIP

The LwIP Raw API standalone version introduced implementation challenges. The amount of data being sent was large, and occasionally the system would encounter a 1 second delay. However, this protocol served as a baseline for the Ethernet communication, with an overall satisfactory streaming performance for the application. Because the OSIMS unit would be stationary, the small delay may not be relevant.

- OSIMS cross-platform PC software for data configuration and visualization.

The next hardware development stage of OSIMS are the following:

- Designing a chip level PCB that will accommodate all the components. It will integrate a dual camera system, both visible and infrared. A 640x480 resolution camera would be sufficient for normal video streaming, and 80x60 can be considered acceptable for the application. It will introduce a new feature to allow the operator to switch between normal and infrared thermal imaging while monitoring. Both camera cores will be integrated within the main board with their corresponding necessary hardware interface. Additionally, a high-power LED will be installed to help illuminate the area around the camera and be controlled by the OSIMS software.
- A temperature and humidity sensor will be integrated in the main board to constantly monitor the internal condition of the OSIMS unit.
- Integrating Power over Ethernet (POE) capability. POE is a technology that allows a network cable to carry electric power. This solution is cost effective by reducing the time and expense of having electric power cabling installed. POE delivery is intelligent, and designed to protect network equipment from overload, underpowering, or incorrect installation. Also, having power available on the network means that installation and scalability of network connections is simple and effective.

The next software development stages of OSIMS are the following:

- Develop the firmware for supporting both the visible camera and infrared camera streaming operation simultaneously over Ethernet.
- Experiment with Internet protocols such as Ethernet/IP, Modbus TCP, Real-Time Streaming Protocol (RTSP), Real-Time Transport Protocol (RTP), etc. The purpose is to determine the appropriate medium for supporting real-time streaming of both cameras over Ethernet without introducing delays.
- OSIMS software will introduce an interactive multiple point detection feature, where the operator selects the area of interest in real-time, and the corresponding temperature will be displayed on the screen. It will provide three modes of monitoring: visible, thermal, and both. The last mode combines video from both cameras, adding more details of the surrounding.

Finally, the OSIMS case will be modelled to satisfy an IP67 rating, protection from immersion in water with a depth of up to 1 meter (or 3.3 feet) for up to 30 mins, and protection from contact with harmful dust. Additionally, the main board PCB will have a conformal coating to protect it from environmental conditions that might introduce moisture or chemical contaminants. Regarding the infrared window material, CTE is the rate at which a material expands or contracts given a change in temperature. As shown previously, germanium has a very high index gradient, possibly degrading optical performance if used in a thermally volatile setting. Despite that in our application, the unit will not be exposed to temperature changes as it would monitor at a certain distance, it is necessary to take into account such effect. Therefore, a Zinc Selenide window would be the next infrared window used for OSIMS.

8 CONCLUSION

The goal of this project was to research and develop a safety monitoring system for industrial power systems, featuring infrared imaging and multiple point sensing network user interface. Within the scope of this MQP, we successfully built the first prototype of Online System Integrity Monitoring System (OSIMS). In addition, the alpha phase developments of OSIMS firmware and software were completed. The prototype was finished with PCB expansion board plus 3D printed case. It was tested to have stable performance over 24 hour continuous thermal video streaming over Ethernet. OSIMS alpha software was designed to support multiple operating systems including Windows, MacOS and Linux. In terms of functionality, it achieved continuous infrared video streaming, real-time data plot including hot corner and pixel plot, setting alarm threshold and historical data log.

The shortcomings of the alpha stage firmware and software is that some functions and algorithms are not optimized for faster and more stable operation. On the firmware side, the Ethernet protocol implementation introduced occasional delays during the transmission due to the large size of data being sent. OSIMS alpha stage software used a temporary solution to circumvent the difficulty in transferring data between the frontend and backend. In the future, we will keep improving based on the alpha stage prototype, addressing issues encountered to improve run-time stability.

REFERENCES

[1] S. Editor, "Safehoo," 06 10 2017. [Online]. Available: http://www.safehoo.com/Case/Case/Electric/201710/498049.shtml.

[2] W. Jiang, "Xinhuanet," 26 07 2015. [Online]. Available: http://www.xinhuanet.com//local/2015-07/26/c_1116043700.htm.

[3] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Texas_City_Refinery_explosion#Explosion.

[4] "efunda," [Online]. Available: https://www.efunda.com/formulae/heat_transfer/home/overview.cfm. [Accessed 2018].

[5] "electrical4u," 9 August 2018. [Online]. Available: https://www.electrical4u.com/joules-law/.

[6] "Hukseflux," 2018. [Online]. Available: https://www.hukseflux.com/applications/industrial-monitoring-and-control-heat-flux-and-heat-transfer-measurement.

[7] "TTS," 2 February 2012. [Online]. Available: https://www.transmittershop.com/blog/know-about-industrial-process-temperature-measurement.

[8] J. Raymond, "AJ Design," 2015. [Online]. Available: https://www.ajdesigner.com/phpwien/wien_equation.php.

[9] "Wikipedia," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Electromagnetic_spectrum#/media/File:Electromagnetic-Spectrum.svg.

[10] J. Merchant, "OMEGA," [Online]. Available: https://www.omega.com/technical-learning/infrared-temperature-measurement-theory-application.html. [Accessed 2018].

[11] "FLIR," [Online]. Available: https://www.flir.com/discover/how-does-an-ir-camera-work/. [Accessed 2018].

[12] V. P. G. J. V. L. M. J. B. F. Usamentiaga R, "Infrared thermography for temperature measurement and non-destructive testing," *Sensors (Basel),* Vols. 14(7):12305-48, 2014.

[13] "OpenCV," [Online]. Available: https://opencv.org/about/.

[14] Y. Zhang, "GitHub," 24 April 2019. [Online]. Available: https://github.com/githubyzhang/OSIMS.

[15] Noria Corporation, 2018. [Online]. Available: https://www.machinerylubrication.com/Read/29331/machine-failure-causes.

[16] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Graphical_user_interface. [Accessed 5 December 2018].

[17] T. Curtis, "RELIABLEPLANT," [Online]. Available: https://www.reliableplant.com/Read/22290/IR-camera-condition-monitoring. [Accessed 2018].

[18] "DoITPoMS," [Online]. Available: https://www.doitpoms.ac.uk/tlplib/thermal_electrical/printall.php. [Accessed 2018].