REMOTE MONITORING SYSTEM FOR INFANT INCUBATOR BASED ON IMAGE FUSION TECHNIQUE

THANAPORN CHUENPIROM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN BIOMEDICAL ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2023

KMITL-2023-EN-M-317-121

REMOTE MONITORING SYSTEM FOR INFANT INCUBATOR BASED ON IMAGE FUSION TECHNIQUE

THANAPORN CHUENPIROM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN BIOMEDICAL ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2023

KMITL-2023-EN-M-317-121

Thesis	Remote Monitoring System for Infant Incubator based on Image Fusion Technique
Student	MissThanaporn Chuenpirom
Student ID.	65016041
Degree	Master of Engineering
Program	Biomedical Engineering
Academic Years	2023
Project Advisor	Assoc.Prof.Dr. Chuchart Pintavirooj

ABSTRACT

Premature infants face several challenges due to their underdeveloped organs and low weight, including difficulty breathing, maintaining body temperature, and receiving adequate light. Therefore, incubators that can address these challenges with appropriate materials and technology are necessary. An ideal incubator should be equipped with sensors that can measure the infant's temperature, humidity, weight, brightness, and movements. The temperature and humidity must be maintained within a proper range to ensure a stable environment and maintain the infant's core temperature at 37°C.

Modern health monitoring systems rely heavily on sensor technology to measure and transmit physiological information, such as temperature, to remote locations. One such system proposed is to remotely monitor newborns and transmit the data via a GSM network, providing real-time measurements of various parameters, including temperature and pulse rate. However, the use of sensors for remote monitoring is limited and can be expensive.

This thesis proposes the development of an affordable smart infant incubator with remote monitoring based on IoT. The monitoring system is controlled through a Raspberry Pi connected to a visual camera and thermal IR camera. The thermal IR camera is equipped with the temperature sensor MLX90640, which is used to measure the infant's body temperature. The monitoring system employs image processing techniques to improve monitoring and can be remotely monitored using IoT. The

results show that the smart infant incubator is accurate, reliable, and cost-effective, and its control system is similar to traditional incubators.

However, there are some limitations to the incubator's performance, including inefficient airflow due to components such as the blower and heater and the performance of the monitoring system not being smooth due to processing multiple functions in the same code. Future research can focus on addressing these issues to further improve the incubator's performance.

ACKNOWLEDGEMENTS

I am deeply grateful to Assoc.Prof. Dr. Chuchart Pintavirooj, my esteemed project advisor, for his unwavering support and guidance throughout this endeavor. His kindness, expertise, and mentorship have been instrumental in the successful completion of this project. I am truly grateful for the valuable advice and meticulous corrections that I have received from him, which have enabled me to achieve the goals I had set for this thesis.

I would also like to extend my sincere appreciation to the Biomedical Engineering Department of the Faculty of Engineering for their generous provision of research equipment and facilities that have been crucial in the completion of this thesis. Their support has been invaluable in facilitating the smooth progress of my research.

Lastly, I would like to express my heartfelt thanks to everyone who has supported and assisted me in various ways, including those who may not be mentioned in this thesis. Your encouragement, assistance, and contributions have been immensely valuable, and I am deeply grateful for your unwavering support.

Thanaporn Chuenpirom

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
API	Application Programming Interface
IoT	Internet of Things
NICU	The Neonatal Intensive Care Unit
OpenCV	Open-Source Computer Vision Library
PID	Proportional Integral Derivative
SSR	Solid State Relay
STC	Steady Temperature Condition

CHAPTER 1
INTRODUCTION

1.1 Background of the study

Approximately ten years ago, the annual global birth rate exceeded 130 million, but unfortunately, approximately 8 million of these newborns did not survive beyond their first year due to complications associated with prematurity [1]. In response to the critical healthcare needs of these vulnerable infants, specialized medical facilities known as Neonatal Intensive Care Units (NICUs) were established to provide specialized medical care to newborns requiring intensive medical attention. NICUs predominantly admit premature infants, who are defined as infants born before 37 weeks of gestation, with low birth weight (less than 2.495 kg), or with medical conditions that necessitate specialized care. The safety of premature infants is a critical concern in neonatal care, and proper monitoring of neonatal incubators is essential to prevent accidents and ensure the well-being of these vulnerable babies [2]. Infant incubators are enclosed equipment designed to provide a clean and controlled environment for premature babies, where they receive observation and care. One of the key functions of an incubator is to monitor various biological parameters to ensure the safety of the babies and prevent adverse outcomes, including mortality rates [2]. Thus, incubators play a vital role in safeguarding the health and well-being of premature babies by providing a controlled environment and continuous monitoring of their physiological status.

Most of the medical devices produced in Thailand are not high in innovation and technological complexity. It is the production of basic equipment which mainly uses domestic raw materials such as rubber and plastic. Regarding medical equipment that requires innovation and advanced technology, Thailand still has to rely on imports from abroad, which may result in increased costs, logistical challenges, and potential delays in the availability of incubators for neonatal care units [3]. However, in Thailand, there is a notable absence of local production of infant incubators, and research on incubators is limited. Therefore, interested in studying this matter in order to foster the growth and development of the Thai medical device industry.

The Internet of Things (IoT) is a networked system based on Internet and information technology, utilizing embedded systems such as processors, sensors, and communication hardware to collect, transmit, and respond to data obtained from their environments. The continuous advancement of IoT has enabled remote monitoring of various parameters, providing medical professionals with the ability to monitor such parameters remotely [4]. The application of IoT technology has been utilized in our incubator to develop a monitoring system with the aim of enhancing the performance of the device. This thesis proposes a smart infant incubator that incorporates IoT technology for monitoring newborn babies. The system uses Raspberry Pi, a visual camera, and a thermal IR camera to control the monitoring system. The MLX90640, a thermal IR camera, is used as a temperature sensor to measure the infant's body temperature. The software of this smart infant incubator is designed to improve the monitoring system by applying image processing techniques to the camera images and combining them with a web server for remote monitoring. The system also utilizes a database server to store the sensor data.

1.2 Objectives

1.2.1 To manufacture an infant incubator with affordable price that has the same functionality as commercially available models.

1.2.2 Applying image processing techniques to enhance the monitoring system of an infant incubator.

1.2.3 To develop a smart infant incubator by using IoT technology.

1.3 Scope of study

1.3.1 The hardware module of the monitoring system is Raspberry Pi along with visual and thermal IR cameras as a temperature sensor.

1.3.2 Design the software of monitoring system by applying image processing technique to enhance the images.

1.3.3 Improve monitoring system by applying IoT technology to remote monitor.

1.3.4 Design the control system to be able to function as the commercial models.

CHAPTER 2
REVIEW OF RELATED STUDIES

2.1 Premature Babies

Premature birth refers to the situation where a baby is born before the completion of 37 weeks of pregnancy, whereas a typical pregnancy lasts approximately 40 weeks. Premature babies often experience serious health issues, especially when they are born very early, and the severity of these problems may vary. The earlier a baby is born, the higher the risk of health challenges [5].

There are different categories of premature births based on the gestational age of the baby:

1. Late preterm: Refers to babies born between 34 and 36 completed weeks of pregnancy. Most premature births occur during this period.
2. Moderately preterm: Refers to babies born between 32 and 34 weeks of pregnancy.
3. Very preterm: Refers to babies born between 28 and 32 weeks of pregnancy.
4. Extremely preterm: Refers to babies born before 28 weeks of pregnancy.

2.1.1 Symptoms

In the case of premature birth, the symptoms experienced by a baby can range from very mild to more serious health problems. Some common signs of a baby being born too early may include [5]:

- Small size, with a head that appears larger compared to the body.
- Sharper and less rounded features compared to a full-term baby, which may be due to a lack of fat-storing cells.
- Fine hair covering much of the body.
- Low body temperature, particularly right after birth in the delivery room.
- Difficulty breathing.
- Feeding problems.

2.1.2 Complications

Not all premature babies experience health complications but being born too early can result in both short-term and long-term medical issues. Generally, the earlier a baby is born, the higher the risk of complications, with birth weight also playing a significant role [5].

Short-term complications of premature birth can include breathing problems due to underdeveloped lungs, which may result in respiratory distress syndrome or bronchopulmonary dysplasia. Heart problems such as patent ductus arteriosus (PDA) and low blood pressure are also common among premature babies and may require medical intervention. Other short-term complications may include brain problems such as intraventricular hemorrhage, temperature control problems leading to hypothermia, digestive problems like necrotizing enterocolitis (NEC), blood problems such as anemia and newborn jaundice, metabolism problems, and immune system problems leading to a higher risk of infections such as sepsis.

Long-term complications of premature birth can manifest in conditions such as cerebral palsy, trouble with learning and developmental milestones, vision problems such as retinopathy of prematurity, hearing problems, dental problems, behavior and mental health problems, and ongoing health issues that may persist into adulthood.

In summary, premature birth can result in a range of short-term and long-term complications, and the risk of complications is higher with earlier gestational age and lower birth weight. It is important for premature babies to receive appropriate medical care and monitoring to address these potential health concerns.

2.1.3 Treatments

The health of your premature baby is meticulously monitored in the neonatal intensive care unit (NICU) or special care nursery [5].

Supportive

Supportive care for neonates may encompass various interventions, including:

- Incubator placement: Neonates may be placed in an enclosed plastic bassinet, known as an incubator, to maintain a stable body temperature. Kangaroo care, a method of direct skin-to-skin contact between the parent and baby, may be demonstrated by NICU staff at a later stage.
- Vital sign monitoring: Sensors may be affixed to the neonate's body to track essential parameters such as blood pressure, heart rate, breathing, and temperature. Mechanical ventilators or continuous positive airway pressure (CPAP) devices may also be employed to assist with breathing.
- Feeding tube administration: Initially, neonates may receive fluids and nutrients through a vein via a feeding tube. Subsequently, breast milk or other feeds may be administered via a tube passed through the baby's nose and into the stomach. Breastfeeding or bottle-feeding may be initiated once the neonate is deemed strong enough to suck.
- Fluid management: Neonates require specific amounts of fluids daily, contingent upon their age and health status. The NICU team will closely monitor fluid, sodium, and potassium levels to maintain appropriate hydration. Intravenous administration may be utilized if fluids are warranted.
- Phototherapy for jaundice: Neonates with jaundice may be placed under specialized lights known as bilirubin lights to facilitate the breakdown of excess bilirubin, a yellow-colored substance that accumulates in the body when the liver is unable to process it adequately. Protective eye masks may be used during this treatment to promote rest.
- Blood transfusions: Some preterm neonates may require blood transfusions due to specific health conditions or frequent blood sampling for tests.

Medications

Medicines may be administered to a baby in the neonatal intensive care unit (NICU) or special care nursery for various reasons. These may include:

- Surfactant, which is used to treat respiratory distress syndrome and improve lung function.
- Bronchodilators or medications given intravenously to support breathing and heart rate.
- Antibiotics to treat bacterial infections or as a preventive measure against potential infections.
- Diuretics to manage excess fluid by increasing urine output.
- Intravitreal medication to inhibit the growth of new blood vessels in the eye, preventing retinopathy.
- Medications to close a heart defect known as patent ductus arteriosus.

Surgery

In certain cases, surgical intervention may be necessary to address health issues in premature babies. It is important to communicate with your baby's healthcare team to gain a comprehensive understanding of the potential complications that may require surgery. Familiarize yourself with the various types of surgeries that may be indicated for the treatment of these issues as well.

Taking baby home

When a baby exhibits the following signs, it indicates that they are ready to be discharged and go home:

- Able to breathe independently without requiring support.
- Maintains a stable body temperature.
- Capable of breastfeeding or bottle-feeding.
- Demonstrates consistent weight gain over time.
- Does not have any significant health problems.

In some cases, a premature baby may be discharged before meeting all of these requirements, but only after the baby's medical team and family have established and agreed upon a plan for home care and follow-up health care. This ensures that appropriate measures are in place to support the baby's ongoing health and well-being after leaving the hospital.

2.2 Infant incubator

Infant incubators provide a healthy and controlled environment for premature babies which are born before their important organs are able to develop as well as some sickness full-term babies. Incubators are used in hospitals, the primary care department or in the neonatal intensive care unit, NICU as shown in figure 2.1. The baby will be placed in a transparent cabinet which maintains a neutral environment, the appropriate temperature and moisture. The features of an infant include temperature and humidity control, increased oxygen delivery, protection against infections and illnesses, and noise reduction. The incubator must be capable of maintaining the baby's body temperature (36-37.2°C) by maintaining an ambient air temperature of 34-38°C with a humidity level of 40-80% [6].

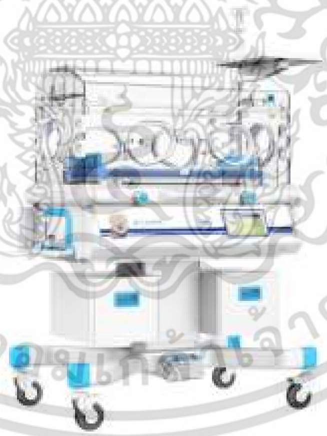

Figure 2.1 Infant incubator [7].

Designing of infant incubator

It is the same concept that regulates the operation of every incubator for newborns as shown in figure 2.2. The inside of the incubator has a fan that blows filtering ambient air above a heat source and a water tank. Air can be enriched with additional oxygen through the use of an air control valve. Moisture, heat, and enriched air are now being delivered to the top cabinet, where the newborn is being kept. Ventilation occurs through the cabinet and re-circulation occurs via the air processing system.

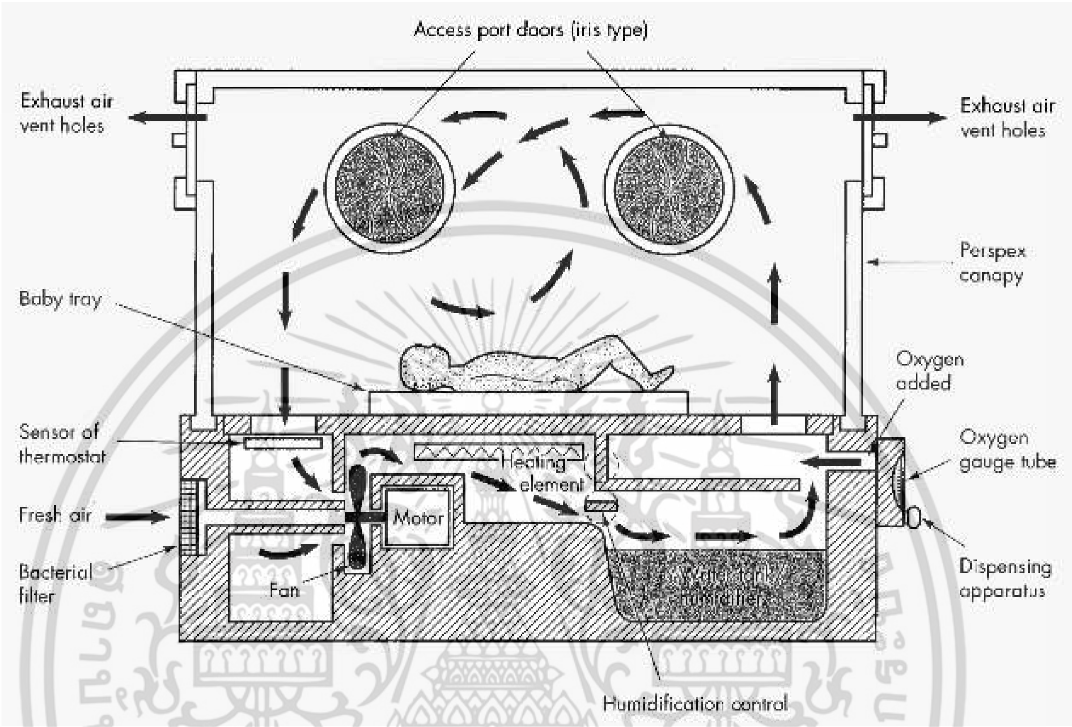

Figure 2.2 Design of Infant incubator [6].

The transparent cabinet is constructed entirely of Plexiglas (Perspex). A hinged hood can be opened to admit or discharge the infant. The nurse may readily access the infant through front portholes without exposing the infant to the environment by opening the hood.

After the nurse sets the desired temperature, the incubator maintains it automatically. Typically, humidity and oxygen concentrations are manually adjusted. Additionally, the incubator features a safety shut-off function that activates when the temperature exceeds 40°C. Additionally, an alert is triggered if the fan does not turn on in the event of a power failure.

Components utilized vary. The older incubators are controlled electrically, the newer ones are controlled electronically, and the latest generation is controlled by a microprocessor. However, the outcome is the same and the older ones are just as efficient and accurate as the newer ones [6].

- Fan

Fan circulates filtered room air through the heat element and humidifier, allowing them to work more efficiently. Without a fan, heat could not be directed away from the heat element and the heat element would overheat, resulting in the incubator being too hot to handle.

- Filter

In simple incubators, washable foam filters are applied. They can be reused after being washed and dried. However, modern incubators typically have disposable bacterial filters. They are not washable and must be replaced.

- Heater

To heat air, a heating element constructed of coiled resistance wire, as seen in hair dryers, or a tube type (flat or coiled) as seen in autoclaves is utilized. However, unlike in autoclaves, the heater is much less efficient and thus does not reach such high temperatures. The power generated is between 100 and 300 watts.
The heater is regulated electronically using relay, TRIAC, or just a thermostat.

- Temperature control

In simple incubators, the temperature is controlled by thermostat (figure 2.3), which is composed of a sensor and a pressure gauge. The sensor consists of a thin capillary tube that enters into the pressure canister and measures the pressure (or expansion chamber). This chamber contains a metal cover, sometimes known as a diaphragm, that can be moved. This closed system contains a liquid or gas that expands as the temperature of the system rises. The movement of the lid triggers an electrical switch that is attached to it. The tiny capillary tube is frequently folded up for storage. This is exactly right. Please do not cut it. It is not a wire in any way. Aside from that, avoid bending the tube. It is possible for it to develop kinks and cease to function.

Figure 2.3 Thermostat [6].

A cable sensor is used in an electronic controlled system. The output can be used to operate a relay (which turns on and off) or a dimmer (infinitely variable control). The advantage of using an electrically controlled kind is that an extra temperature sensor may be added to the system. This sensor, which is glued to the baby's skin, monitors the temperature of the baby's core body temperature.

There should be a physical separation between the temperature display or thermometer and the temperature control device. Either a digital display or an alcohol glass thermometer can be utilized. Mercury thermometers are not permitted to be used in neonatal incubators for concerns of safety. Mercury is extremely poisonous.

- Humidity control

The warm air moistens the water in the water container as it passes over it. Controlling the humidity is as simple as closing and opening a deflector plate placed over the container. Other incubators include a water heater that increases the humidity in the water as it warms. The humidity level should be set between 40% and 90%. A hygrometer, either digital or analog, is used to determine the relative humidity. To minimize corrosive damage to the incubator, fill the humidifier only with pure water.

- Oxygen control

The majority of infant incubators come equipped with a hose connection for the addition of oxygen from an external cylinder, oxygen concentrator, or the central gas supply. The warmed and moisturized air is supplemented with oxygen in this case as well.

Alternatively, the increased oxygen is administered directly to the infant via a nasal cannula.

- Alarm and safety features

All electronically controlled infant incubators monitor all functions for safety purposes and emit an alert if something does not operate properly.
The following alarms are triggered in the event of a failure:

- Overheating. When the cabinet's air becomes very heated.
- Failure of a fan. When the fan stops whirling.
- Switch off the electricity. When the electricity supply is disrupted. This requires an extra alarm battery to operate.

Additionally, all incubators, including older non-electronic controlled models, have an overheating precaution. When the cabinet temperature hits 40°C, the heater is totally turned off.

2.3 Medical Device Industry in Thailand

The field of medical devices encompasses both medical devices and medical equipment, and is considered a high-value industry that accounted for 1.2% of GDP in 2020. The sector has experienced continuous growth due to the increasing number of patients and the aging population. As medical devices and equipment are essential for sustaining life, the sector has shown resilience to changes in economic conditions. Medical devices and equipment can be categorized into three main types based on their use [3]:

1.Single-use devices are typically used in general medical treatments and are not usually considered high-tech. These items are designed to be disposed of after a single use. Examples of single-use devices include syringes, hypodermic needles, tubes, catheters, cannulas, disposable gloves, and certain items used in dentistry or ophthalmology.

2.Durable medical devices generally have a lifespan of at least one year. Examples of durable medical devices include first aid kits, wheelchairs, medical beds, technical equipment used in medicine, surgery, and dentistry, as well as electrical diagnostic tools and x-ray machines.

3.Reagents and test kits encompass equipment used for diagnosing illnesses and conditions, as well as chemical kits used for testing patient samples. Examples of reagents and test kits include equipment for blood typing before dialysis, pregnancy tests, and HIV infection detection.

In the context of Thailand's medical devices industry, the majority of exports are comprised of single-use devices such as rubber gloves, catheters, syringes, and bandages, which account for 90.0% of the export value. The United States is the main export market for Thailand's medical devices, followed by Japan, the Netherlands, and Germany. Many of the manufacturers and exporters in Thailand are foreign-owned companies that utilize Thailand as a production base for exporting to other regions, such as Japan, the US, and France. On the other hand, imports into Thailand mainly consist of durable medical items and single-use devices, which together make up 80.7% of the total import value. Examples of these imported items include ultrasound equipment, x-ray machines, electrocardiogram (ECG) and electroencephalogram (EEG) monitors, and ophthalmological equipment. The major sources of imports are the United States, China, Germany, and Japan (as shown in figure 2.4).

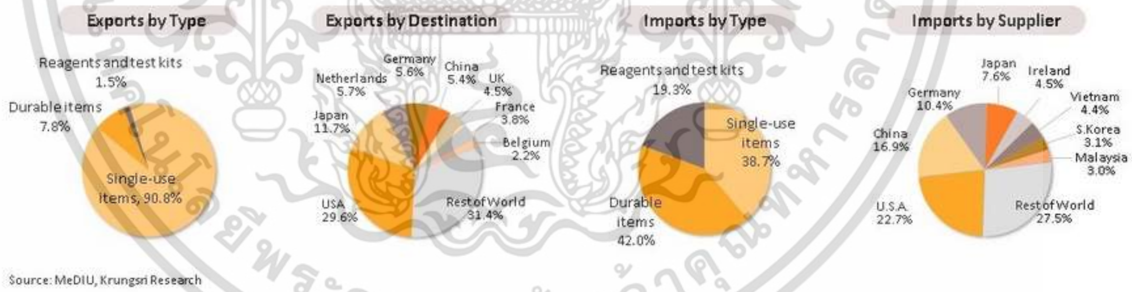

Figure 2.4 Trade Value of Thai Medical Device (2020) [3].

As of June 2021, there are 965 registered producers of medical devices with the Department of Business Development in Thailand. Among these producers, 98.0% are small and medium-sized enterprises (SMEs), which collectively account for 19.1% of the total revenue. The remaining 2.0% are large manufacturers, which command 80.9% of the total revenue in 2019 as shown in figure 2.5. Most of these large manufacturers are multinational companies (MNCs) such as Nipro (Thailand), Hoya Optics (Thailand), and Kawasumi Laboratories (Thailand), as listed in Tabble 2.1. Additionally, there are over 2,000 registered importers of medical devices according to data from the Food & Drug Administration [3].

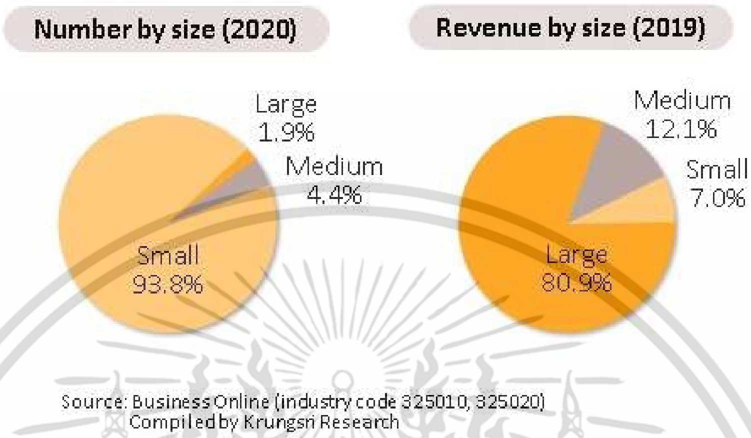

Figure 2.5 Thai Medical Device Manufacturers [3].

Table 2.1 Major Domestic Medical Device Manufacturers [3].

Company	Major share holders
Hoya Lens Thailand	Dutch
Nipro (Thailand)	Japanese
Hong Seng Knitting	Thai
Thong Thai Textile	Thai
Kawasumi Laboratories (Thailand)	Japanese
Reckitt Benckiser Healthcare Manufacturing (Thailand)	Thai
Dester	Dutch
M.E.Nikkiso	Japanese
Greiner Bio-One (Thailand)	Australian
Kendall-Gammatron	Luxembourgers

Source: Business Online, compiled by Krungsri Research

The majority of medical devices manufactured in Thailand do not necessitate advanced technology. The main production output comprises basic goods that utilize locally available raw materials such as rubber and plastic. Approximately 70% of the total production is intended for export. Producers can be categorized based on the intended use of the final product [3]:

1. Single-use devices: The segment of medical device producers in Thailand that manufacture single-use devices accounts for approximately 43% of the total number of producers, which has increased from 39% in 2019 (as shown in figure 2.6). This segment is known for high-potential and globally competitive products such as rubber gloves/medical gloves, as they do not require sophisticated technology for production. Thailand, being a major producer of rubber (a key raw material for rubber gloves), has targeted the export market for rubber gloves, which accounts for 90% of total sales of rubber gloves. Other competitive products in this segment include catheters and syringes, which are partially made of plastics.

2. Durable medical devices: Around 28% of the total medical device producers in Thailand are involved in manufacturing durable medical devices. Hospital beds, examination tables, and wheelchairs are among the more important products in this segment for manufacturing and export, as they do not require advanced manufacturing technology.

3. Reagents and test kits: Only 6% of the manufacturers in the medical devices industry in Thailand produce reagents and test kits, and these are mostly through joint ventures with multinational companies aiming to enter the Thai market. The products in this segment include test kits for diabetes, kidney disease, and hepatitis. Between 2015 and 2019, Thailand's exports of reagents and test kits have experienced a significant surge, increasing more than tenfold.

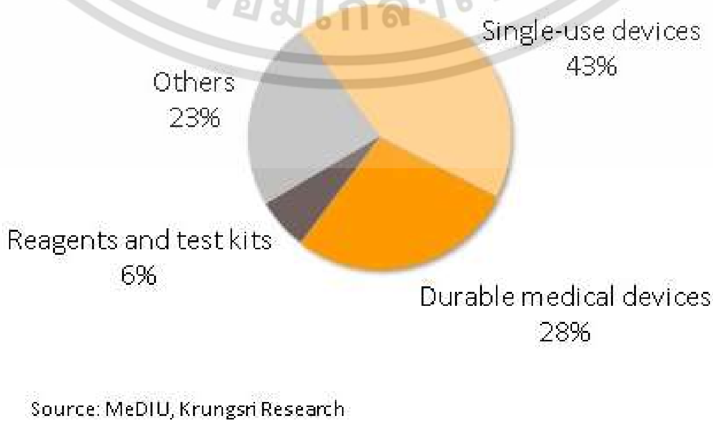

Figure 2.6 Thai Medical Device Manufacturers by Type (2020) [3].

2.4 IoT

The Internet of Things (IoT) refers to a system where interconnected computing devices, machines (both mechanical and digital), objects, animals, or even humans are equipped with unique identifiers (UIDs) and the capability to exchange data over a network without relying on human-to-human or human-to-computer interaction [8].

In the realm of IoT, a "thing" can encompass diverse entities such as a person with a heart monitor implant, a farm animal with a biochip transponder, a vehicle equipped with built-in sensors to alert the driver about low tire pressure, or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and is capable of data transfer over a network.

Many organizations across different industries are increasingly utilizing IoT to optimize their operations, gain deeper insights into customer behavior for improved customer service, enhance decision-making processes, and boost overall business value. [8].

An IoT ecosystem is a network of smart devices that are connected to the internet and equipped with embedded systems like processors, sensors, and communication hardware. These devices collect data from their surroundings and use web connectivity to send and receive data. The data collected by IoT devices is either analyzed locally or sent to the cloud for analysis through an IoT gateway or edge device. These devices may also communicate with other devices and act on the information received. Human intervention is not always required as the devices are designed to work autonomously, although people can interact with them for setup, instructions, or data access. The connectivity, networking, and communication protocols used in IoT depend on the specific applications being deployed. Additionally, IoT can leverage artificial intelligence (AI) and machine learning to streamline data collection processes and make them more dynamic [8].

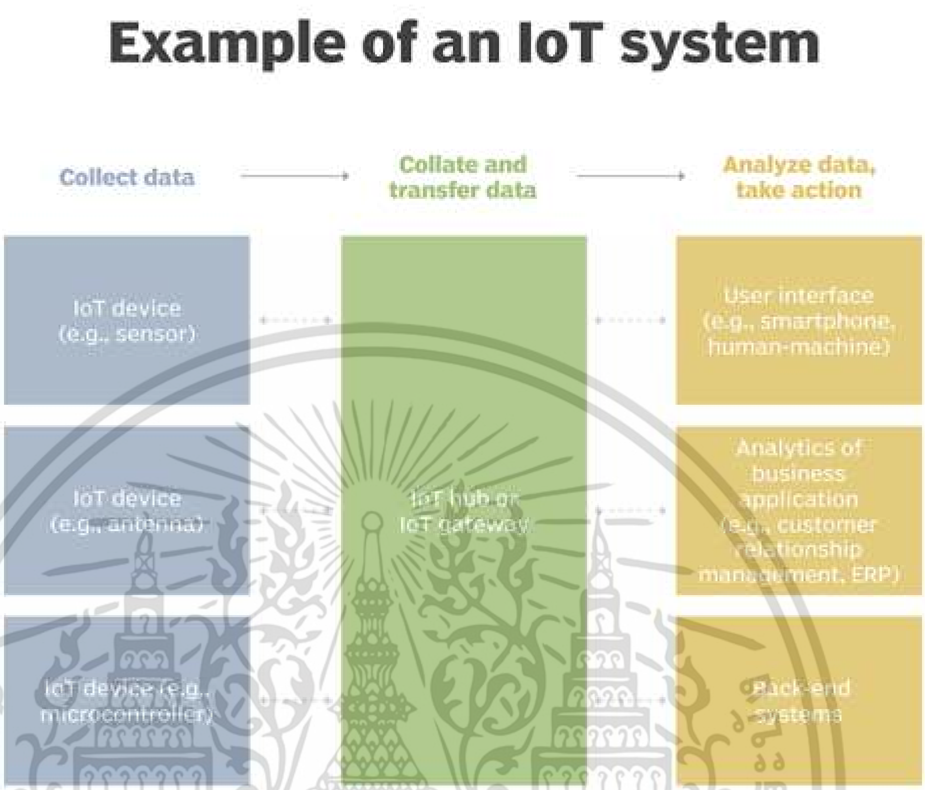

Figure 2.7 Example of an IoT system [8].

The Internet of Things (IoT) is supported by various emerging standards and frameworks that facilitate communication and interoperability among IoT devices. These standards include [8]:

1. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN): This open standard, defined by the Internet Engineering Task Force (IETF), allows low-power radios such as 804.15.4, Bluetooth Low Energy (BLE), and Z-Wave to communicate with the internet. It is commonly used in home automation and other IoT applications.

2. ZigBee: ZigBee is a low-power, low-data rate wireless network standard used primarily in industrial settings. It is based on the IEEE 802.15.4 standard and is supported by the ZigBee Alliance, which has developed Dotdot, a universal

language for IoT that promotes secure communication and interoperability among smart objects on any network.

3. LiteOS: LiteOS is a Unix-like operating system (OS) designed for wireless sensor networks. It supports various applications, including smartphones, wearables, intelligent manufacturing, smart homes, and the internet of vehicles (IoV). LiteOS also serves as a platform for smart device development.

4. OneM2M: OneM2M is a machine-to-machine service layer that can be embedded in software and hardware to connect IoT devices. It is a global standardization body created to develop reusable standards for enabling communication among IoT applications in different domains or verticals.

5. Data Distribution Service (DDS): DDS is an IoT standard developed by the Object Management Group (OMG) for real-time, scalable, and high-performance machine-to-machine communication.

6. Advanced Message Queuing Protocol (AMQP): AMQP is an open source published standard for asynchronous messaging by wire. It enables encrypted and interoperable messaging between organizations and applications, making it suitable for client-server messaging and IoT device management.

7. Constrained Application Protocol (CoAP): CoAP is a protocol designed by the IETF specifically for low-power, compute-constrained devices to operate in the IoT.

8. Long Range Wide Area Network (LoRaWAN): LoRaWAN is a protocol for wide area networks (WANs) that is designed to support large-scale IoT networks, such as smart cities, with millions of low-power devices.

These emerging standards and frameworks play a crucial role in enabling seamless communication and interoperability among IoT devices, facilitating the development and deployment of IoT applications across different domains and industries.

IoT applications

The Internet of Things (IoT) has numerous real-world applications across various verticals, including consumer IoT, enterprise IoT, and manufacturing and industrial IoT

(IIoT). These applications are seen in industries such as automotive, telecom, energy, and healthcare, among others.

In the consumer segment, IoT is widely used in smart homes where devices like smart thermostats, smart appliances, and connected lighting and electronic devices can be controlled remotely through computers and smartphones. This allows for convenient management of home devices, energy efficiency optimization, and enhanced comfort for users.

Wearable devices equipped with sensors and software are also prominent in consumer IoT. These devices collect and analyze user data, and can send messages to other technologies to make users' lives easier and more comfortable. For example, wearable devices are used for public safety, such as optimizing routes for first responders during emergencies or tracking vital signs of construction workers or firefighters in life-threatening situations.

In the healthcare industry, IoT offers significant benefits. It enables close patient monitoring through data analysis, allowing for improved patient care and outcomes. Hospitals and healthcare facilities use IoT systems for tasks such as inventory management for pharmaceuticals and medical instruments, helping to streamline operations and ensure efficient supply chain management.

Furthermore, IoT applications are also present in other industries such as manufacturing and industrial settings. IoT is revolutionizing industries by enabling real-time monitoring, automation, and data analysis to optimize production processes, enhance operational efficiency, and improve overall productivity. For example, IoT is used in smart factories for predictive maintenance, remote monitoring of equipment, and supply chain management, leading to cost savings and increased competitiveness.

Overall, the real-world applications of IoT are vast and diverse, spanning across various verticals and industries. From consumer IoT to enterprise IoT, and from healthcare to manufacturing and industrial IoT, IoT technologies are transforming industries and improving processes, leading to increased efficiency, convenience, and better outcomes [8].

2.5 Image Processing

Digital image processing refers to a set of techniques that utilize computer algorithms to manipulate digital images. It serves as a critical preprocessing step in various applications, including face recognition, object detection, and image compression. The purpose of image processing is to enhance existing images or extract significant information from them. This is particularly valuable in Deep Learning-based Computer Vision applications, where preprocessing can significantly enhance model performance. Additionally, image manipulation, such as adding or removing objects from images, finds applications in fields such as the entertainment industry [9].

A computer interprets digital images as matrices in either 2D or 3D. Each pixel in the matrix represents the intensity of the image, and this intensity is measured in amplitude. The common practice is to use 8-bit images where the amplitude value falls between 0 to 255.

Figure 2.8 How Machines See Images [9].

Therefore, digital images are perceived by a computer as a function, represented as I(x,y) or I(x,y,z), where "I" is the intensity of the pixel and (x,y) or (x,y,z) represent the coordinates of the pixel in the image, depending on whether it's a binary/grayscale or RGB image. Different types of images are managed by computers based on their functional representations.

1. Binary Image

A binary image is an image that has only two possible values for pixel intensity: 0, which represents black, and 1, which represents white. These types of images are often used to emphasize a specific area of a colored image, and they are particularly useful for image segmentation tasks, as shown in figure 2.9.

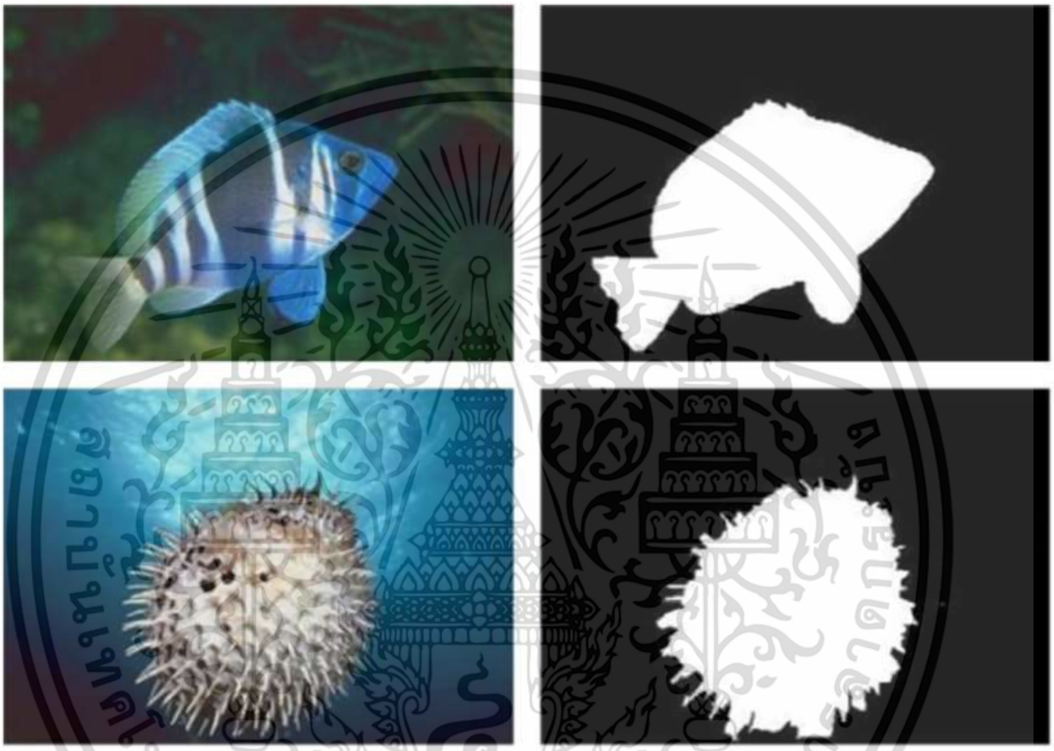

Figure 2.9 Binary Image [9].

2. Grayscale Image

Grayscale or 8-bit images consist of 256 different colors, where a pixel intensity of 0 corresponds to the color black, and a pixel intensity of 255 corresponds to the color white. The other 254 values in between represent different shades of gray. When an RGB image is converted to grayscale, the resulting image will still have the same histogram shape as the original RGB image (as shown in figure 2.10).

Figure 2.10 Grayscale Image [9].

3. RGB Image

Computers interpret colored images as 16-bit matrices with 65,536 possible colors for each pixel, which is known as RGB or Red, Green, Blue channels. Unlike the previous types of images, an RGB image has three matrices or channels, each with values ranging from 0 to 255, that are stacked on top of each other. Thus, three coordinates are required to specify a matrix element's value, unlike the previous images that only required two coordinates. A pixel is considered black when the pixel value is (0, 0, 0) and white when it is (255, 255, 255), and any combination of numbers in between results in all possible colors. For instance, the color red is represented by (255, 0, 0), while green and blue are represented by (0, 255, 0) and (0, 0, 255), respectively. In an RGB image, each channel corresponds to a primary color (red, green, or blue), and a combination of these channels determines the final color of each pixel. An illustration of an RGB image split into its channel components is provided, and the histograms' shapes for each channel are distinct (as shown in figure 2.11).

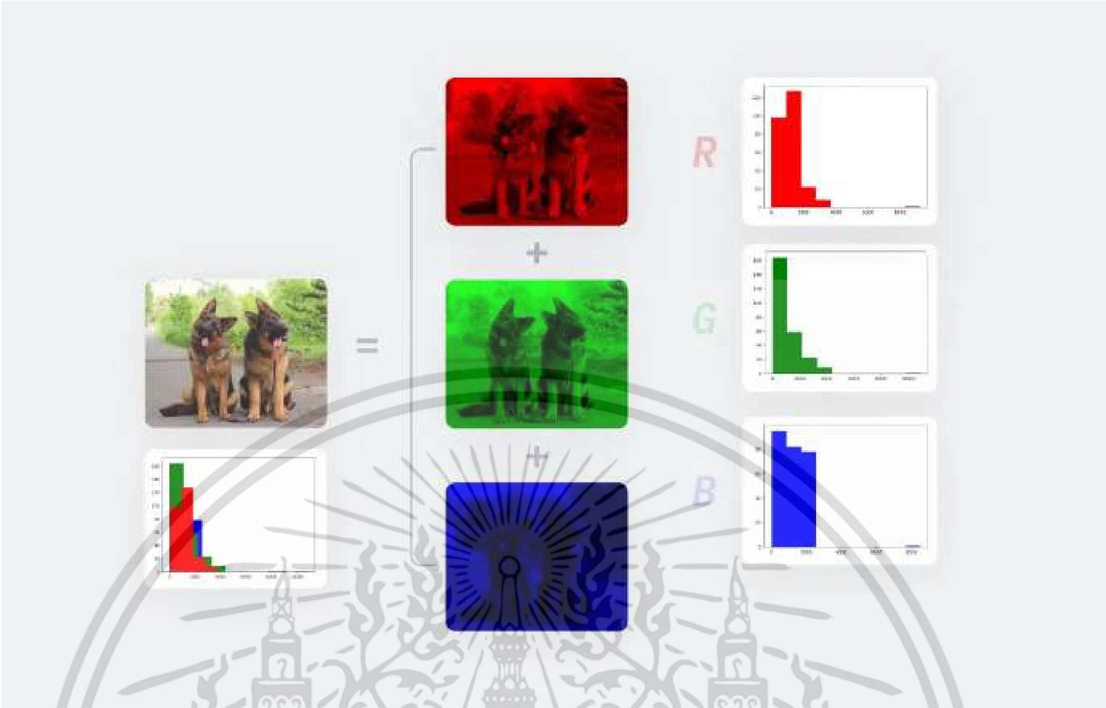

Figure 2.11 RGB Image [9].

4. RGBA Image

An RGBA image is an RGB image that includes an additional channel called "alpha" which represents the opacity of the image, ranging from 0% to 100%. Opacity is a measure of how much light passes through an object, with transparent materials having a high opacity and opaque materials having a low opacity. The alpha channel in an RGBA image is used to simulate this property. The following example demonstrates this concept (as shown in figure 2.12).

Figure 2.12 RGBA Image [9].

Phases of Image Processing

The process of Digital Image Processing involves a series of fundamental steps that are commonly followed in various applications of image processing. These steps are Image Acquisition, Image Enhancement, Image Restoration, Color Image Processing, Wavelets and Multi-Resolution Processing, Image Compression, Morphological Processing, Image Segmentation, Representation and Description, Object Detection and Recognition, and Knowledge Base [9].

In the first step, the image is captured and digitized for further processing. The acquired image is then manipulated in the second step using various techniques to meet the specific requirements of the task for which the image will be used. Image enhancement is subjective in nature, while image restoration is an objective operation aimed at improving the appearance of the image.

Color image processing is used to handle processing of colored images, while wavelets and multi-resolution processing are used to represent images at various degrees of resolution. Image compression is used to reduce the size of the image, which is useful for transferring images to other devices or for storage constraints.

Morphological processing provides the tools to extract image components that are useful in the representation and description of shape. Image segmentation involves partitioning the image into different key parts to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Representation and description extract attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object detection and recognition involves assigning a label to the object, such as "vehicle" or "person", and the knowledge base contains information that helps in solving the problem for the specific task at hand. Overall, Digital Image Processing provides a structured approach for processing images that is widely used in various fields, including medical imaging, remote sensing, and computer vision.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This chapter will describe materials and methodology in detail, including the overall process (section 3.2), the entirety of the incubator's design (section 3.3), definition and methodology of each image processing used in this thesis (section 3.4) and the details of materials used in control system (section 3.5). This project is divided into 2 main parts including monitoring system and control system. This thesis primarily focusses on the control system, a crucial component of an infant incubator that requires careful consideration due to several pertinent issues. Furthermore, the monitoring system employed in this study differs from conventional commercial incubators as it incorporates image processing and software programming to enhance the functionality of the device.

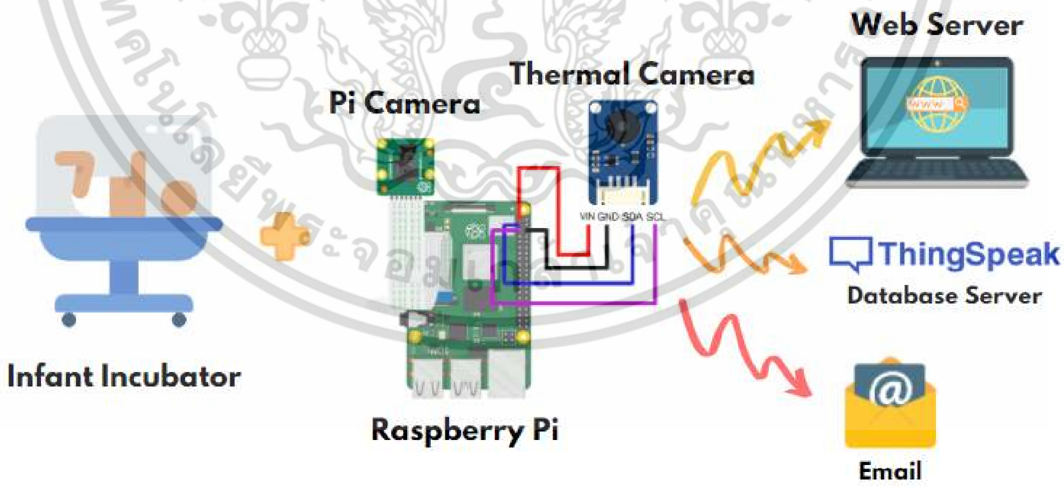

Figure 3.1 Diagram of Smart infant incubator.

3.2 Action Plan

Table 3.1 presents the research plan.

Action plan	Research progress per 2 months											
	2	4	6	8	10	12	14	16	18	20	22	24
Study related research												
Calibrate thermal camera with temperature sensor												
Design software program for monitoring system												
Design and create incubator												
Design circuit of control system												
Test operation of control system												
Record and evaluate temperature control and compare with the commercial model												
Improve the operation of control system												
Record and evaluate control system with incubator analyzer												
Develop software program for monitoring system												

3.3 Designing of infant incubator

The aim of this thesis is to develop a smart infant incubator, with the purpose of demonstrating the viability of manufacturing such equipment domestically at an economically feasible price point. This thesis will mainly focus on the design and implementation of advanced monitoring and control systems. The model for our incubator's design has been referenced by the YP-90A incubator model in our department laboratory, based on the standard shape observed in most conventional incubators. Notably, the transparent cabinet component of our incubator is constructed using acrylic material with size of 40x85x45 cm^3 as shown in figure 3.3.

Figure 3.2 YP-90A incubator [10].

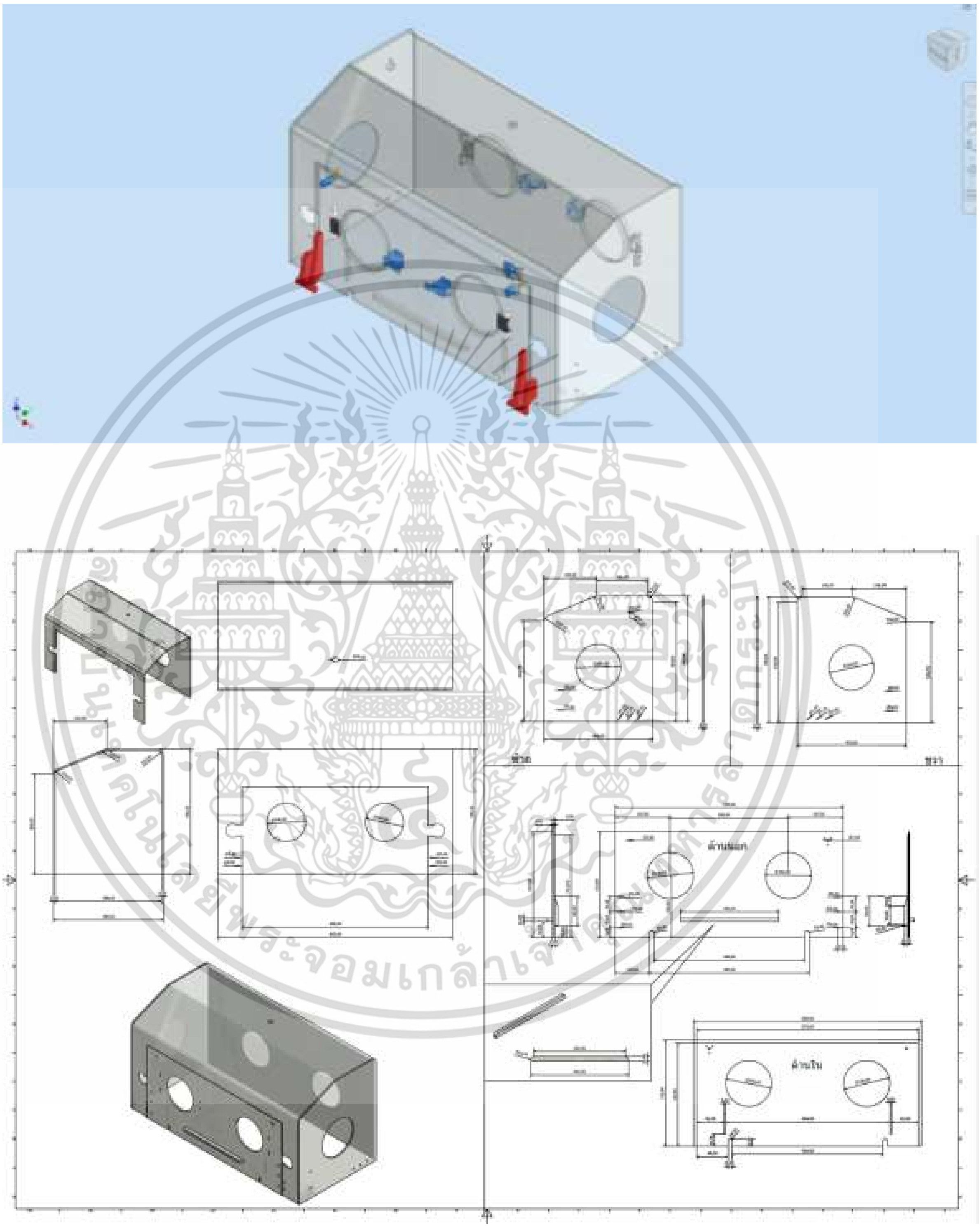

Figure 3.3 Designing of the transparent cabinet.

The transparent cabinet of the incubator features four circular openings and a door equipped with two additional circular openings as shown in figure 3.3. These openings are designed to facilitate the insertion of hands into the cabinet for operational purposes. Each circular opening is intended to be covered with an acrylic circular door (as shown in figure 3.4 and 3.5) to prevent air from escaping, thereby maintaining the desired environmental conditions inside the incubator.

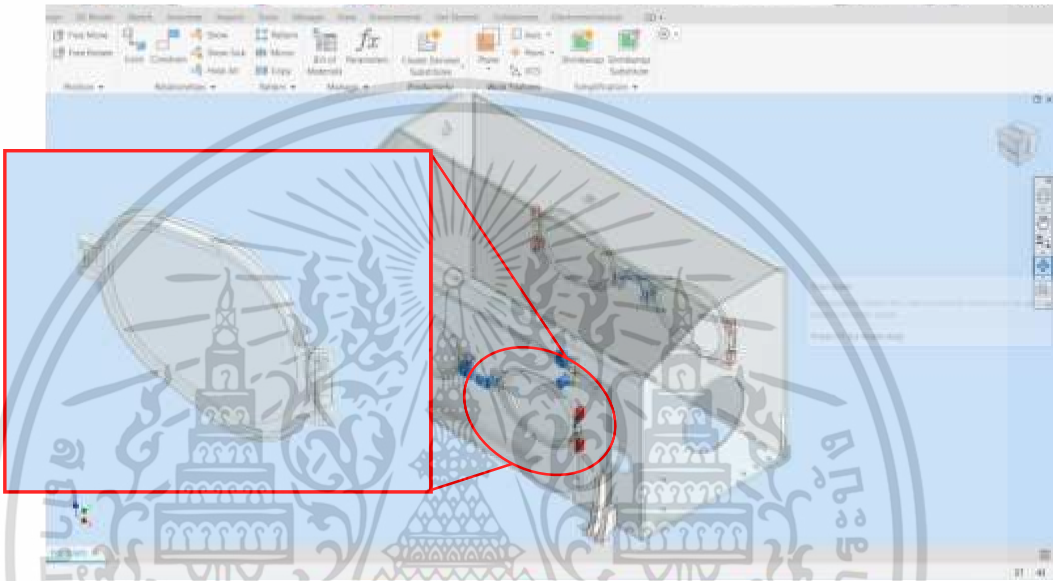

Figure 3.4 Designing of the door with same shape as the model.

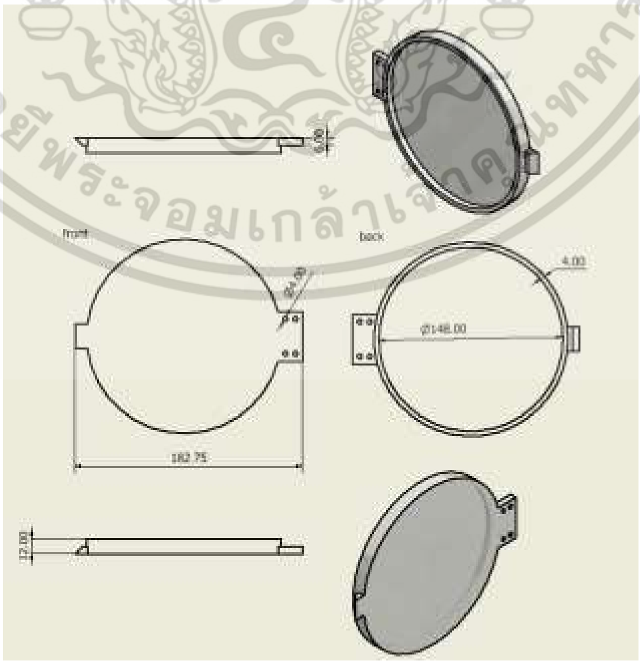

Figure 3.5 Designing of the hand's door.

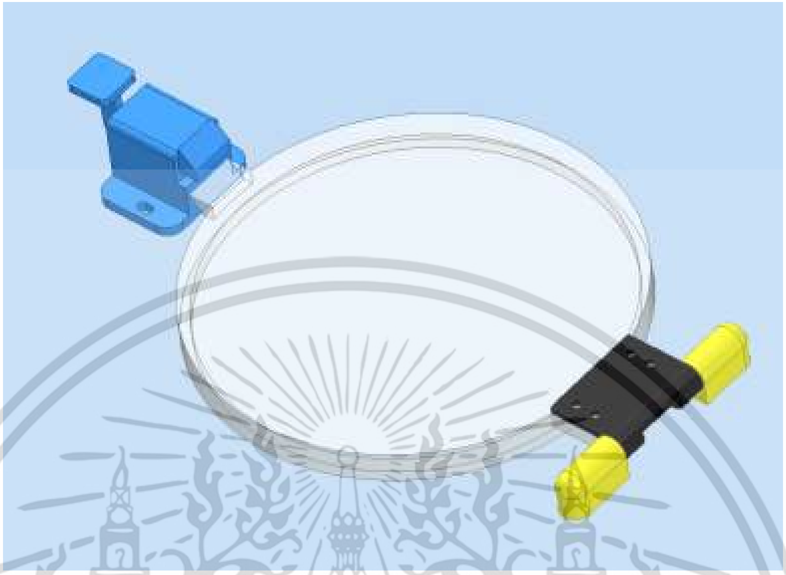

Figure 3.6 Designing of the hand's door.

Every acrylic door is equipped with a lock and hinges that are fabricated using 3D printing technology and PLA (polylactic acid) material. The utilization of 3D printing allows for precise manufacturing of these components to ensure smooth operation of the locks and hinges, while PLA material provides durability and stability to support the functionality of the doors.

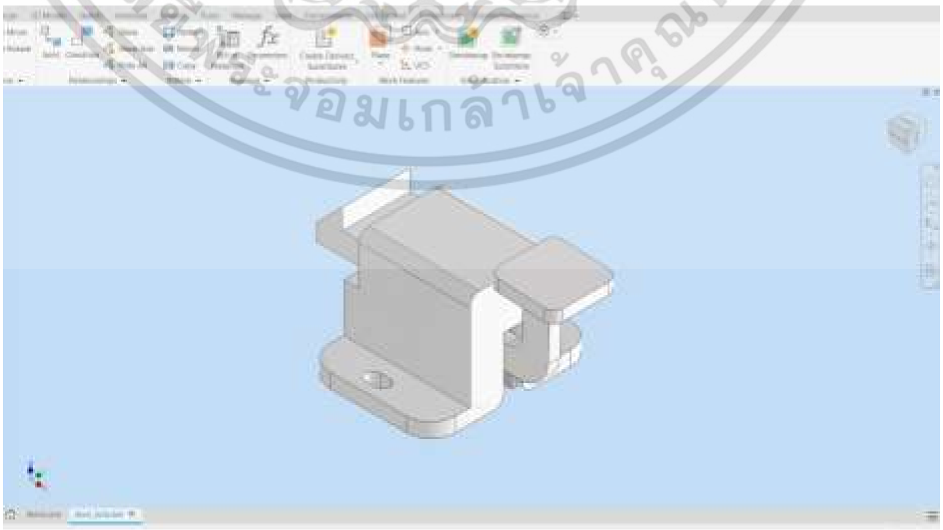

Figure 3.7 Designing of door lock.

Lastly, to enable the admission or transfer of a newborn, we have utilized stainless steel hinges that are capable of supporting the weight of the acrylic door. The design of these hinges, as shown in figure 3.9, resembles the model, and they have been selected to ensure ease of opening and closing operations while maintaining structural integrity.

Figure 3.8 Designing of the cabinet with the hinges.

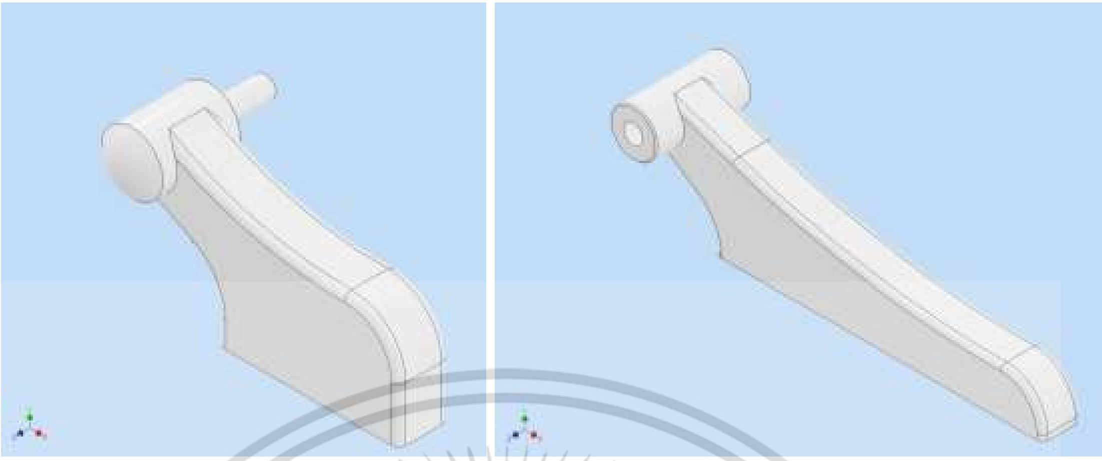

Figure 3.9 Designing of the hinges.

Figure 3.10 Designing of the hinges.

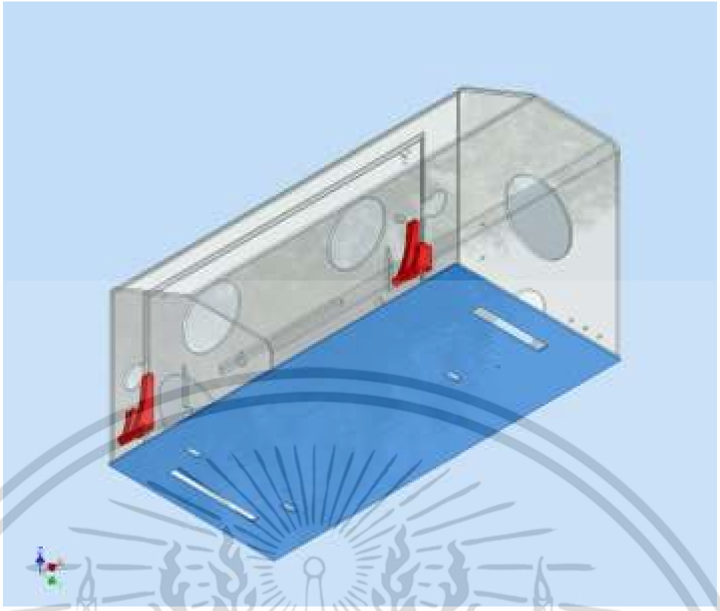

Figure 3.11 Designing of the cabinet with base part.

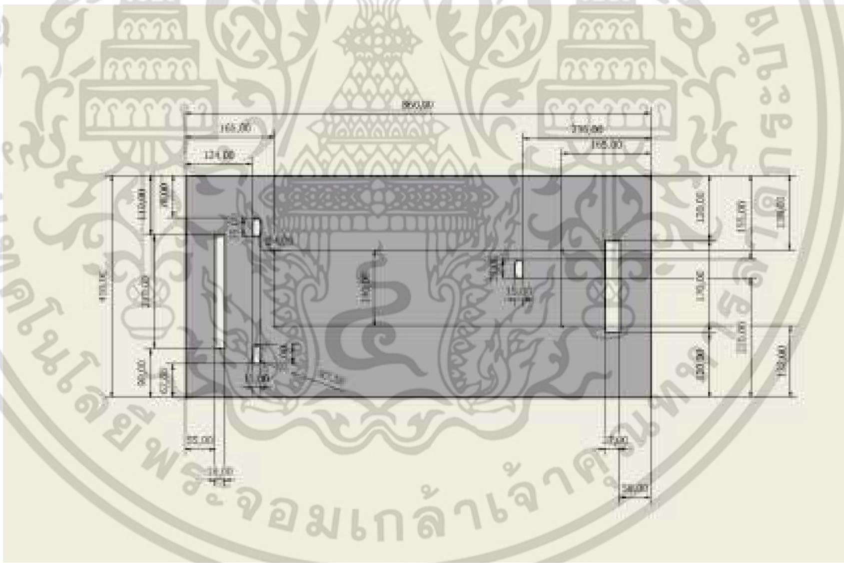

Figure 3.12 Designing of the base part.

Furthermore, the base part of the incubator operates as a protective barrier between the control system and the infant placed inside the cabinet, as shown in figure 3.11. The base is constructed from aluminum material with a thickness of 5 mm, chosen for its ability to withstand the heat generated by the heater in the control system. Lastly, the base part which places the control system, is also constructed from aluminum to ensure optimal thermal management and structural stability.

3.4 Monitoring System

We intend to add more monitors to make it different from other incubators, introducing advanced image processing techniques to enhance the system's image quality and detail. This innovation sets our monitoring system apart from typical commercial incubators, which are equipped with patient monitors that display basic parameters. Instead, our system operates similarly to a surveillance, utilizing sophisticated image fusion and background subtraction techniques to extract crucial information, will be explained in section 3.4.2. The monitoring system consists of a Raspberry Pi acts as a controller and connects to two cameras; pi camera and thermal camera. The programming language we use is Python which is frequently employed for the purpose of image processing due to its broad usage. Its vast collection of libraries and tools are instrumental in accomplishing image processing tasks with remarkable efficiency [11]. In our system, we have also incorporated OpenCV, which stands for Open Source Computer Vision. It is a frequently used tool in computer vision and image processing tasks, with a variety of applications including but not limited to face detection, video capture, tracking moving objects, and object disclosure. More recently, OpenCV has also been utilized in COVID-related applications such as face mask detection and social distancing [12]. In addition to the image processing, we have used to develop the system, we have displayed the data in various formats, online stream, database and email, which explains section 3.4.2.

We divided the software process into two parts: video processing and result display as shown in figure 3.13.

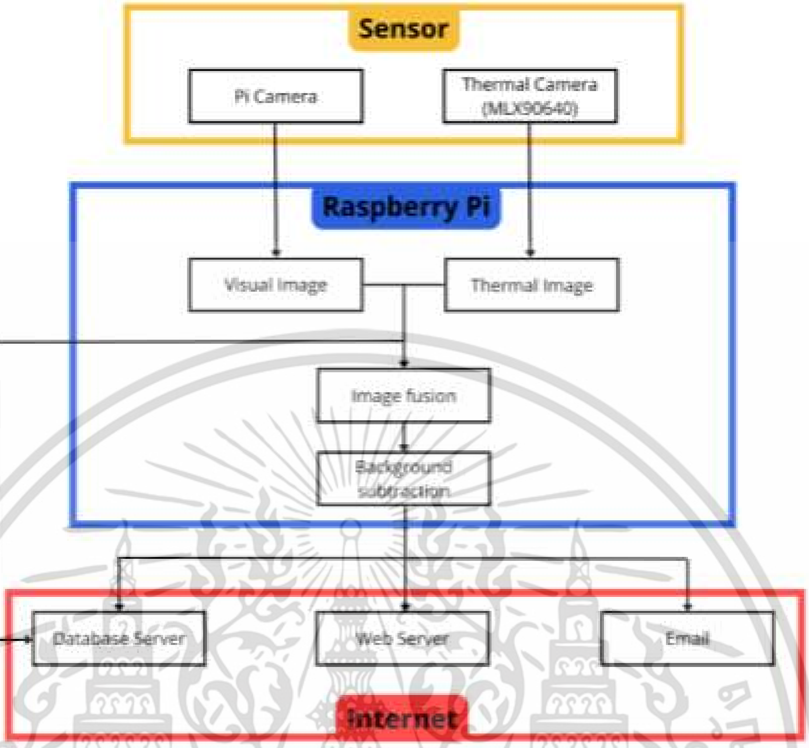

Figure 3.13 Operation diagram of the monitoring system.

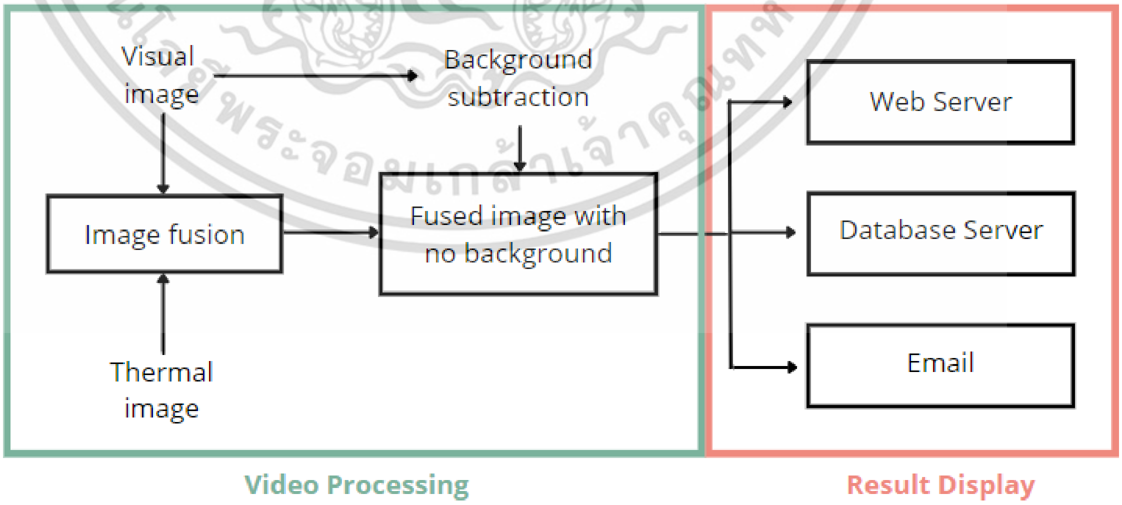

Figure 3.14 Diagram of the monitoring system.

3.4.1 Video Processing

The video processing aspect of our project involved enhancing the quality of the video, measuring temperature data and transmitting both the video and measured data over the internet. OpenCV was our primary tool in this process. Initially, we utilized the MLX90640 thermal camera to capture temperature readings and combined this with video footage from the Pi camera module using image fusion techniques, resulting in a clearer video output. We then implemented background subtraction techniques to eliminate the background and focus solely on the object being measured for temperature.

i. Image Fusion

Image fusion is a burgeoning area of research that involves the integration of images acquired from various sensors to produce a more informative image for decision-making purposes. The integration of different images can enhance both the analytical and visual quality of the resulting image. Effective image fusion techniques are capable of extracting all relevant information from the input images while maintaining consistency in the output image. The fused image is thus better suited for both machine and human perception, as it offers a more comprehensive representation of the original data [13]. We used the image fusion technique to combine the visual image from Pi camera module and the thermal IR image from the MLX90640 together as shown in figure 3.15.

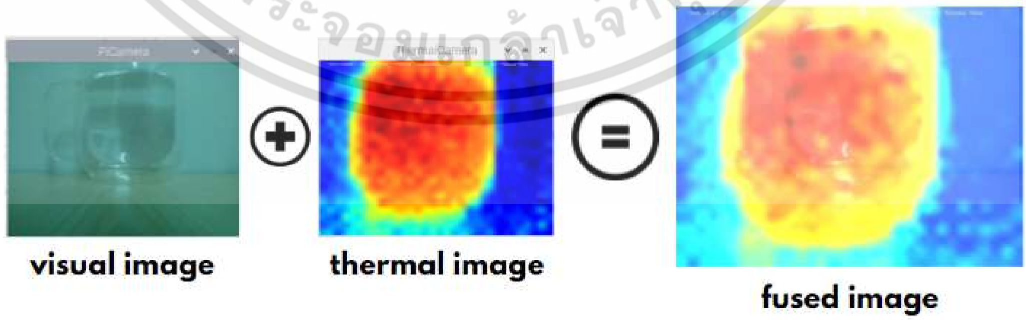

Figure 3.15 Image fusion technique.

ii. Background Subtraction

The technique of background subtraction is commonly employed for detecting motion in a sequence of frames captured by static cameras. It involves comparing the current frame to a reference frame (referred to as the "background image" or "background model") to identify moving objects. The primary objective of this technique is to detect the foreground objects in the video frame. For certain applications, it is unnecessary to analyze the entire sequence, and instead, attention is focused on specific objects in the foreground. Once preprocessing steps such as denoising and morphological processing have been completed, object localization is carried out using foreground detection. All current detection techniques rely on modeling the background of the image, i.e., establishing the background and identifying any changes that occur. Defining an appropriate background can be challenging when the scene contains shapes, shadows, and moving objects. In all techniques, it is assumed that stationary objects may change in color and intensity over time [14].

To measure the temperature solely on the object of interest, we utilized background subtraction by removing the image's background. Figure 3.16 illustrates the process of background subtraction, which commenced by storing the initial frame, representing the background image without any object. This image was then subtracted from the original image to remove the background. The next step involved converting the resulting background-subtracted image into a binary or black and white image, which was then combined with the fused image. As a result, the resultant image only contains thermal data for the focused object.

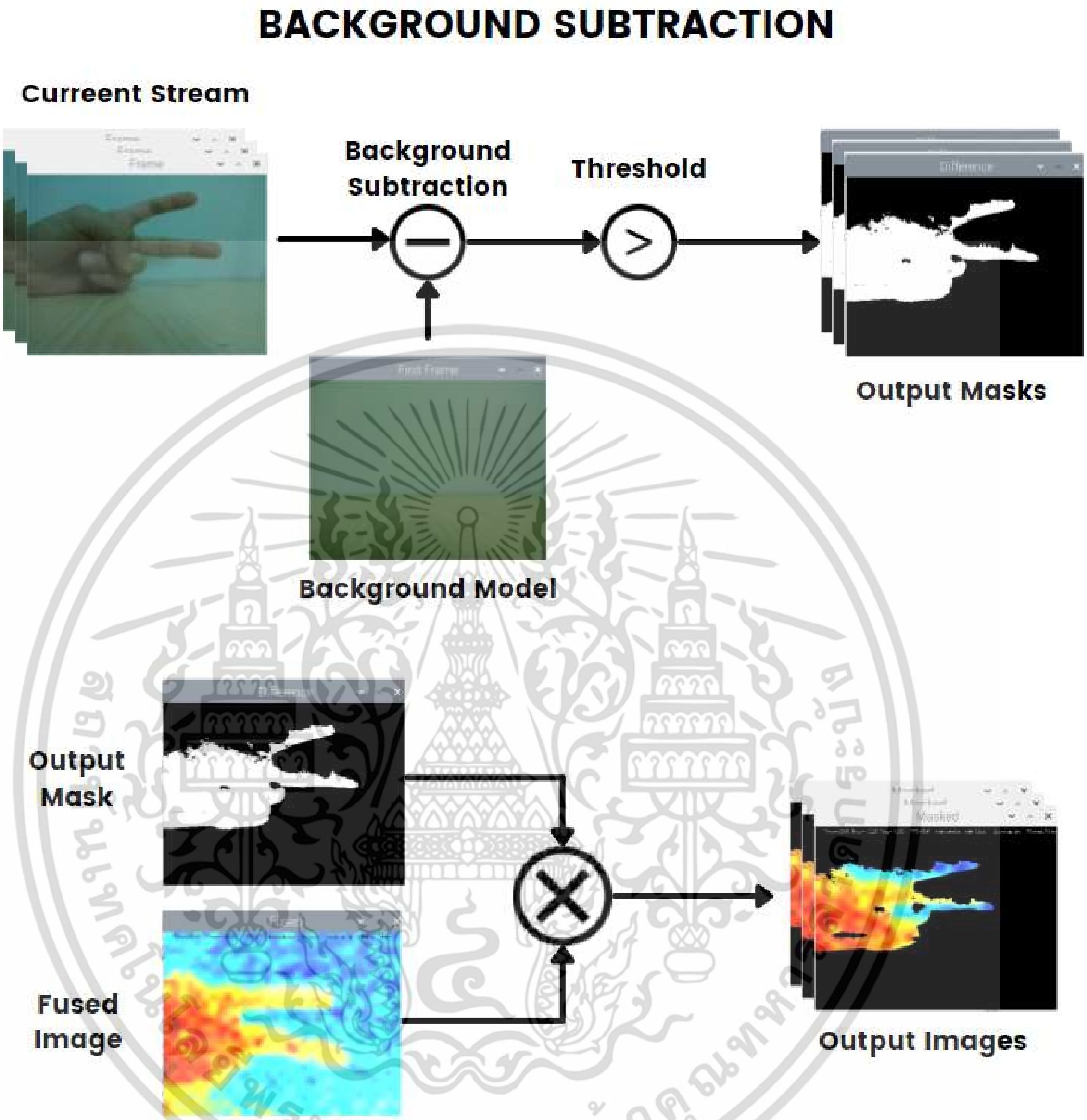

Figure 3.16 Background subtraction technique.

3.4.2 Result Display

i. Webserver

We integrated the OpenCV code for image processing with a simple Flask server, a Python-based web framework, is known for its compactness and lightness, along with the provision of a range of tools and features that aid in the development of web applications in Python. The design of Flask offers developers greater flexibility, and as it allows the creation of web applications using a single Python file, it is particularly suitable for those new to web development. Additionally, Flask is a highly customizable framework, allowing developers to structure their code in a way that suits their needs, without requiring adherence to specific directory structures or complex boilerplate code [15-16]. To achieve this, we utilized the streaming web server code from The PiThermalCam Project [17] and modified it to incorporate our OpenCV image processing code. Setting up the web server involved several components, including the html code for webpage layout, the primary Python code for image processing using OpenCV, streaming web server code, and a folder for storing images, as illustrated in figure 3.17.

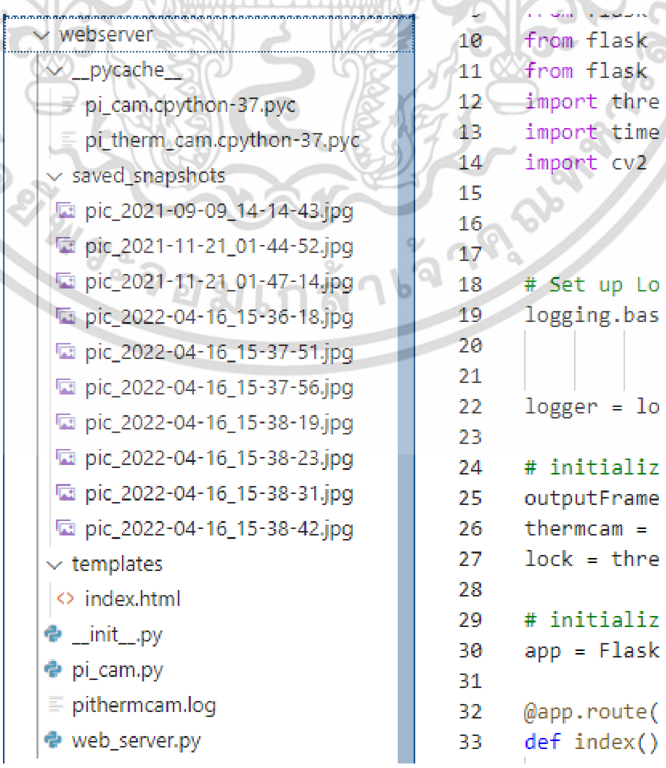

Figure 3.17 The folder of webserver code.

ii. Database Server

To enable online monitoring of the temperature data obtained from the image, we transmitted the data to ThingSpeak which is a cloud-based IoT analytics service that enables the aggregation, visualization, and analysis of live data streams. By using ThingSpeak, users can view visual representations of data posted by their devices in real-time. Additionally, the ability to execute MATLAB® code within ThingSpeak allows for online data analysis and processing as it is received. Typically, ThingSpeak is employed for prototyping and testing IoT systems that necessitate analytics capabilities [18-19]. The temperature readings we transmitted consisted of data from the fused image and background-subtracted image. Initially, we created a dedicated channel on the ThingSpeak website (as shown in figure 3.18) for our project to store the temperature data. On the channel page, ThingSpeak provided us with an API key which we incorporated into our code to access the channel, as shown in figure 3.19.

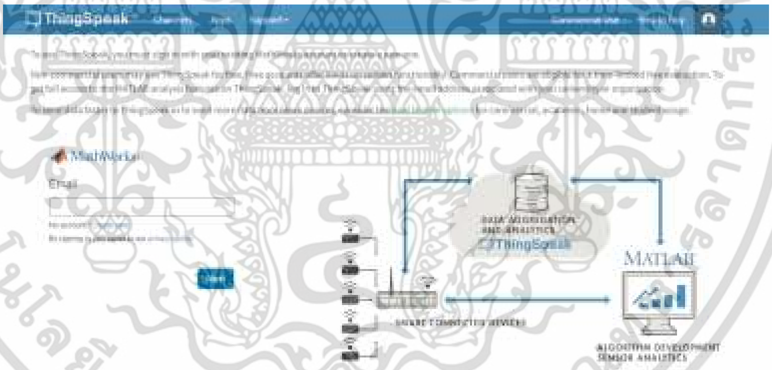

Figure 3.18 ThingSpeak.

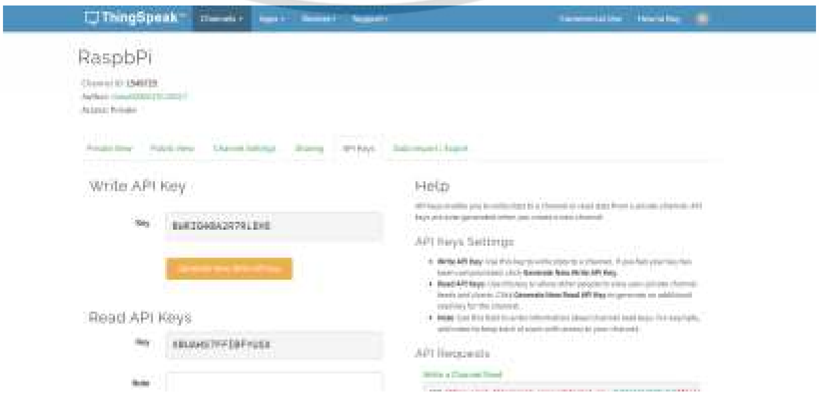

Figure 3.19 API Key of ThingSpeak.

iii. Email

This email function we planned to use for alarming the doctor or someone who is involved in taking care of a newborn baby when their temperatures are abnormal. Firstly, we create email content in code and send the email using Google / Gmail service as the mail server [20]. Then, permission for "less secure apps" (figure 3.20) must be enabled in order to access a Gmail account from an external device like the Raspberry Pi. In our Python code, we added smtp library reference for the email tools.

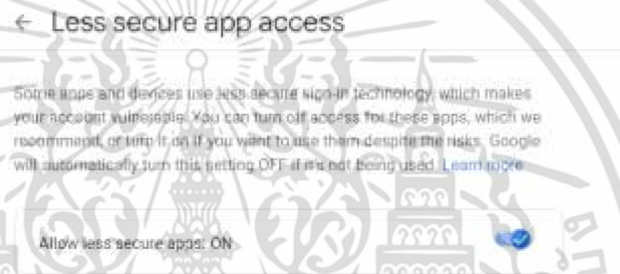

Figure 3.20 Set Gmail permissions.

3.5 Control System

There are various types of infant incubators that may be used depending on the needs of the infant. These include open-box incubators which provide heat from underneath but are open in other areas; closed-box incubators with a filtration system to prevent moisture loss and reduce infection risks; double-wall incubators offering enhanced protection from heat and moisture loss; servo-control incubators that use attached sensors to regulate temperature and humidity levels; and transport incubators designed for moving infants between hospital locations. The primary objective is to maintain a suitable temperature and environment for the infant. To achieve this, we propose constructing an infant incubator with a simple circuit, comprising PID controller, blower, heater, solid state relay (SSR) and thermostat, as described in the below diagram. Those components are placed inside incubator under the cabinet. A blower circulates filtered ambient air around a heater and a container of water. The moistened and heated air is now being drawn into the above cabinet.

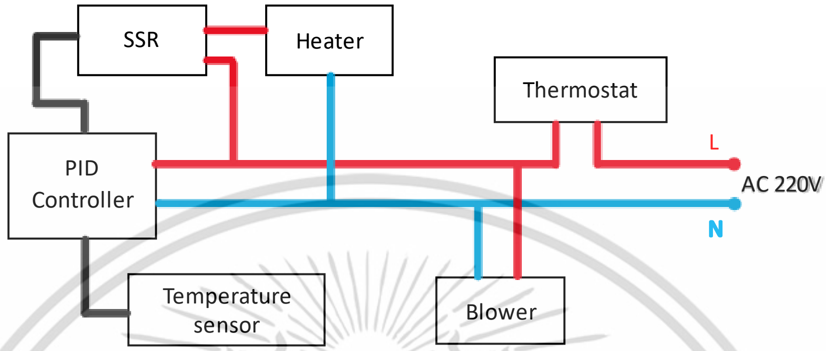

Figure 3.21 Wiring circuit of the control system.

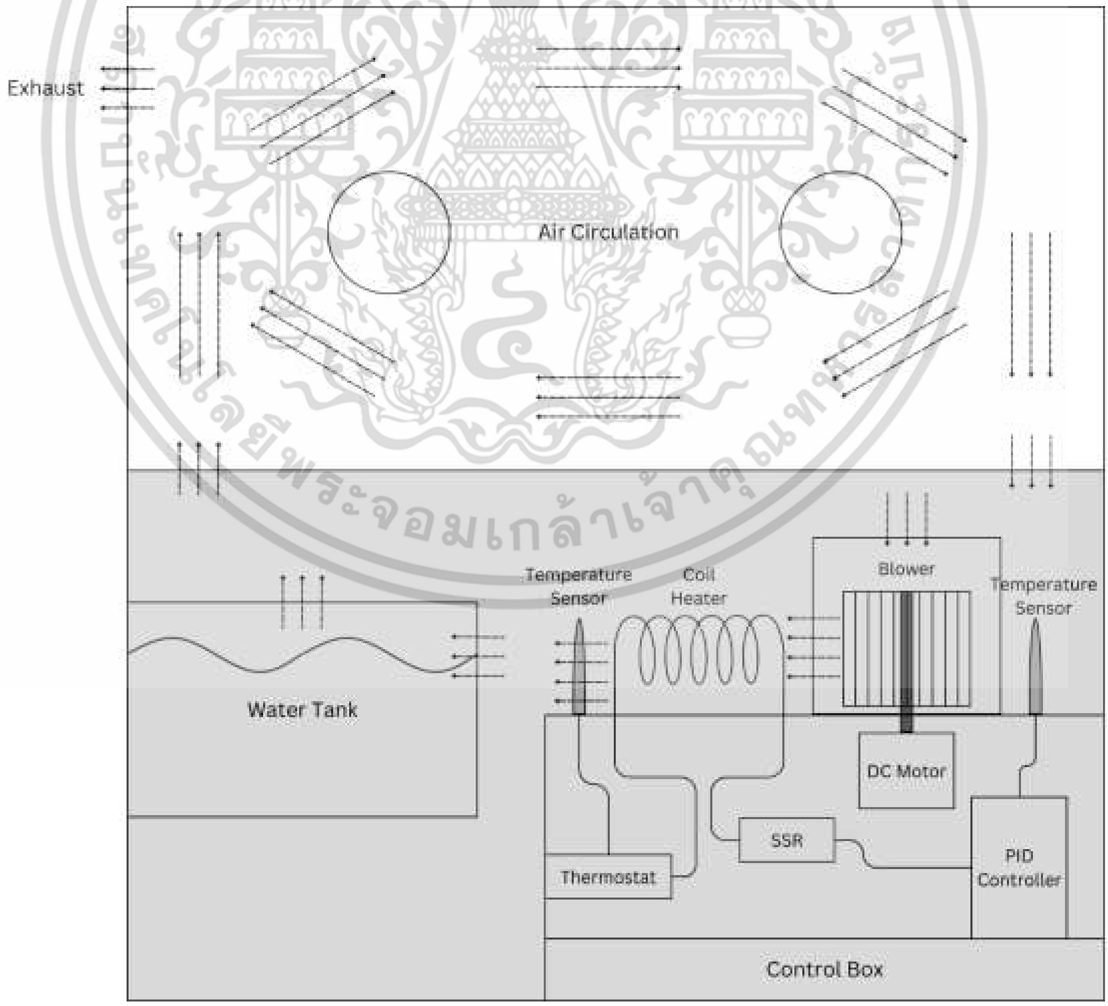

Figure 3.22 Incubator Airflow Diagram.

- PID controller

We use a PID controller named AI-518P ARTIFICIAL INTELLIGENCE INDUSTRIAL CONTROLLER (as shown in figure 3.23) to control the temperature and humidity in our incubator. Firstly, we set the controller with the temperature setting at 37 °C. This controller uses PID control technique to maintain temperature inside the incubator at setting point.

Figure 3.23 AI-518P Artificial Intelligence Industrial Controller [21].

Since the control is ready-made, you can set the modes and desired temperature for automatic calculation. However, the downside is that the controller is only able to show temperature readings and it can be challenging to fix any issues because the machine's operating system is complex and identifying the faulty component is difficult. By developing our own control system, we can incorporate additional parameters like humidity and noise, which will simplify the troubleshooting process. Additionally, a custom control system can be designed to be user-friendly for easier operation.

- Blower

The blower circulates filtered room air over or through the heater. Without the blower, heat cannot be directed away from the heater and the heater would overheat, resulting in the incubator being too hot to handle.

Initially, an AC motor was chosen as it aligned with the requirements of the AC circuit that was to be utilized. The necessary features of the motor we desired were high revolutions, low noise, and long-term usability. However, the motor that was selected encountered problems, and alternative motors did not resolve the issue.

Specifically, the motor was excessively noisy, heavy, and prone to overheating, causing it to shut down after just two hours of operation. Consequently, a decision was made to substitute the AC motor with a DC motor, which provided lower noise levels at the same RPM, was lightweight, and was capable of operating for extended periods without overheating. However, it was necessary to incorporate a current converter, a switching, into the circuit to link the DC motor since it required a different type of switching than the previous motor. We utilized a DC motor with a specification of 12 Volts, a power output of 30 Watts and RPM 2000.

Figure 3.24 DC Motor 12V.

- Heater

We used a coil heater to heat up the air The heater is controlled by an SSR which is connected to the controller, when the temperature is above the setting point, the controller will signal the SSR to block current then the heater will stop working until the temperature falls below the set point. We utilized a coil heater with a specification of 220 Volts and a power output of 400 Watts.

- Thermostat

Other than the alarm from the controller, we used a thermostat as an over-heating protection for safety purposes. Thermostat will switch off the circuit when the temperature above the heater reaches 80°C to avoid overheating inside the circuit box.

CHAPTER 4
EXPERIMENTAL RESULT

4.1 Introduction

In this project, the purpose of this studying is to build the infant incubator and adding the monitoring system base IoT to make it be the smart device. This chapter presents an overview of the monitors and controls used in the incubator, along with their performance and cost. Specifically, we will focus on the control system, which serves as the core component of the incubator. To evaluate the performance of the incubator, we conducted a comparison with conventional incubators and calibrated the system using analyzers that are typically used with conventional devices.

Figure 4.1 Overview of the incubator.

4.2 Monitoring System

Before using the thermal camera, we test it by testing with hot water and then compare the temperature data from the thermal camera and temperature sensor as shown in figure 4.2.

Figure 4.2 Results of testing the thermal camera with temperature sensor.

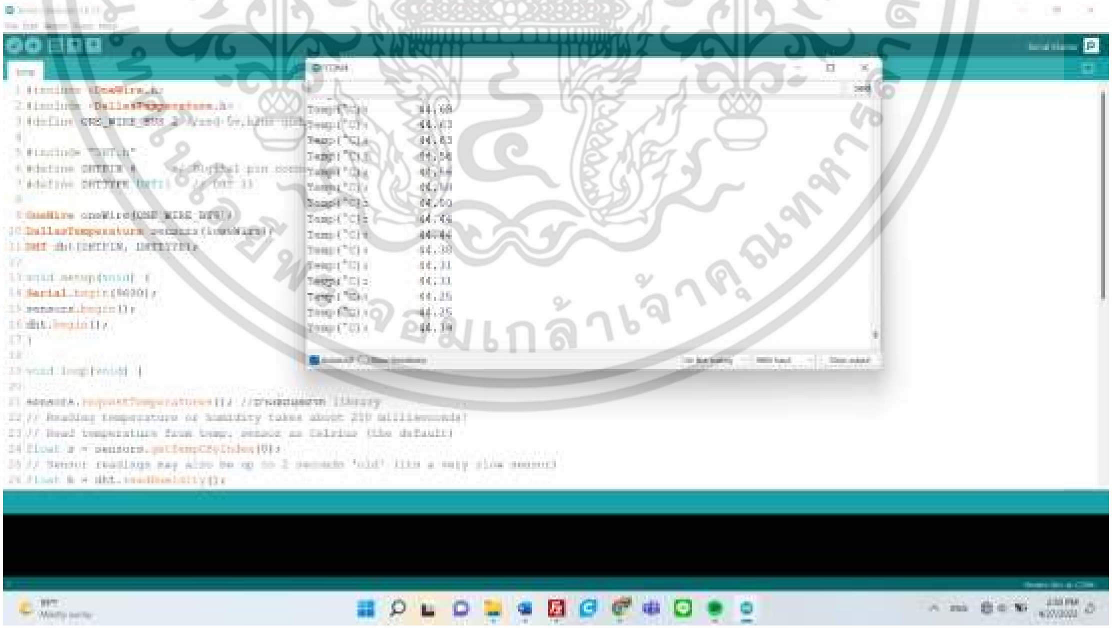

Figure 4.3 Results of temperature sensor show in serial port of Arduino.

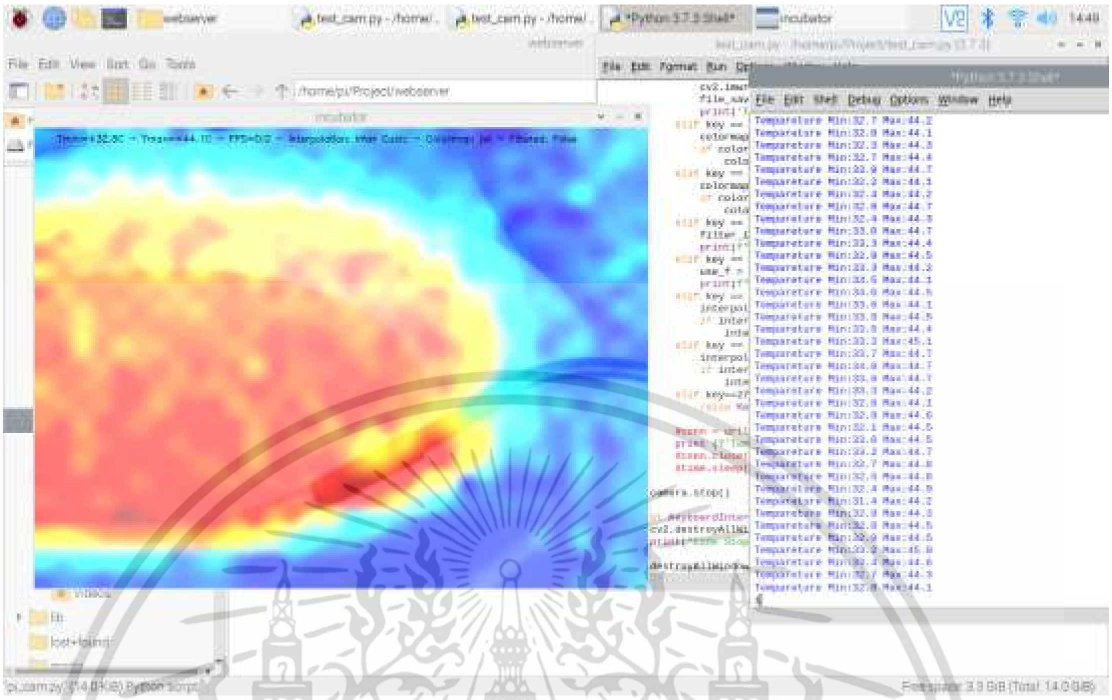

Figure 4.4 Results of thermal camera show in Raspberry Pi.

- Error calculation

Further investigations are performed to ensure that the measured measurements are accurate and reliable. In this section, we compared the thermal camera and temperature sensor. The percentage error is measured using this formula.

$$Percentage\ Error = \frac{|E - T|}{|T|} \times 100$$

Where E is the experimental value and T is the theoretical value.

In the same period of time, we used the temperatures measured from both camera and sensor and calculated average to find the error. The average of data from camera is 44.33 ℃ and from sensor is 44.388 ℃ so, the percentage error is 0.131%.

4.2.2 Results of Video processing

i. Image Fusion

In the video processing component, we first combine images from two cameras using image fusion, and the program calculates and displays the minimum and maximum temperature on the screen.

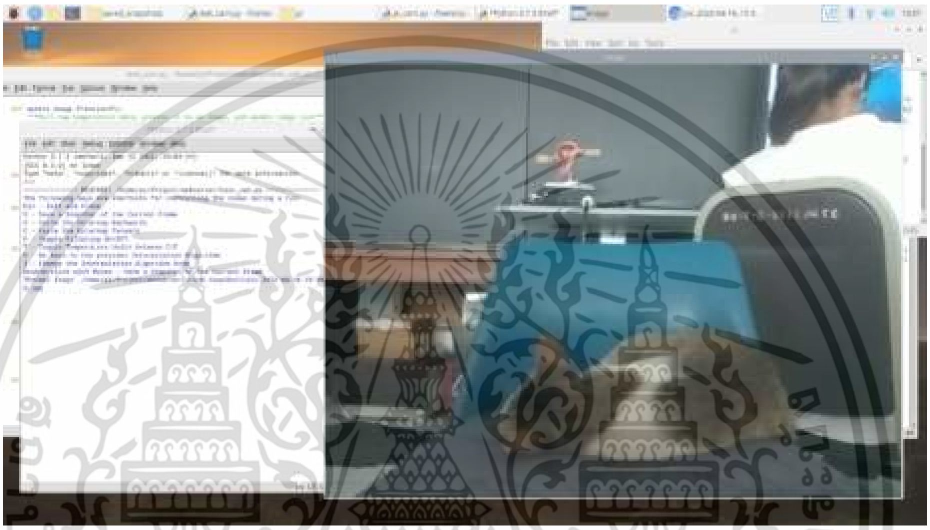

Figure 4.5 Result of Pi camera.

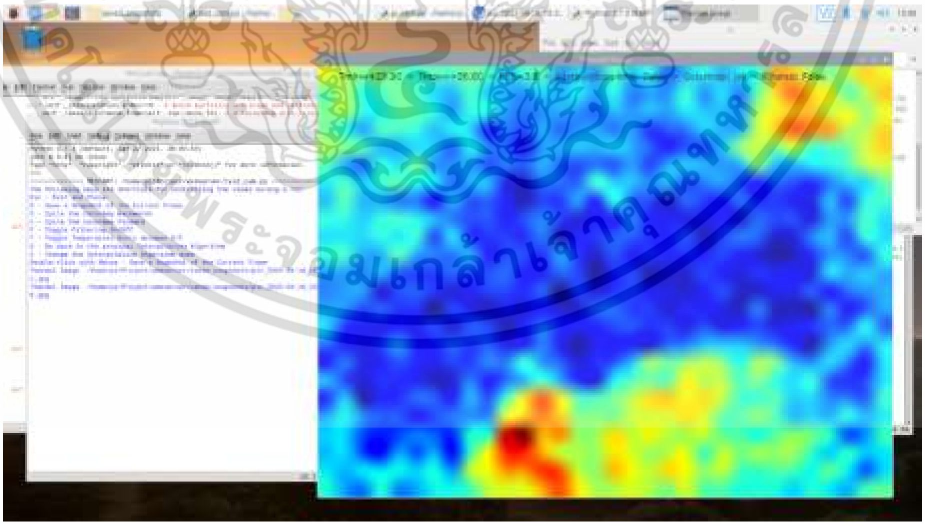

Figure 4.6 Result of MLX90640 camera.

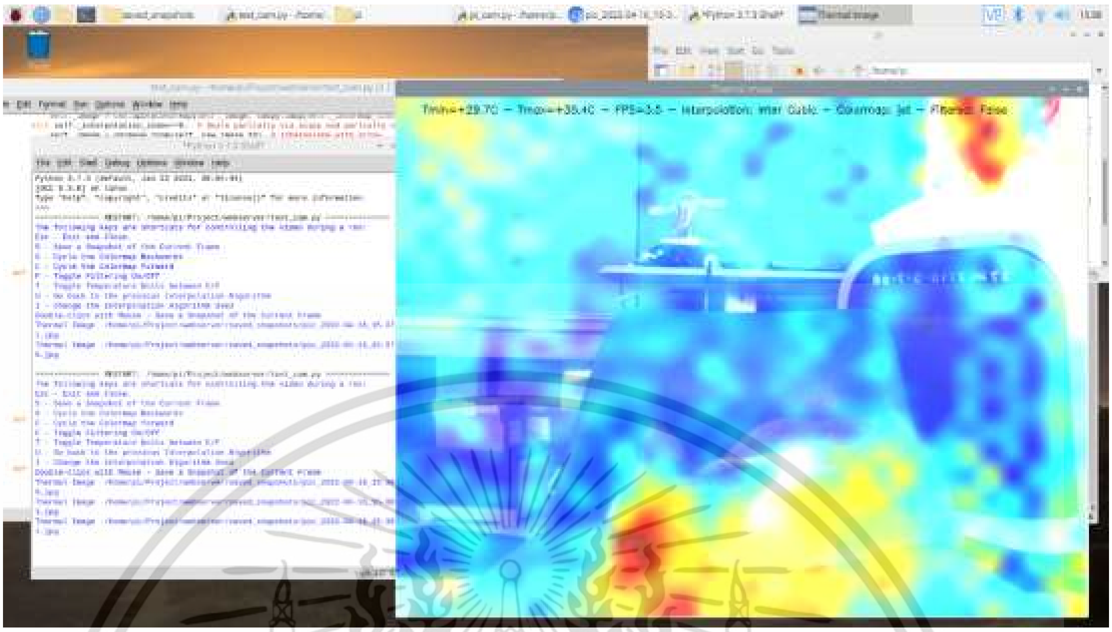

Figure 4.7 Result of Image fusion technique.

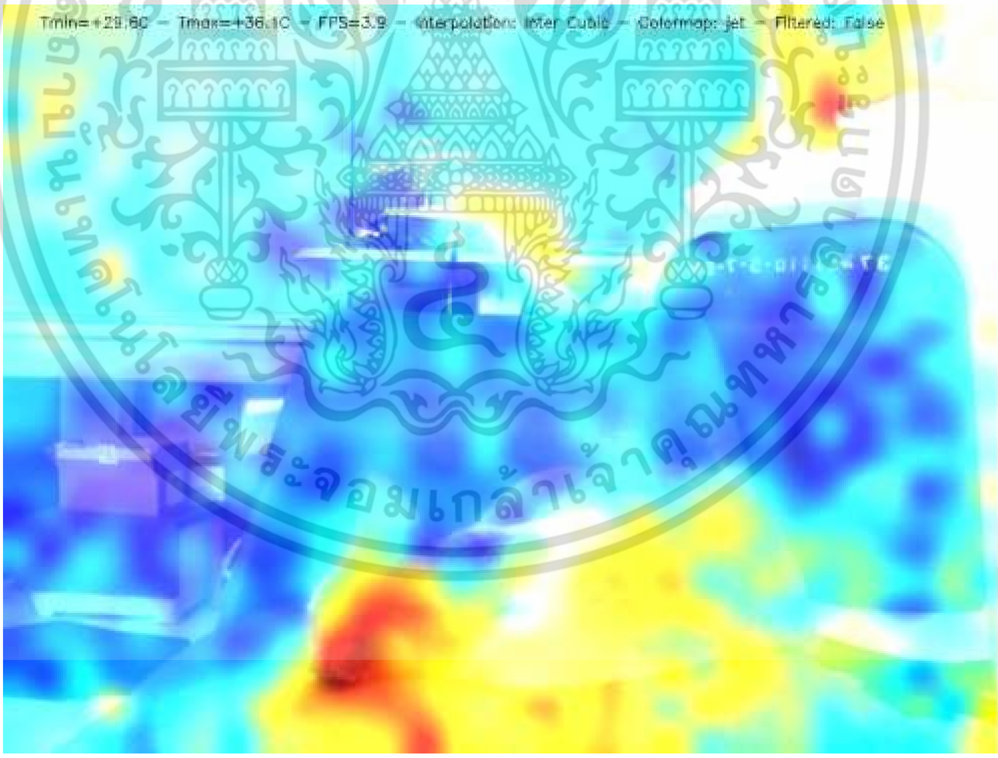

Figure 4.8 Fused image that captured by program.

iii. Background Subtraction

We then apply background subtraction to the image, which changes the minimum temperature. However, a problem arises with this function, as it significantly delays the video, which we will discuss in the next chapter.

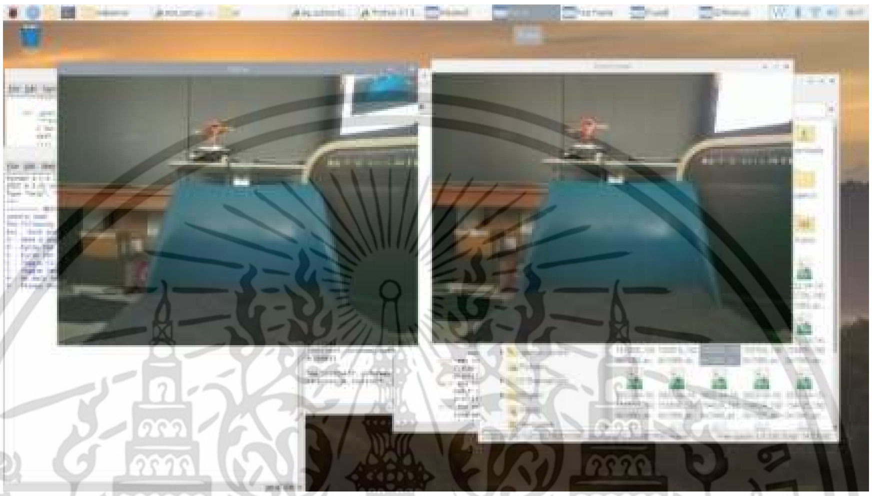

Figure 4.9 First frame the first image remembered as background, and Frame, the real-time video, image without any object.

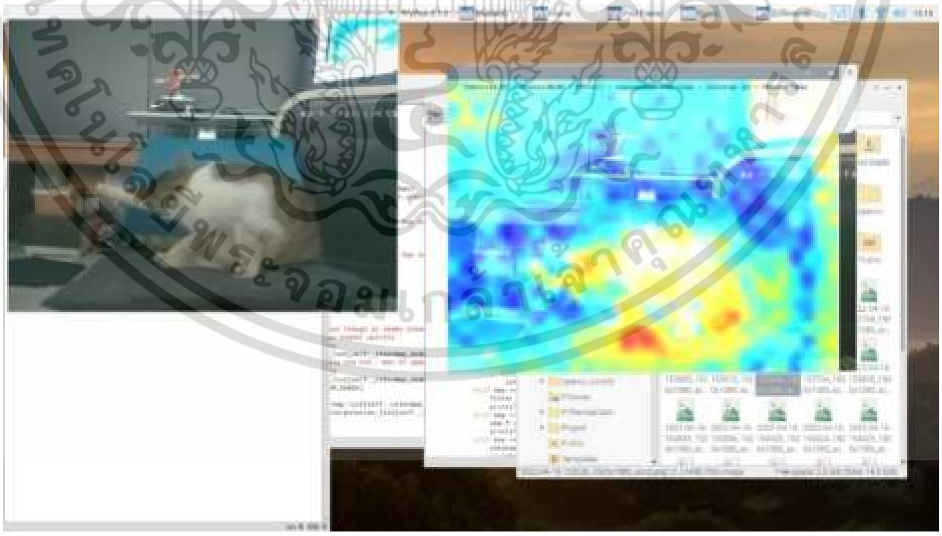

Figure 4.10 Fused image with object.

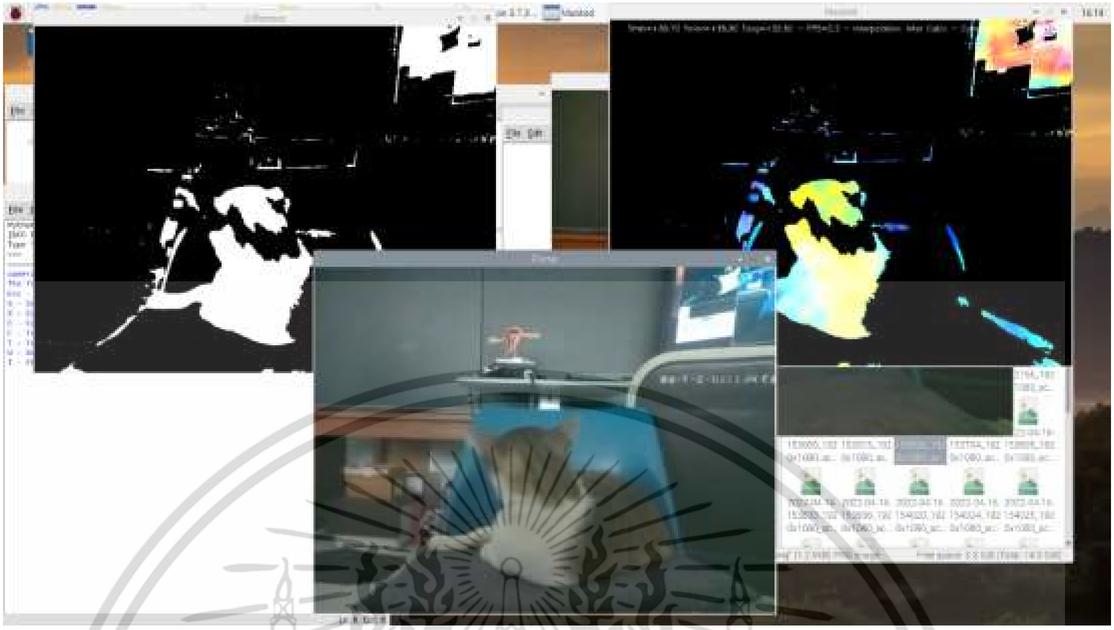

Figure 4.11 Frame with object (center), Difference (left), the binary image of background subtraction image, image with object and Masked image with object (right).

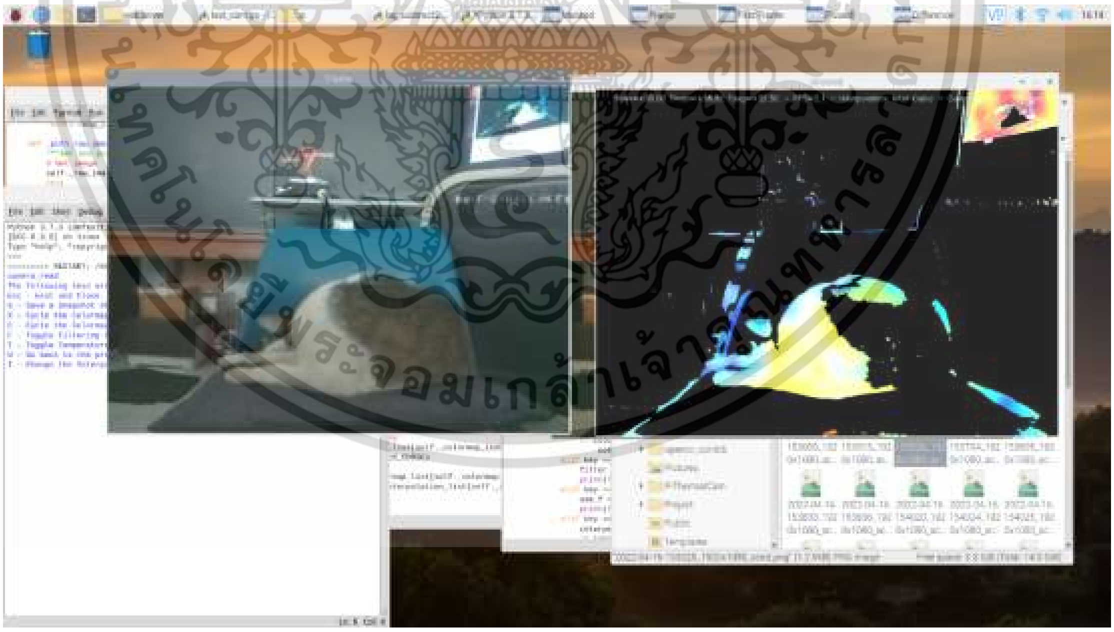

Figure 4.12 Frame with object (left), and Masked images with object (right).

4.2.4 Results of Result Display

Regarding the result display section, there are three separate sections: web server, database server, and alarming email.

i. Webserver

Initially, we only streamed one image on the web server, but we found that displaying original images would make it more engaging, so we modified the webpage and removed some redundant functions for better usability.

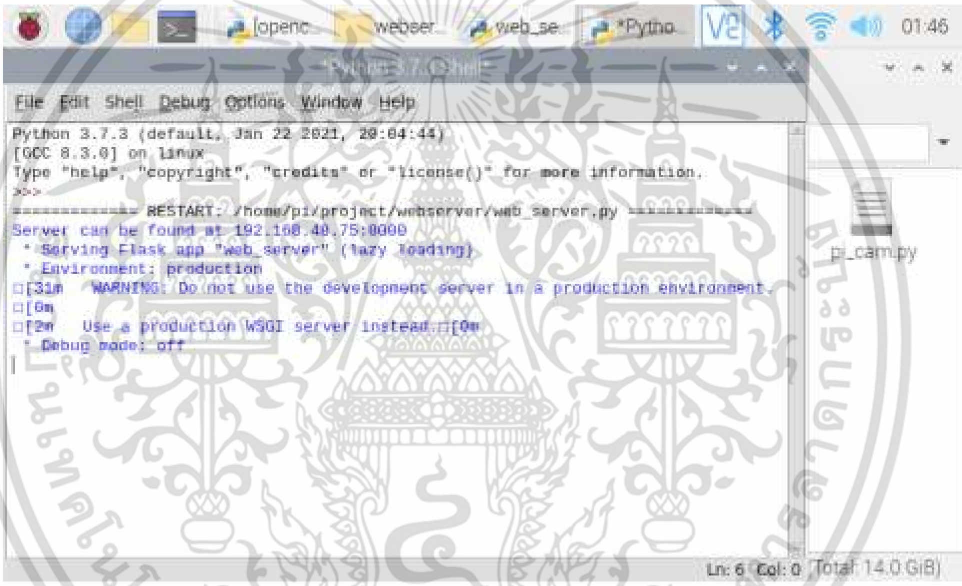

Figure 4.13 The address used to connect to Flask server, showed in the display of the image with web server code.

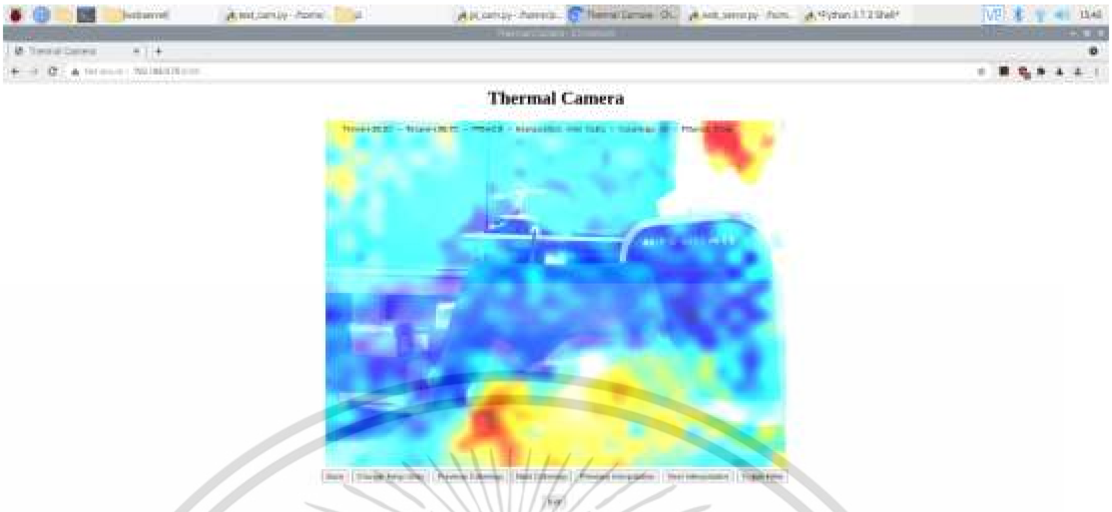

Figure 4.14 Flask server on web browser showed video streaming.

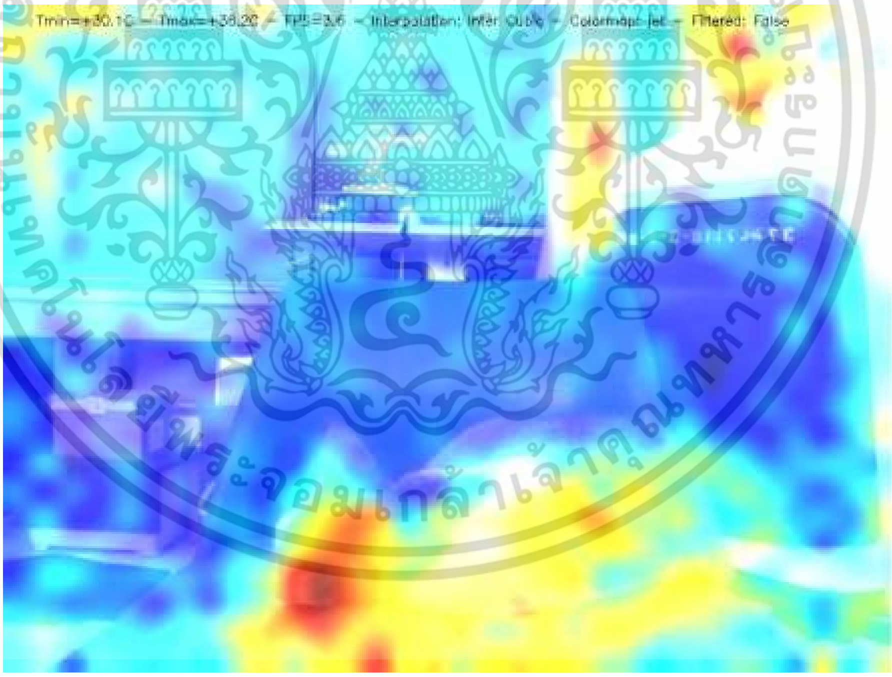

Figure 4.15 Saved image from Flask server.

ii. Database Server

Next, we utilized ThingSpeak's website to graphically display data and export it as needed. Multiple channels can also be established for numerous incubators.

Figure 4.16 Results of ThingSpeak code.

Figure 4.17 The data updated into ThingSpeak website.

iii. Alarming Email

Lastly, the system has been programmed to send a notification email when the temperature is too high. Once the temperature exceeds a certain threshold, the system can detect it and send an email notification within seconds.

Figure 4.18 The data sending to an assigned email.

4.3 Control system

After checking the operation of each component, including the heater, blower, PID controller, and thermostat, we proceed to test the control system by operating it and measuring the temperature inside the cabinet. Once the system can maintain the temperature as expected, we calibrate it using analyzers that are typically used with conventional devices.

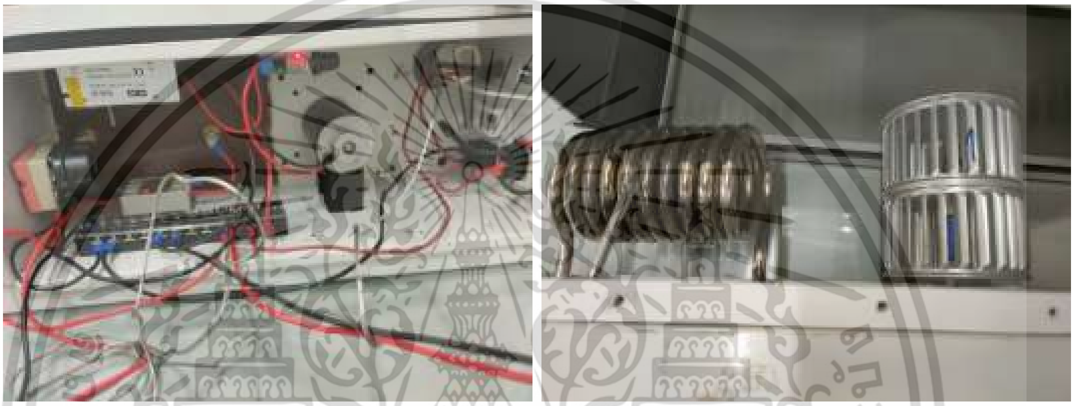

Figure 4.19 Circuit box of control system.

4.3.1 Operation of Control System

Since we switched the motor to a DC motor, our system can operate for an entire day without overheating and shutting down. However, the control box generates some heat from the heater, and it would be beneficial to add a fan inside the box to improve ventilation and allow the fan to rest periodically. In addition, our desired control system should incorporate a PID control algorithm to maintain temperature. PID control is a method used in control systems to maintain a desired output in the presence of disturbances or changes in the system. PID stands for Proportional-Integral-Derivative, which are the three components that make up the control algorithm.

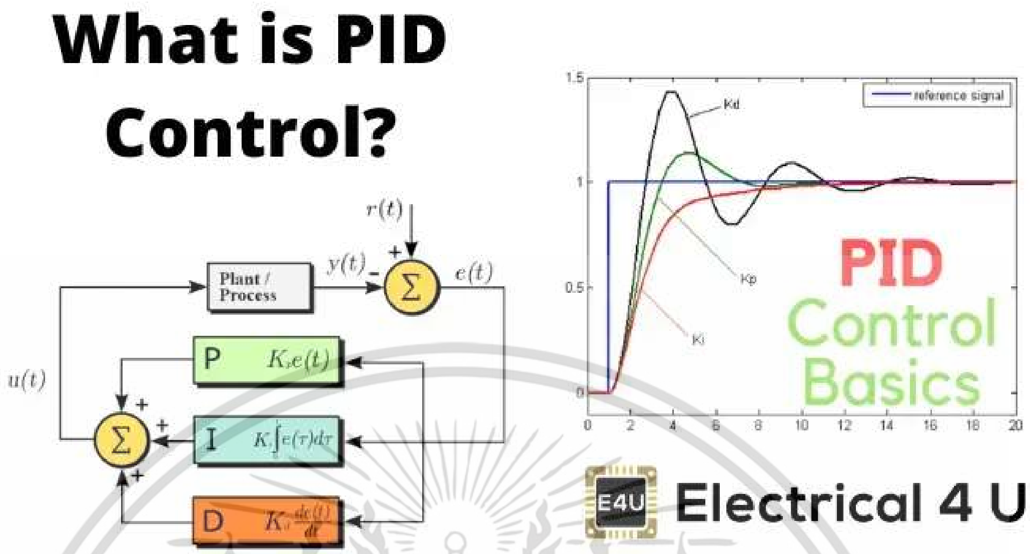

Figure 4.20 PID Control [22].

The proportional term provides a control signal that is proportional to the difference between the desired set point and the actual output. The integral term sums up the past errors over time and generates a corrective signal to eliminate any steady-state error. The derivative term predicts future changes in the error signal and generates a control signal to counteract them.

By combining these three terms, a PID controller can adjust the system's input to maintain the desired output in real-time. PID control is widely used in industrial control applications, such as temperature control, speed control, and position control of mechanical systems.

In order to gather the data presented in Figure 4.21, we utilized an Arduino-based temperature sensor. Subsequently, we plotted a graph based on the data retrieved from the serial port of the Arduino, which is depicted in Figure 4.23-4.26. Our findings indicated that the temperature steadily increased and remained stable around the designated set point due to the implementation of a PID controller.

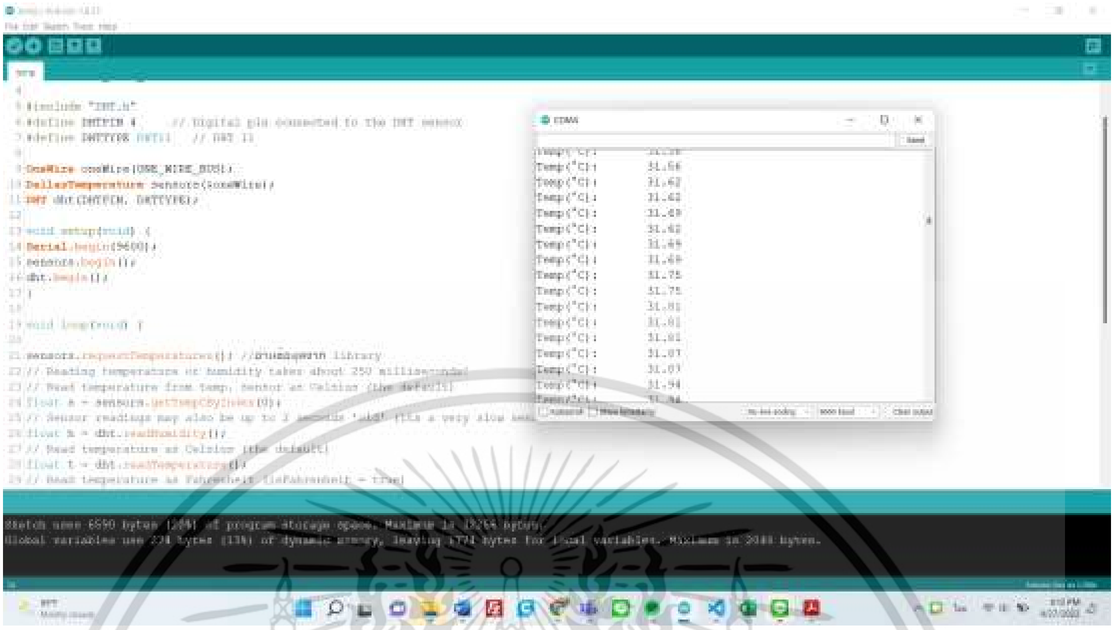

Figure 4.21 Recorded temperatures show in serial port of Arduino.

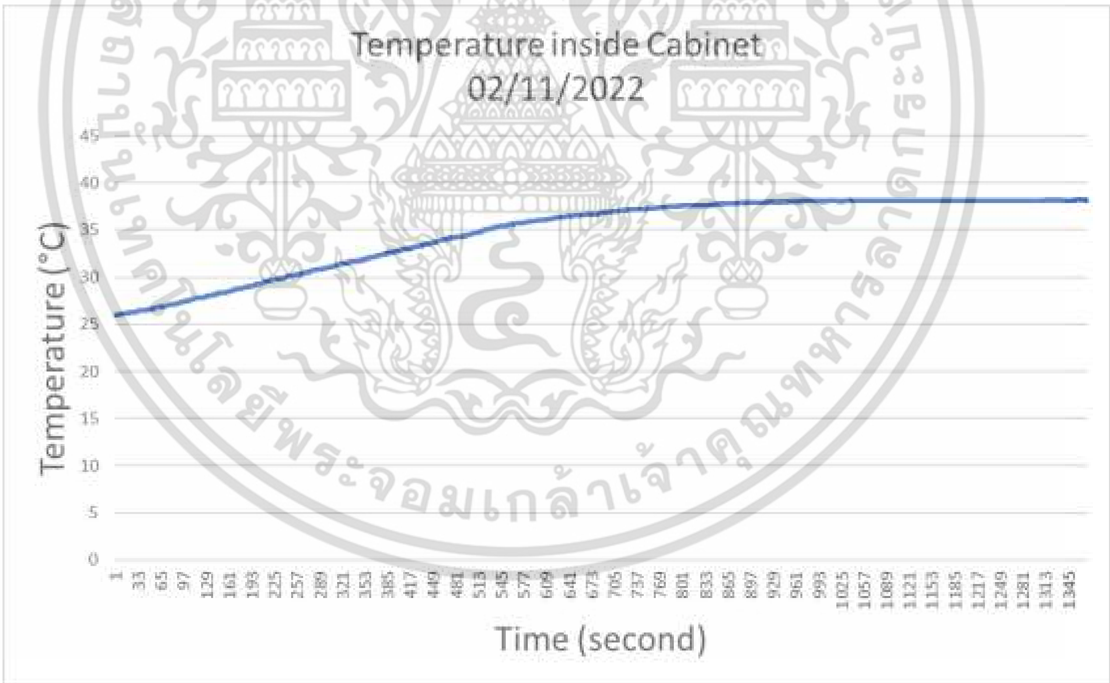

Figure 4.22 Recorded temperatures plot in graph form of YP-90A.

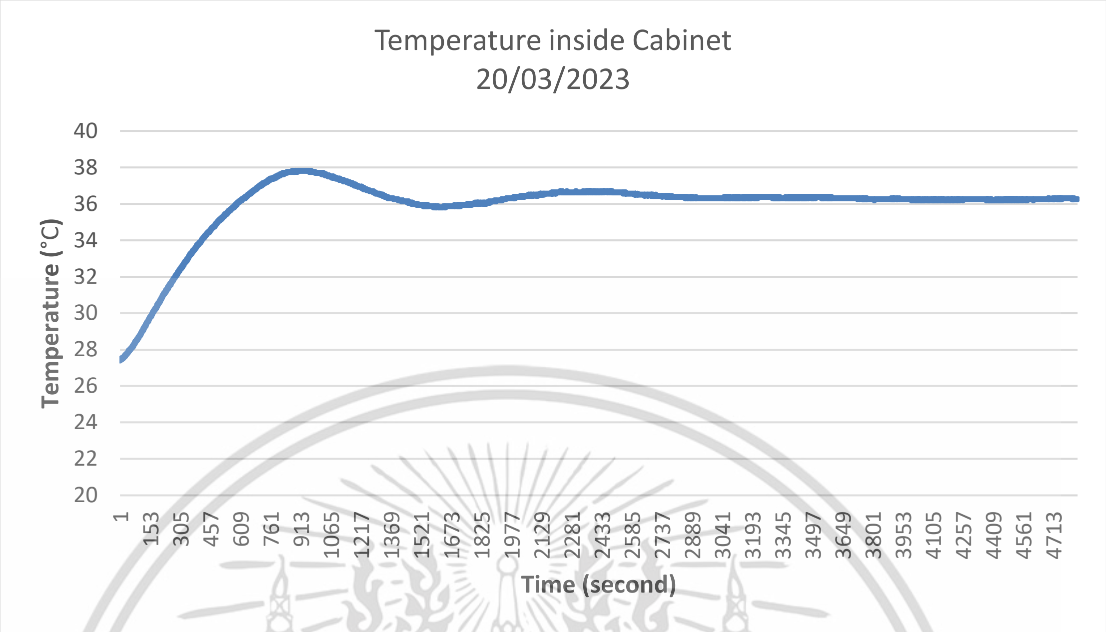

Figure 4.23 Recorded temperatures plot in graph form.

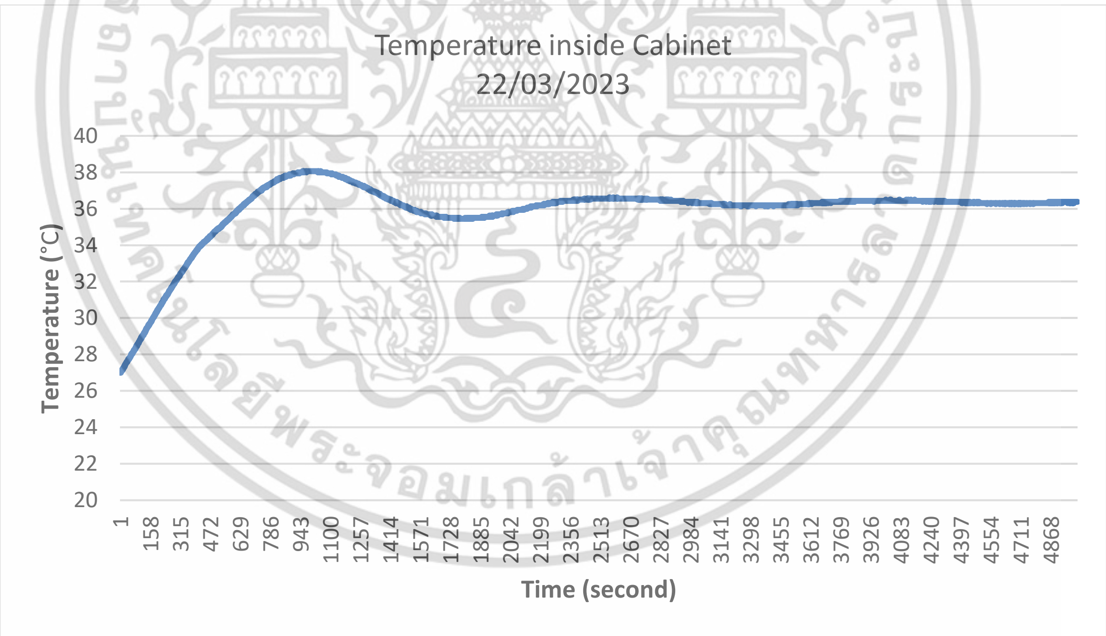

Figure 4.24 Recorded temperatures plot in graph form.

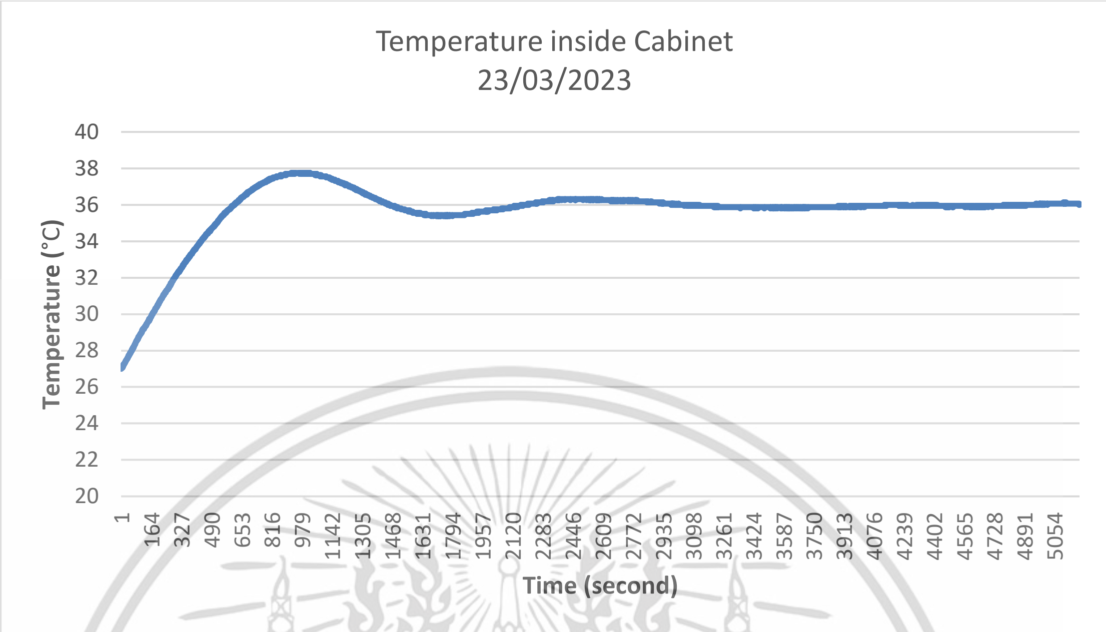

Figure 4.25 Recorded temperatures plot in graph form.

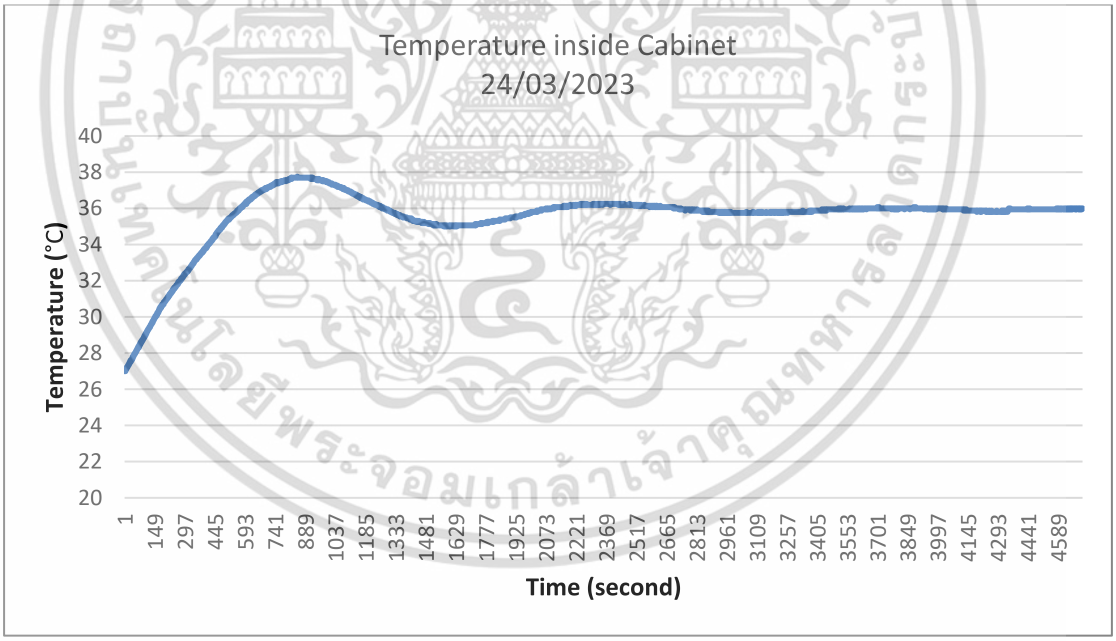

Figure 4.26 Recorded temperatures plot in graph form.

4.3.3 Testing Control System

Furthermore, we conducted testing on the control system using the INCU II analyzer. The Fluke Biomedical INCU II Neonatal Warmer is a specialized testing device designed for evaluating the performance and safety of neonatal warmers, incubators, and transport incubators. This comprehensive analyzer ensures compliance with global standards, such as IEC 60601-2-19, IEC 60601-2-20, and IEC 60601-2-21, while simplifying the testing process [23].

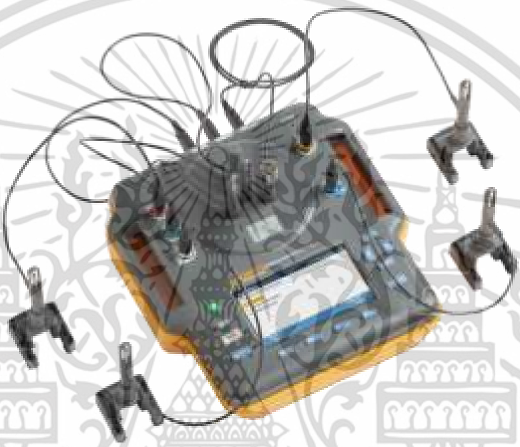

Figure 4.27 The Fluke Biomedical INCU II Neonatal Warmer [23].

We selected various testing modes to assess the performance of our system, including warm-up time, internal sound levels, temperature uniformity, stability of incubator temperature, temperature control accuracy, and temperature overshoot.

- Warm-up Time

  In order to meet the required criteria specified in the DUT manual, the temperature should rise by 11 °C within the time frame specified, with an allowable tolerance of ±20%. Specifically, we utilized the YP-90a Model to establish the appropriate timeframe for temperature rise, which amounted to 30 minutes ±20%. Our system demonstrated a temperature increase of 11 degrees within a time frame of 25.5 minutes, thereby satisfying the established criteria and passing the test.

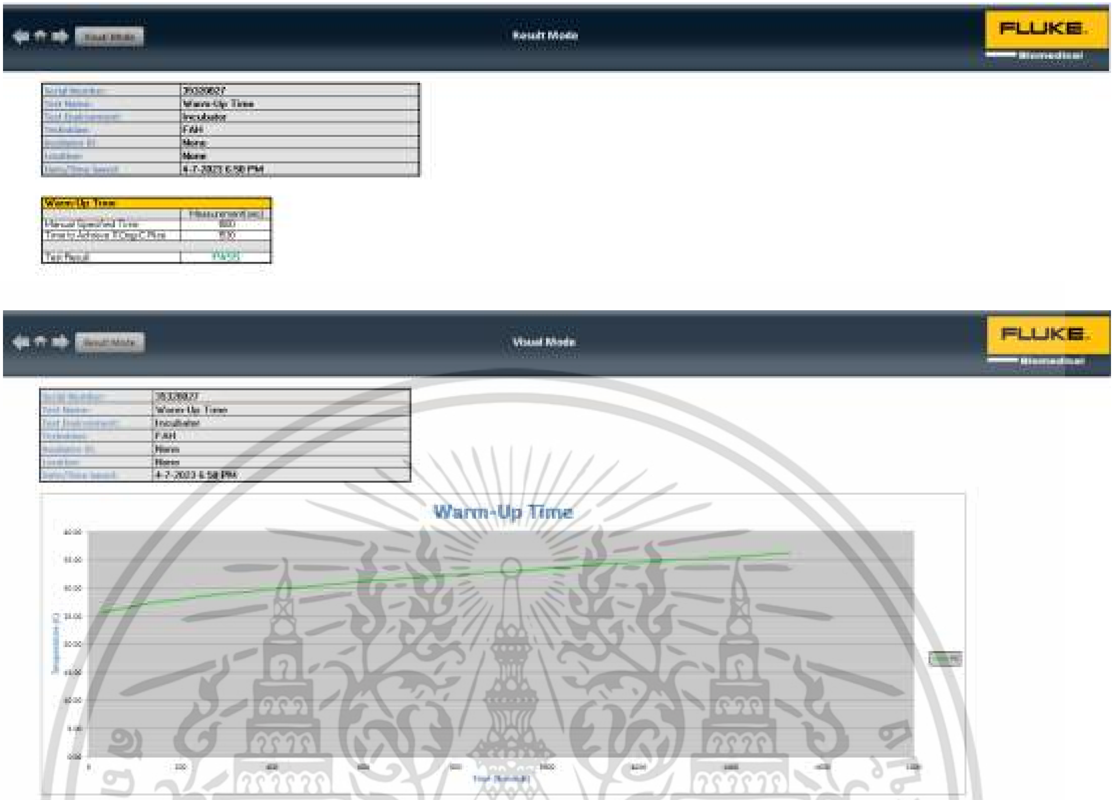

Figure 4.28 Warm-up Time Testing Result.

- Inside-Sound level

To meet the requirements for passing, the noise level within the compartment must not exceed 60 dBA, and the background noise should also not surpass the measured sound by more than 10 dBA. Our system exhibited a noise level of 59.9 dBA, successfully fulfilling the established criteria and passing the test.

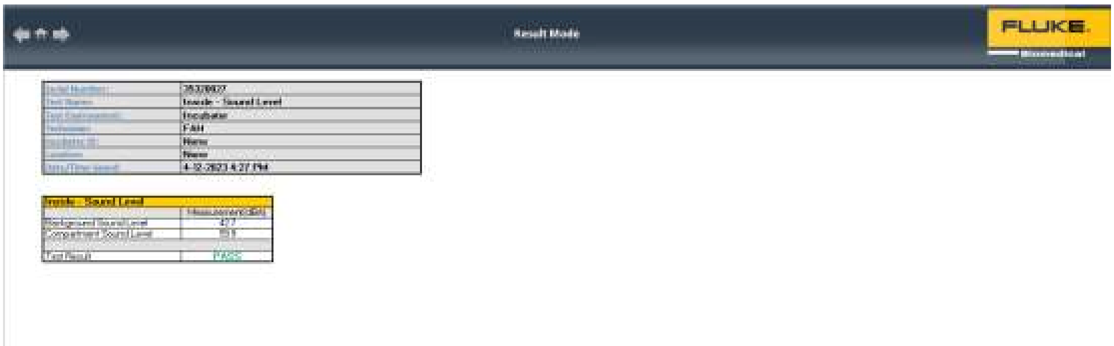

Figure 4.29 Inside-Sound level Testing Result.

- Uniformity of Temperature (For tests at 36 °C)

In order to meet the established standards for passing, the average temperature in each quadrant should be within 0.8 °C of the midpoint or within 1.0 °C if the mattress is tilted. Our system successfully satisfied these conditions and passed the test.

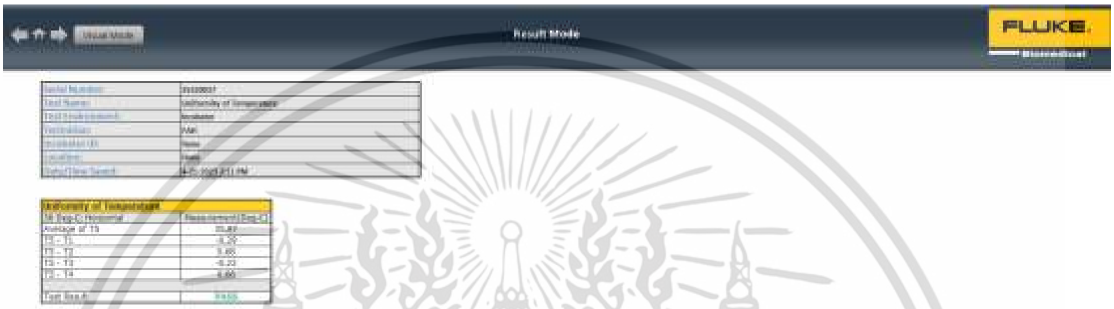

Figure 4.30 Uniformity of Temperature (For tests at 36 °C) Testing Result.

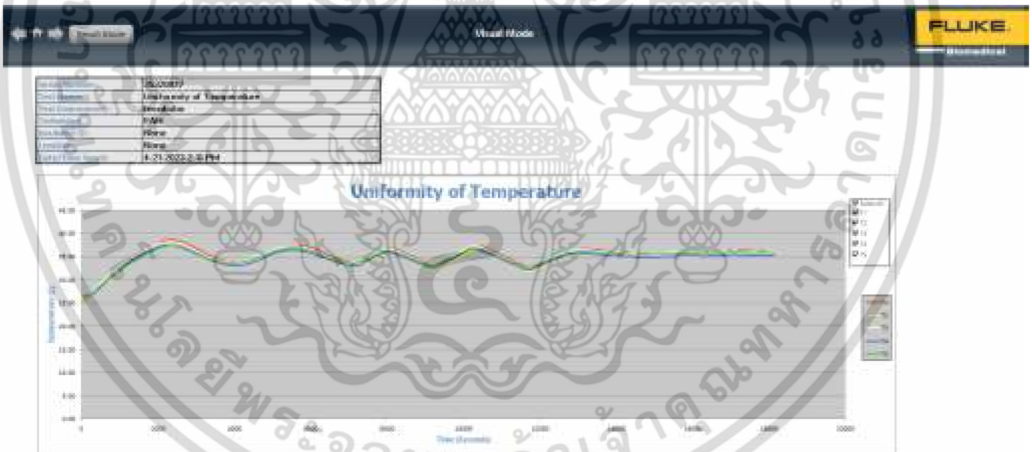

Figure 4.31 Uniformity of Temperature (For tests at 36 °C) Testing Result.

- Stability of Incubator Temperature (For tests at 36 °C)

In order to meet the requirements for passing, the average temperature must fall within ±0.5 °C of any temperature reading, with a specified setting temperature of 36 °C. Our system exhibited an average temperature of 36.13°C, with the largest deviation amounting to +0.12°C, successfully satisfying the established criteria and passing the test.

Figure 4.32 Stability of Incubator Temperature (For tests at 36 °C) Testing Result.

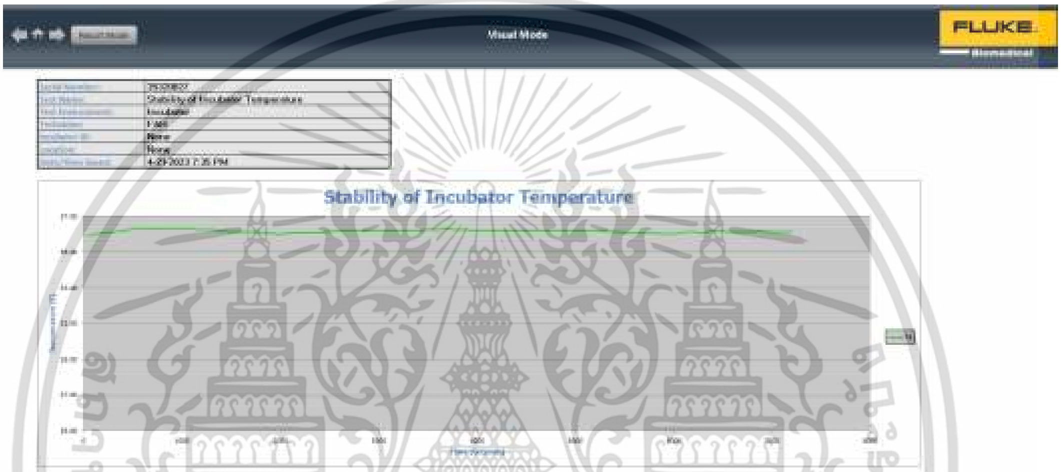

Figure 4.33 Stability of Incubator Temperature (For tests at 36 °C) Testing Result.

- Accuracy of Indicator

To meet the requirements for passing, the average temperature should not deviate from the temperature indication by more than ±0.8 °C. In our system, we set the temperature at 36 °C, and the average temperature reading was 36.08°C, which satisfied the established criteria and allowed our system to pass the test.

Figure 4.34 Accuracy of Indicator Testing Result.

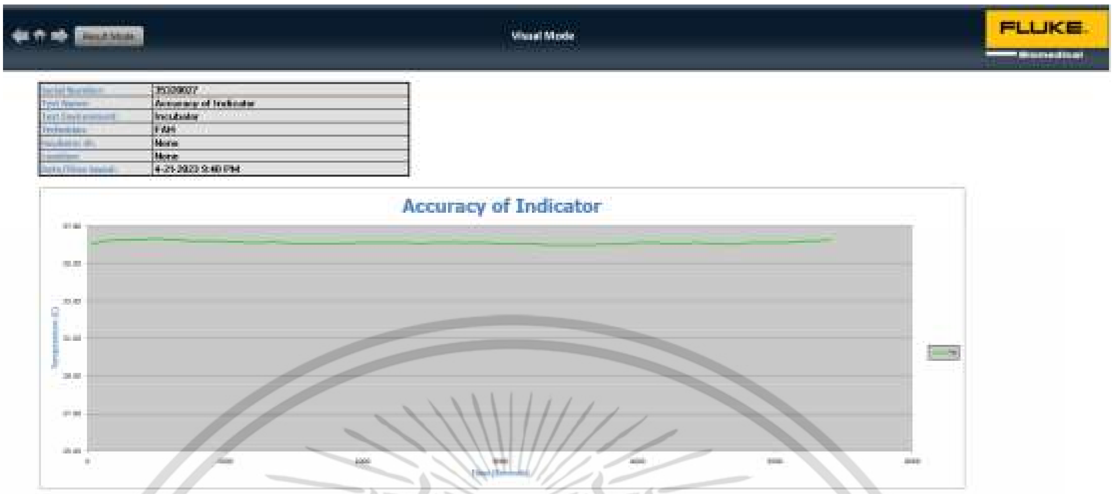

Figure 4.35 Accuracy of Indicator Testing Result.

- Overshoot for Incubator Temperature

  To pass the test, the incubator must not exceed 38 °C when the temperature is increased from 32 °C to 36 °C. Additionally, within 15 minutes, the temperature must reach 36 °C and remain stable enough to start the STC measurement. In our system, the highest temperature reached was 36.51 °C, which did not exceed 38 °C. However, the time it took to reach the STC at 36 °C was 27 minutes, which exceeded the 15-minute time limit, leading to a failed test.

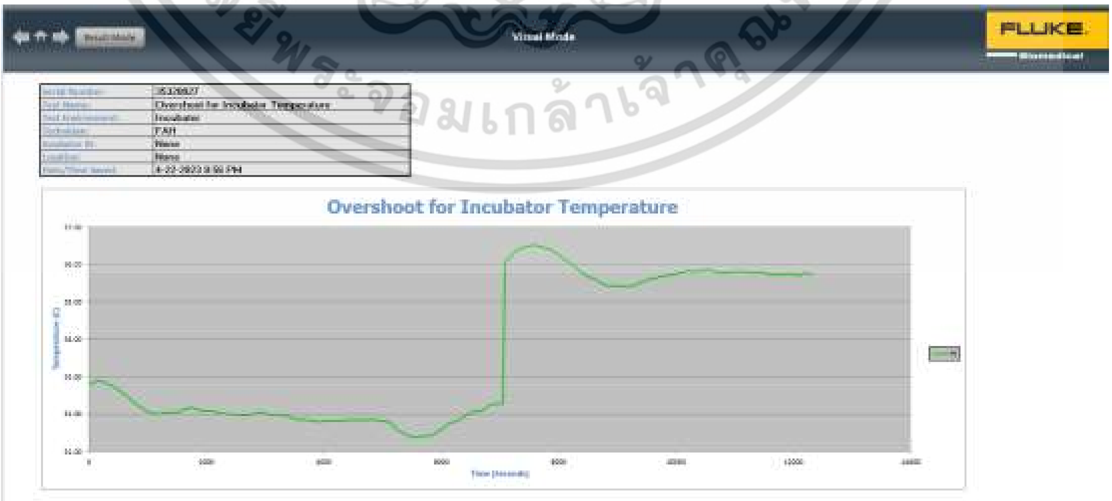

Figure 4.36 Overshoot for Incubator Temperature Testing Result.

Figure 4.37 Overshoot for Incubator Temperature Testing Result.

4.3.4 Comparing with YP-90A's specification

Apart from employing the YP-90A model as a reference template for the design of our incubator, we have further selected its operational attributes to undertake a comparative analysis with our own work, with the overarching objective of evaluating the efficacy of our project. The specific properties of the model that we have utilized for this comparative assessment encompass as following table:

Table 4.1 compare result with specifications of YP-90A.

Specifications	YP-90A	Our Incubator
Power Requirements	AC220V-230V/50Hz	AC220V/50Hz
Heater Power Output	377W/240V	400W/220V
Temperature rise Time*(environment temperature is +22 ℃)	≤30min	25.5min
Temperature Variation*	≤0.5℃	0.12℃
Temperature Uniformity (mattress is on the horizontal position)	≤0.8℃	≤0.66℃
Correlation of displayed incubator temperature to actual incubator temperature	≤0.8℃	0.08℃
Noise level within Hood Environment	< 50dB(A)	59.9dB

Through this methodical examination, we endeavor to gain valuable insights into the strengths and weaknesses of our own approach, thereby contributing to the scholarly discourse in this field and enhancing the overall quality of our research endeavor.

4.4 Budget

The cost of incubators typically used in neonatal intensive care units (NICUs) varies between \$1,500 and \$35,000 (approximately 50,000฿ to 120,000฿). Our incubator costs around 43,400฿, with the main function priced at 31,400฿, which operates similarly to a typical commercial incubator. In addition, there is an advanced monitoring system priced at 12,000฿, which is not included in the typical commercial incubator.

Table 4.2 shows budget for this research.

Items	Cost (฿)
Cabinet	25000
Control box	
-motor	600
-blower	300
-heater	2500
-PID controller	2000
Monitoring System	
-Raspberry Pi	3800
-Pi camera	1200
-Thermal camera MLX90640	7000
-wiring, switch, etc.	1000
Total	43,400

CHAPTER 5
CONCLUSION

5.1 Introduction

In this chapter, we discuss and summarize the discussion and conclusion about the work completed in each section from chapter 4 including designing and manufacturing incubator, control system, and finally monitoring system in order to provide a better understanding of the development of smart infant incubators.

5.2 Discussion

The section discusses various issues related to the design, development, and implementation of a system that includes cabinets, monitors, and control systems.

Firstly, the cabinets are designed based on existing models in the laboratory, but the cost of making transparent cabinets is quite high due to the required bending technique of a single sheet of acrylic. The discussion suggests the need for more research into the cabinet's shape to come up with a design that can help reduce the cost of building the cabinet while also reducing environmental noise.

Secondly, this part highlights a problem with the monitoring system, where combining code from various parts into one file results in a video with a lot of delay. We suggest consulting a professional programmer to solve the program and improve the output's display smoothness. Additionally, the proposed of sending reminders via email to help doctors or nurses who are involved in taking care of the babies, is simply an idea, and it is uncertain whether it will be useful or not. We recommend consulting with them to identify other potential benefits that can be added to the system.

Lastly, we discuss the limitations of the current control system, which can only control temperature. We propose switching to using a microcontroller to write a program to control itself and incorporate with other sensors, such as sound, humidity, and oxygen sensors. Additionally, it suggests modifying the system so that both monitoring and control systems can be controlled from the same place, making it easier for users to use.

Overall, we highlight the need for further research, consultation with professionals and end-users, and the use of advanced technologies to improve the design, development, and implementation of the system.

5.3 Conclusion

This project has achieved the primary design objectives: (1) improve the monitoring system of the incubator using IoT, and (2) the system can function at the same level as a commercially available infant incubator. By combining images and performing background subtraction, the monitoring system's efficiency may be enhanced, and the thermal image can be observed more effectively. The measured temperature is accurate and reliable. The display result part that streams the results online including real-time video on Flask server, real-time database on ThingSpeak and alarm data sending to email, are able to work perfectly. The incubator has been designed and manufactured with inexpensive cost. The control system of the incubator is able to work similarly to the simple incubators.

However, more studies still needed to be done to improve it further. If the smart infant incubator proves to be a success, it may make working more convenient for the doctor or the other people involved through its development. With more studies into developing infant incubators, it will be possible to turn it into a commercially viable product in the future. It would be beneficial and cost reduction a lot if we could manufacture incubators in Thailand.

REFERENCES

[1] P. T. Kapen, Y. Mohamadou, F. Momo, D. K. Jauspin, N. Kanmagne and D. D. Jordan, "Development of a neonatal incubator with phototherapy, biometric fingerprint reader, remote monitoring, and heart rate control adapted for developing countries hospitals," Journal of Neonatal Nursing, vol. 25, no. 6, pp. 298-303, December 2019.

[2] A. Rajalakshmi, K. A. Sunitha and R. Venkataraman, "A Survey on Neonatal Incubator Monitoring System," *Journal of Physics: Conference Series,* vol.1362, p. 012128, November 2019.

[3] "**Industry Outlook 2021-23: Medical Devices**" [Online]. Available: https://www.krungsri.com/en/research/industry/industry-outlook/other-industries/ medical-devices/io/medical-devices-21. [Accessed April 2023].

[4] W. Shalannanda, I. Zakia, E. Sutanto and F. Fahmi, "Design of Hardware Module of IoT-based Infant Incubator Monitoring System," 2020 6th International Conference on Wireless and Telematics (ICWT), 2020, pp. 1-6, doi: 10.1109/ICWT50448.2020.9243665.

[5] "**Premature Birth**" [Online]. Available: https://www.mayoclinic.org/diseases-conditions/premature-birth/symptoms-causes/syc-20376730 [Accessed April 2023].

[6] "**Infant incubator**" [Online]. Available: http://www.frankshospitalwork-shop.com/equipment/infant_incubators_equipment.html. [Accessed October 2021].

[7] "**Infant incubator**" [Online]. Available: https://www.medicalexpo.com/prod/ suzhou-being-medical-device/product-130153-969547.html. [Accessed October 2021].

[8] " **What is the internet of things (IoT)?**" [Online]. https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT. [Accessed April 2023].

[9] " **Image Processing: Techniques, Types, & Applications [2023]**" [Online]. https://www.v7labs.com/blog/image-processing-guide. [Accessed April 2023].

[10] " **YP-90A incubator** " [Online]. Available: https://www.medicalexpo.com/prod/ningbo-david-medical-device/product-69526-432601.html [Accessed April 2022].

[11] " **Image Processing in Python: Algorithms, Tools, and Methods You Should Know** " [Online]. https://neptune.ai/blog/image-processing-python. [Accessed April 2023].

[12] " **Image Processing Using OpenCV – With Practical Examples** " [Online]. https://www.analyticsvidhya.com/blog/2021/05/image-processing-using-opencv-with-practical-examples/. [Accessed April 2023].

[13] Kaur, H., Koundal, D. & Kadyan, V. Image Fusion Techniques: A Survey. Archives of Computational Methods in Engineering, vol.28, pp.4425–4447, January 2021.

[14] " **Using Background Subtraction Methods in Image Processing** " [Online]. https://analyticsindiamag.com/using-background-subtraction-methods-in-image-processing/. [Accessed April 2023].

[15] " **How to build a web application using Flask and deploy it to the cloud** " [Online]. https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/. [Accessed April 2023].

[16] " **How To Make a Web Application Using Flask in Python 3** " [Online]. https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3. [Accessed April 2023].

[17] " **PiThermalCam** " [Online]. Available: https://tomshaffner.github.io/PiThermalCam/ [Accessed November 2021].

[18] " **ThingSpeak** " [Online]. https://en.wikipedia.org/wiki/ThingSpeak. [Accessed April 2023].

[19] " **ThingSpeak** " [Online]. https://www.mathworks.com/products/thingspeak.html. [Accessed April 2023].

[20] " **Sending An Email Using Python On The Raspberry Pi Sending An Email Using Python On The Raspberry Pi** " [Online]. https://bc-

robotics.com/tutorials/sending-email-using-python-raspberry-pi/. [Accessed April 2023].

[21] " **CG AI-518P 32-Segment Programmable Temperature Controller Intelligent Digital Process RS485 MODBUS Protocol Pid Thermostat** " [Online]. https://www.alibaba.com/product-detail/CG-AI-518P-32-Segment-Programmable_62432625972.html. [Accessed April 2023].

[22] " **PID Controllers and PID Control in Control Systems** " [Online]. https://www.electrical4u.com/pid-control/. [Accessed April 2023].

[23] " **INCU II Incubator Radiant Warmer Analyzer** " [Online]. https://www.flukebiomedical.com/products/biomedical-test-equipment/incubator-radiant-warmer-analyzers/incu-ii-incubator-radiant-warmer-analyzer. [Accessed April 2023].

APPENDICES

APPENDIX A

PYTHON CODE FOR IMAGE PROCESSING

```
# import the necessary packages
from imutils.video import VideoStream
from mail import Emailer

import argparse
import imutils

import time,board,busio
import numpy as np
import adafruit_mlx90640
import datetime as dt
import cv2
import logging, configparser
import cmapy
import traceback
from scipy import ndimage

import urllib.request as urllib2
import smtplib

#Email Variables
SMTP_SERVER = 'smtp.gmail.com' #Email Server (don't change!)
SMTP_PORT = 587 #Server Port (don't change!)
GMAIL_USERNAME = 'xxxx@kmitl.ac.th' #change this to match your gmail
account
GMAIL_PASSWORD = 'xxxxx' #change this to match your gmail password


class Emailer:
    def sendmail(self, recipient, subject, content):

        #Create Headers
        headers = ["From: " + GMAIL_USERNAME, "Subject: " + subject,
"To: " + recipient,
        "MIME-Version: 1.0", "Content-Type: text/html"]
        headers = "\r\n".join(headers)

        #Connect to Gmail Server
        session = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
        session.ehlo()
```

```
        session.starttls()
        session.ehlo()

        #Login to Gmail
        session.login(GMAIL_USERNAME, GMAIL_PASSWORD)

        #Send Email & Exit
        session.sendmail(GMAIL_USERNAME, recipient, headers +
"\r\n\r\n" + content)
        session.quit

MASK_COLOR = (0.0,0.0,0.0) # In BGR format

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--picamera", type=int, default=-1,
    help="whether or not the Raspberry Pi camera should be used")
args = vars(ap.parse_args())

# Enter Your API key here
myAPI = 'BWRIG48A2R7RLEH5'
# URL where we will send the data, Don't change it
baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI


i2c = busio.I2C(board.SCL, board.SDA, frequency=800000)  # setup I2C
mlx = adafruit_mlx90640.MLX90640(i2c)  # begin MLX90640 with I2C comm
mlx.refresh_rate = adafruit_mlx90640.RefreshRate.REFRESH_8_HZ  # set
refresh rate
time.sleep(0.1)


def c_to_f(temp:float):
    """ Convert temperature from C to F """
    return ((9.0/5.0)*temp+32.0)


# function to convert temperatures to pixels on image
def temps_to_rescaled_uints(f,Tmin,Tmax):
    norm = np.uint8((f - Tmin)*255/(Tmax-Tmin))
    norm.shape = (24,32)
    return norm


def take_pic(use_f=True):
    """Take single pic"""
    image = np.zeros((24*32,))
```

```
    # Get image
    mlx.getFrame(image)  # read mlx90640
    temp_min = np.min(image)
    temp_max = np.max(image)
    image=temps_to_rescaled_uints(image,temp_min,temp_max)

    # Image processing
    img = cv2.applyColorMap(image, cmapy.cmap('bwr'))
    img = cv2.resize(img, (320,240), interpolation=cv2.INTER_CUBIC)
    img = cv2.flip(img, 1)

    if use_f:
        temp_min=c_to_f(temp_min)
        temp_max=c_to_f(temp_max)
        text = f'Tmin={temp_min:+.1f}F Tmax={temp_max:+.1f}F'
    else:
        text = f'Tmin={temp_min:+.1f}C Tmax={temp_max:+.1f}C'

    text = 'Tmin = {:+.1f} Tmax = {:+.1f} '.format(temp_min/100,
temp_max/100)
    cv2.putText(img, text, (5, 15), cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0,
0, 0), 1)
    cv2.imshow('Output', img)

    fname = 'pic_' + dt.datetime.now().strftime('%Y-%m-%d_%H-%M-%S') +
'.jpg'
    cv2.imwrite(fname, img)
    print('Saving image ', fname)


def save_snapshot(event,x,y,flags,param):
    """Used to save an image on double-click"""
    img=param[0]
    global mouseX,mouseY
    if event == cv2.EVENT_LBUTTONDBLCLK:
        fname = 'pic_' + dt.datetime.now().strftime('%Y-%m-%d_%H-%M-
%S') + '.jpg'
        cv2.imwrite(fname, img)
        print('Thermal Image ', fname)


def camera_read(use_f:bool = False, filter_image:bool = False):
    camera = VideoStream(usePiCamera=args["picamera"] > 0).start()
    camera.resolution = (640, 480)
    camera.framerate = 32
```

```
    first_frame = camera.read()
    first_frame = cv2.flip(first_frame, 0)
    # first_frame = cv2.rotate(first_frame, cv2.ROTATE_90_CLOCKWISE)
    first_frame = cv2.resize(first_frame, (24*32,24*24))
    first_gray = cv2.cvtColor(first_frame, cv2.COLOR_BGR2GRAY)
    first_gray = cv2.GaussianBlur(first_gray, (5, 5), 0)

    image = np.zeros((24*32,))
    t0 = time.time()
    colormap_index = 0
    interpolation_index = 3
    file_saved_notification_start = None
    # See https://gitlab.com/cvejarano-oss/cmapy/-
/blob/master/docs/colorize_all_examples.md to develop list
    colormap_list=['jet','bwr','seismic','coolwarm','PiYG_r','tab10','t
ab20','gnuplot2','brg']
    interpolation_list
=[cv2.INTER_NEAREST,cv2.INTER_LINEAR,cv2.INTER_AREA,cv2.INTER_CUBIC,cv2
.INTER_LANCZOS4,5,6]
    interpolation_list_name = ['Nearest','Inter Linear','Inter
Area','Inter Cubic','Inter Lanczos4','Scipy', 'Scipy/CV2 Mixed']

    print_shortcuts_keys()


    try:
        while True:
            # Get image
            frame = camera.read()
            frame = cv2.flip(frame, 0)
            # frame = cv2.rotate(frame, cv2.ROTATE_90_CLOCKWISE)
            try:
                mlx.getFrame(image)  # read mlx90640
            except RuntimeError as e:
                if e.message == 'Too many retries':
                    print("Too many retries error caught, potential I2C
baudrate issue: continuing...")
                    continue
                raise
            temp_min = np.min(image)
            temp_max = np.max(image)
            img=temps_to_rescaled_uints(image,temp_min,temp_max)


            # Image processing
```

```
            # Can't apply colormap before ndimage, so reversed in first
two options, even though it seems slower
            if interpolation_index==5:  # Scale via scipy only -
slowest but seems higher quality
                img = ndimage.zoom(img,25)  # interpolate with scipy
                img = cv2.applyColorMap(img,
cmapy.cmap(colormap_list[colormap_index]))
            elif interpolation_index==6:  # Scale partially via scipy
and partially via cv2 - mix of speed and quality
                img = ndimage.zoom(img,10)  # interpolate with scipy
                img = cv2.applyColorMap(img,
cmapy.cmap(colormap_list[colormap_index]))
                img = cv2.resize(img, (800,600),
interpolation=cv2.INTER_CUBIC)
            else:
                img = cv2.applyColorMap(img,
cmapy.cmap(colormap_list[colormap_index]))
                img = cv2.resize(img, (800,600),
interpolation=interpolation_list[interpolation_index])
            img = cv2.flip(img, 1)
            if filter_image:
                img=cv2.bilateralFilter(img,15,80,80)


            # For a brief period after saving, display saved
notification
            if file_saved_notification_start is not None and
(time.monotonic()-file_saved_notification_start)<1:
                cv2.putText(img, 'Snapshot Saved!',
(300,300),cv2.FONT_HERSHEY_SIMPLEX, .8, (0, 0, 0), 2)


            frame = cv2.resize(frame, (24*32,24*24))
            img = cv2.resize(img, (24*32,24*24))
            img = cv2.addWeighted(img,1,frame,1,0,img) #combine the
images


            gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            gray_frame = cv2.GaussianBlur(gray_frame, (5, 5), 0)
            difference = cv2.absdiff(first_gray, gray_frame)
            _, difference = cv2.threshold(difference, 25, 255,
cv2.THRESH_BINARY)
```

```
            mask_stack = np.dstack([difference]*3)    # Create 3-
channel alpha mask

            #-- Blend masked img into MASK_COLOR background -----------
---------------------------
            mask_stack  = mask_stack.astype('float32') /
255.0          # Use float matrices,
            img         = img.astype('float32') /
255.0                 #  for easy blending

            masked = (mask_stack * img) + ((1-mask_stack) * MASK_COLOR)
# Blend
            masked = (masked *
255).astype('uint8')                     # Convert back to 8-bit


            if use_f:
                temp_min=c_to_f(temp_min)
                temp_max=c_to_f(temp_max)
                text = f'Tmin={temp_min:+.1f}F - Tmax={temp_max:+.1f}F
- FPS={1/(time.time() - t0):.1f} - Interpolation:
{interpolation_list_name[interpolation_index]} - Colormap:
{colormap_list[colormap_index]} - Filtered: {filter_image}'
            else:
                text = f'Tmin={temp_min:+.1f}C - Tmax={temp_max:+.1f}C
- FPS={1/(time.time() - t0):.1f} - Interpolation:
{interpolation_list_name[interpolation_index]} - Colormap:
{colormap_list[colormap_index]} - Filtered: {filter_image}'
            cv2.putText(img, text, (30, 18), cv2.FONT_HERSHEY_SIMPLEX,
.4, (0, 0, 0), 1)
            img = cv2.resize(img, (800,600))

            img_masked = (difference[1]/255)*image


            temp1_min = np.min(img_masked)
            temp1_max = np.max(img_masked)
            temp1_avg = np.mean(img_masked)
            if temp1_max > 0:
                img_masked = np.extract(img_masked>0,img_masked)
                temp1_min = np.min(img_masked)
                temp1_avg = np.mean(img_masked)


            if use_f:
                temp_min=c_to_f(temp1_min)
```

```
                temp_max=c_to_f(temp1_max)
                temp_avg=c_to_f(temp1_avg)
                text = f'Tmin={temp1_min:+.1f}C Tmax={temp1_max:+.1f}F
Tavg={temp1_avg:+.1f}C - FPS={1/(time.time() - t0):.1f} -
Interpolation: {interpolation_list_name[interpolation_index]} -
Colormap: {colormap_list[colormap_index]} - Filtered: {filter_image}'
            else:
                text = f'Tmin={temp1_min:+.1f}C Tmax={temp1_max:+.1f}C
Tavg={temp1_avg:+.1f}C - FPS={1/(time.time() - t0):.1f} -
Interpolation: {interpolation_list_name[interpolation_index]} -
Colormap: {colormap_list[colormap_index]} - Filtered: {filter_image}'
            cv2.putText(masked, text, (30, 18),
cv2.FONT_HERSHEY_SIMPLEX, .4, (255, 255, 255), 1)
            masked = cv2.resize(masked, (800,600))

            first_frame = cv2.resize(first_frame, (800,600))
            frame = cv2.resize(frame, (800,600))
            difference = cv2.resize(difference, (800,600))

            # Display image on screen
            cv2.namedWindow('Thermal Image', cv2.WINDOW_NORMAL)
            cv2.resizeWindow('Thermal Image', 800,600)
            cv2.imshow('Thermal Image', img)
            cv2.imshow('fgmask', fgmask)
            cv2.imshow('frame',img )

            # Set mouse click events
            cv2.setMouseCallback('Thermal
Image',save_snapshot,param=[img])

            # if 's' is pressed - saving of picture
            key = cv2.waitKey(1) & 0xFF
            if key == ord("s"):  # If s is chosen, save an image to
filec
                fname = 'pic_' + dt.datetime.now().strftime('%Y-%m-
%d_%H-%M-%S') + '.jpg'
                cv2.imwrite(fname, img)
                file_saved_notification_start = time.monotonic()
                print('Thermal Image ', fname)
            elif key == ord("c"):  # If c is chosen cycle the colormap
used
                colormap_index+=1
                if colormap_index==len(colormap_list):
                    colormap_index=0
            elif key == ord("x"):  # If c is chosen cycle the colormap
used back
```

```
                colormap_index-=1
                if colormap_index<0:
                    colormap_index=len(colormap_list)-1
            elif key == ord("f"):  # If f is chosen cycle the image
filtering
                filter_image = not filter_image
                print(f"Filter On: {filter_image}")
            elif key == ord("t"):  # If t is chosen cycle the units
used for Temperature
                use_f = not use_f
                print(f"Using Fahrenheit: {use_f}")
            elif key == ord("u"):  # If u is chosen cycle interpolation
algorithm back
                interpolation_index-=1
                if interpolation_index<0:
                    interpolation_index=len(interpolation_list)-1
            elif key == ord("i"):  # If i is chosen cycle interpolation
algorithm
                interpolation_index+=1
                if interpolation_index==len(interpolation_list):
                    interpolation_index=0
            elif key==27:  # Break if escape key is used
                raise KeyboardInterrupt

            conn = urllib2.urlopen(baseURL +
'&field1=%f&field2=%f&field3=%f&field4=%f&field5=%f' % (temp_min,
temp_max, temp1_min, temp1_max, temp1_avg))
            print (f'Tempareture Min:{temp_min:.1f} Max:{temp_max:.1f}
Min:{temp1_min:.1f} Max:{temp1_max:.1f} Avg:{temp1_avg:.1f}' )
            conn.close()
            # time.sleep(1)



            if temp_max>38 :
                emailSubject = "SMART INFANT INCUBATOR" #The Subject of
the Email
                sender = Emailer()
                sendTo = 'faahthana@gmail.com' #Change this to the
email you wish to send to
                emailContent = "Now temperature is %.1fC !!"
%(temp1_max)

                sender.sendmail(sendTo, emailSubject, emailContent)
```

```
        camera.stop()

    except KeyboardInterrupt:
        cv2.destroyAllWindows()
        print("Code Stopped by User")

    cv2.destroyAllWindows()


def print_shortcuts_keys():
    """Print out a summary of the shortcut keys available during video 
runtime."""
    print("The following keys are shortcuts for controlling the video 
during a run:")
    print("Esc - Exit and Close.")
    print("S - Save a Snapshot of the Current Frame")
    print("X - Cycle the Colormap Backwards")
    print("C - Cycle the Colormap forward")
    print("F - Toggle Filtering On/Off")
    print("T - Toggle Temperature Units between C/F")
    print("U - Go back to the previous Interpolation Algorithm")
    print("I - Change the Interpolation Algorithm Used")


if __name__ == "__main__":
    # Pick the mode to run in. Mode corresponds to functions in elif 
below.
    mode=2

    if mode==1:
        take_pic()
    elif mode==2:
        print("camera_read")
        camera_read()
    else:
        print("Incorrect or missing mode.")
```

BIOGRAPHY

Name Miss Thanaporn Chuenpirom

Date of Birth 6 JSeptember 1999

Address 16, Mab Chalude-LeamSon Road, HuaiPong Sub-district,

Muang District, Rayong, 20150

Education 2021 Bachelor of Engineering Biomedical Engineering

King Mongkut's Institute of Technology Ladkrabang

Specialized field 1.) Medical Instruments

2.) Image Processing

Publications and Awards

Present - Publication

“Remote Temperature Monitoring for Infant Incubator using Thermal Camera,"

In 2022 14th Biomedical Engineering International Conference (BMEiCON), Songkhla, Thailand, 2022,