SMART AND SUSTAINABLE IOT BASED HUMIDITY AND TEMPERATURE CONTROL SYSTEM FOR VEGETABLE/FRUIT STORAGE

CHAN YI CHUEN

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Electronics Engineering with Honours

Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

May 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ______________________

Name : Chan Yi Chuen

ID No. : 19AGB02358

Date : 19 April 2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SMART AND SUSTAINABLE IOT BASED HUMIDITY AND TEMPERATURE CONTROL SYSTEM FOR VEGETABLE/FRUIT STORAGE"** was prepared by **CHAN YI CHUEN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Electronics Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ______________________

Supervisor : Dr. Toh Pek Lan

Date : 22 April 2024

Specially dedicated to

my beloved mother and father

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Toh Pek Lan for her invaluable advice, guidance, and her enormous patience throughout the development of the research. In addition, I would like to express my gratitude to my loving parent and friends who had helped and given me mental support and encouragement throughout the project. Other than that, I would also like to thank UTAR as this work was supported by UTAR Research Fund Project with the grant number IPSR/RMC/UTARRF/2023-C1/O01, 2023.

SMART AND SUSTAINABLE IOT BASED HUMIDITY AND TEMPERATURE CONTROL SYSTEM FOR VEGETABLE/FRUIT STORAGE

ABSTRACT

Traditional post-harvest techniques serve as a crucial component in agricultural activities. However, due to climate change and escalating global demands, these methods are facing increasing challenges. The lack of advanced technology and the disregard for the impact of humidity and temperature on fresh produce have led to an anticipated rise in food loss and wastage along the supply chain. Currently, there is an urgent need for innovative solutions to minimize post-harvest losses and enhance food security. This project introduces a futuristic concept aimed at manipulating post-harvest practices. The focus is to implement an Internet of Things (IoT) cold storage chamber that monitors temperature, humidity, and carbon dioxide levels in real-time. This research also investigates the application of technologies to control spoilage, enhance quality, and contribute to a more sustainable agricultural ecosystem. It offers a sustainable and effective approach to reduce post-harvest losses, addressing the shortcomings of traditional storage methods. Experimental observations show that fresh produce stored in the chamber maintained better in terms of weight loss and appearance compared to those stored at room temperature. This indicates the effectiveness of the IoT cold storage chamber in preserving the quality of fresh produce, thus reducing wastage. Looking forward, the integration of technology in agriculture not only ensures a more predictable and healthy food supply but also signals a shift towards a future where agriculture is able to coexists with our changing environment. Hence, enhancing post-harvest practices through innovative ideas like the IoT cold storage chamber is not just achievable, but also essential for the future of agriculture.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AC	Alternating Current
CO_2	Carbon Dioxide
DC	Direct Current
eCO_2	Equivalent Carbon Dioxide
EPS	Expanded Polystyrene
FYP	Final Year Project
GPIO	General Purpose Input Output
HVAC	Heating, Ventilation, and Air Conditioning
I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IoT	Internet of Things
LPDDR	Low-Power Double Data Rate
MCU	Microcontroller Unit
MOX	Metal Oxide Sensor
NTC	Negative Temperature Coefficient
OS	Operating System
PID	Proportional–integral–derivative
PVC	Poly Vinyl Chloride
RTV	Room Temperature Vulcanising
RWL	Relative Weight Loss
SDRAM	Synchronous Dynamic Access Memory
TVOC	Total Volatile Organic Compound
USB	Universal Serial Bus
VOC	Volatile Organic Compound
Wi-Fi	Wireless Fidelity
XPS	Extruded Polystyrene

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

In recent years, the integration of technology and agriculture has given rise to a transformative approach known as the Internet of Things (IoT) in agriculture. The IoT has revolutionized various industries by connecting everyday objects to the internet, enabling them to gather and exchange data in real-time. In the agricultural sector, IoT has emerged as a powerful tool, offering innovative solutions to enhance productivity, sustainability, and efficiency (Friha *et al*., 2021). IoT devices including sensors are deployed to collect and transmit data of the fresh produce cold chain. They play a crucial role in monitoring and maintaining optimal conditions during the transportation of fresh produce, specifically by tracking temperature and humidity levels. Maintaining the right environmental conditions is vital to preserving the quality and shelf life of fresh produce. These conditions are being monitored in the cold storage during transportation or when serving as a temporary storage facility for fresh produce. Cold storage plays a crucial role in the cold chain process as it enables the preservation of goods for an extended period before their utilization. They help to maintain the freshness, quality, and safety of the fresh produce throughout the transportation process (Mohammed, Riad and Alqahtani, 2022). A cold storage equipped with IoT monitoring system can be designed to suite this application. The storage device will use temperature, humidity, and carbon dioxide (CO_2) sensors to monitor the environment of the device. These condition means a lot for the fresh produce, fruits, and vegetables, as they have specific requirements to slow down the rate of spoilage and maintain their quality for a longer period (Gast, 2019). At higher temperatures accelerate the respiration and metabolic rate of vegetables, leading to a more rapid deterioration, resulting in the increase of moisture loss from the vegetables. Cold temperatures help inhibit the growth of bacteria that can cause spoilage and foodborne illnesses. This is due to the growth of these microorganisms is significantly slowed down, reducing the risk of foodborne pathogens, and maintaining food safety (Ahmad and Siddiqui, 2015). Furthermore, regulating humidity levels can prevent excess moisture that can lead to mold growth or condensation that can damage the produce. By maintaining appropriate humidity levels, the quality and texture of the fresh produce can be preserved (Sensitech, n.d.). Moreover, keeping track of the carbon dioxide will be able to monitor the respiration of the fresh produce. Fruits and vegetables continue to respire even after being harvested, consuming oxygen, and producing carbon dioxide. If CO_2 levels are not regulated, the accumulation of CO_2 can lead to adverse effects, such as accelerated ripening, premature senescence (aging), and increased susceptibility to decay (Kinhal, 2024). Cold storage provides a controlled environment where the temperature, humidity and CO_2 can be maintained within the optimal range for the specific type of produce. This helps to extend its shelf life and preserve its taste, appearance, and nutritional value. The world is increasingly adopting cold chain logistics and incorporating IoT sensors to track and monitor goods (Cil, Abdurahman, and Cil, 2022). With this system, it enables easy identification of issues and determine if the shipment has been compromised.

1.2 Problem Statements

Proper handling and secure post-harvest systems play a vital role in the agriculture industry. Neglecting to protect crops after harvest can result in substantial losses and wasted resources (Naik, Amin and Mahdi, 2022). In more severe instances of post-harvest losses, can lead to business failures and a significant decline in income, particularly impacting farmers. Farms globally, particularly those managed by smallholder farmers, often rely on traditional storage practices that inadvertently harm their harvests. Thus, postharvest lost are mainly caused by postharvest handling.

Temperature control is crucial for preserving the quality and extending the shelf life of harvested crops. Inadequate temperature management during postharvest handling, such as exposure to high temperatures or temperature fluctuations, can accelerate the spoilage process. Furthermore, managing moisture levels is essential to prevent the growth of mold, decay, and quality (weight, appearance, colour) degradation in harvested crops (Food and Agriculture Organization (FAO), 2009). Excessive moisture can contribute to elevated microbial activity, while insufficient moisture can lead to dehydration and shrinkage. Thus, this project proposed to develop an environmentally friendly Internet of Things (IoT) storage chamber equipped with various sensors to extend the shelf life of fresh harvests during transportation and regular storage.

1.3 Aims and Objectives

This project aims to enhance the shelf life of fresh produce through the implementation of a IoT based storage chamber. To ensure optimal conditions, temperature, humidity, and carbon dioxide (CO_2) sensors are installed to keep track of the real-time environment within the storage chamber. The objectives of the project are shown as following:

i. To establish an automated IoT-based control system that enables precise temperature and humidity control and monitors carbon dioxide levels within the storage chamber.
ii. To conduct tests and analyse the impact on shelf life in different conditions storing the fresh produce.

CHAPTER 2

LITERATURE REVIEW

2.1 Background of Study

Internet of Things (IoT) devices widely incorporate embedded system, where dedicated systems are specifically designed for IoT that utilize microcontrollers and microprocessors along with software to facilitate communication among interconnected devices. During this Fourth Industrial Revolution (Industry 4.0), IoT is expected to find wider applications in various industries due to increasing interconnectivity and smart automation (Venkateshwari and Subramaniam, 2022). Agriculture stands to benefit significantly from IoT, as it enables the monitoring and management of crops and domestic animals. One crucial aspect of agriculture is the efficient transportation and storage of fresh vegetables and fruits. This process must find a balance between simplicity and cost-effectiveness while providing sufficient space to preserve the products under low temperatures for extended periods, thus maintaining their freshness.

2.1.1 Cold Storage Monitoring System using Raspberry Pi

The cold storage monitoring system offers a significant advantage over traditional refrigerators as it incorporates thermostats and sensors to monitor the temperature of enclosed systems. It will not only collect the data but also transmits the entire dataset to a centralized platform in real time. This enables continuous monitoring and control of humidity and temperature using electronic devices (ADRTECHINDIA, 2024). Through this concept, it ensures that the fresh products are stored in a humid and cool environment. This PID system will maintain stable temperature and humidity within the cold storage, responding promptly to any changes detected by the sensors, using misting and cooling systems. The collected data assists in monitoring the shipment of fresh goods and ensuring the maintenance of optimal temperature levels. In current situations, fresh goods will be transported from farms to markets through a long distance and the freshness will be highly reduced. With this IoT system the freshness of the goods will be monitored along the way and ensuring product quality throughout the transportation process. Consequently, by reducing the wastage of goods, the profitability of the farmers will increase.

2.2 Literature Studies

2.2.1 IoT framework for Smart Food Monitoring System by Srivasta and Gulati (2016)

This paper introduces a system designed to analyse the environmental conditions in which food items are stored and transported. The system detects and measures the temperature, moisture, and light parameters of the surrounding environment, as these factors have an impact on the nutritional values of food items. It uses a Raspberry Pi to analyse and record the values obtained from the sensors. Subsequently, the values of these parameters are compared to predetermined standard values, which serve as threshold values for their respective parameters. These parameters are sent to a web server to store real-time sensed data values and analysis results. So, the user can view these data values through an Android Mobile Application. The design of the system is shown in Figure 2.1. The system can measure the changes in temperature and humidity, which are the important ambient conditions. These two are the main factors leading to food contamination, so the system prompt the user when these parameters exceed the threshold values.

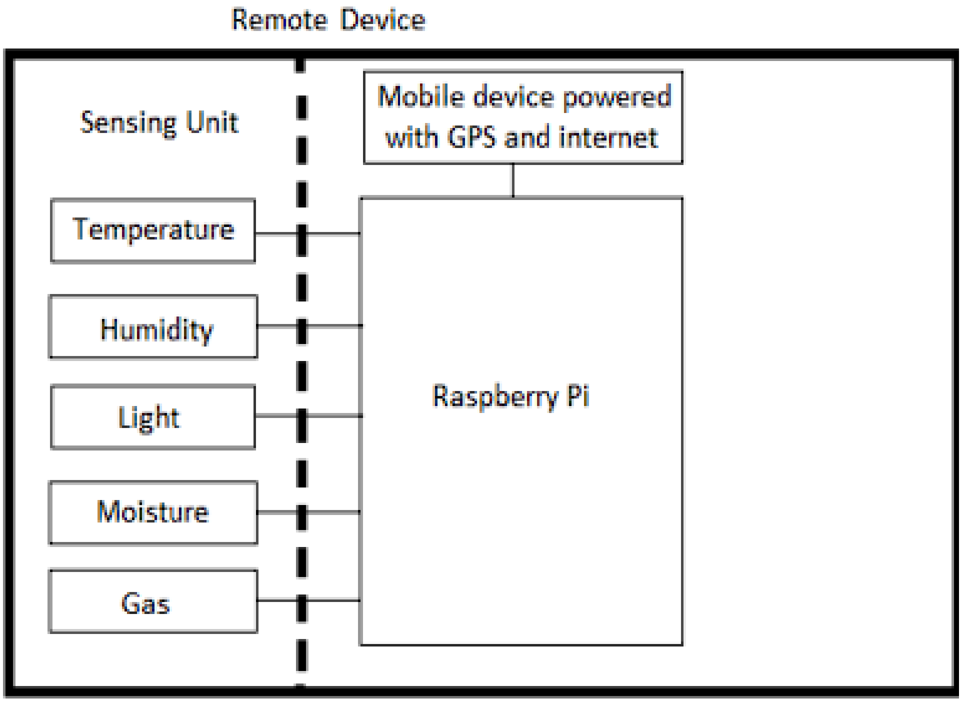


Figure 2.1: Design of Smart Food Monitoring System (Srivasta and Gulati, 2016).

2.2.2 IoT Based Sensor for Humidity and Temperature Measurements in Smart HVAC Systems by Darji (2021)

HVAC stands for Heating, Ventilation, and Air Conditioning, which collectively describe the functions of an HVAC system. This mechanical system is primarily designed to regulate and monitor the environmental conditions within a workspace by controlling the room temperature through heating and cooling methods. The concept of this paper is to develop a smart HVAC system that incorporates a smart sensor for measuring humidity and temperature, along with cloud storage support. The primary function of the smart HVAC system is to provide comprehensive real-time monitoring of the environment, within a given space. It also features automatic temperature control capabilities and transmits information wirelessly to a cloud server. The system design of the smart HVAC system is shown in Figure 2.2. The system powered by an Arduino UNO will obtain data from the DHT11 temperature and humidity sensor for processing and sends it to the cloud server for storage. The system will control the fan speed based on the temperature and humidity reading, and by varying the fan speed will achieve designated conditions in the given space.

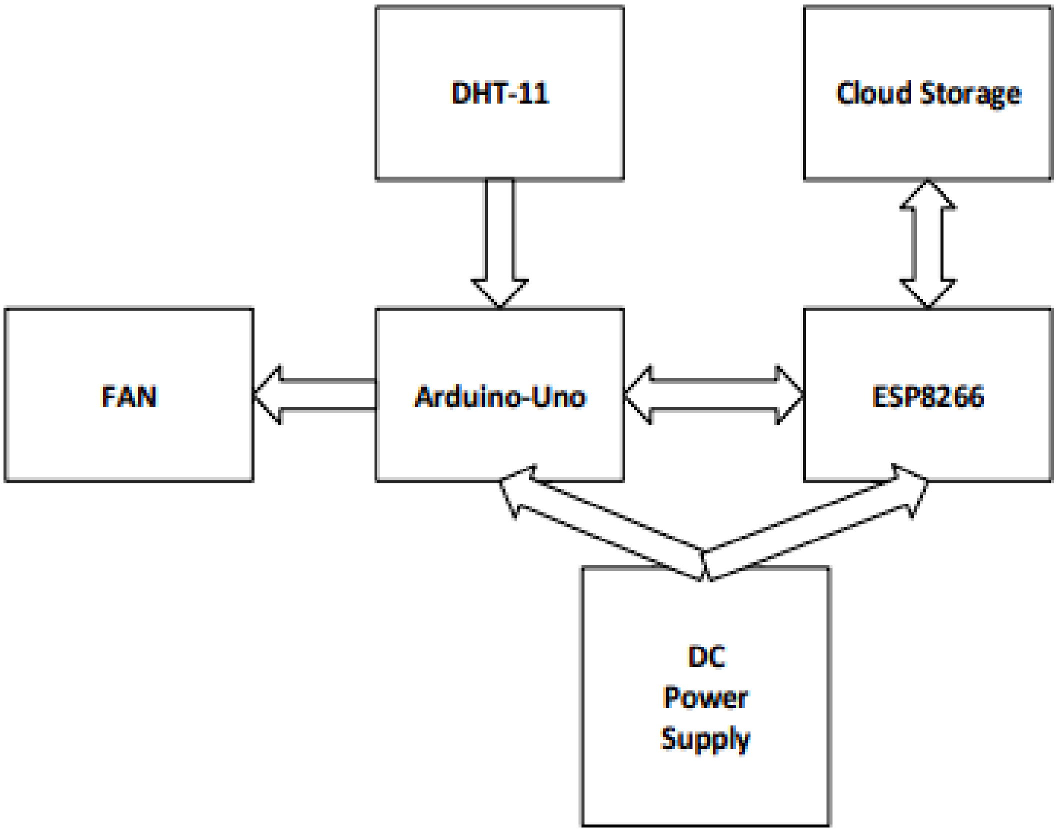


Figure 2.2: Design of Smart HVAC System (Darji, 2021).

2.2.3 IoT and Neural Network-Based Water Pumping Control System for Smart Irrigation by Karar *et al.* (2020)

The article suggested the development of a system that utilizes an Arduino board to process the sensor data and automate the control of the water pump. The sensors employed in the system measure environmental factors such as temperature, humidity, and soil moisture. Based on these measurements, the system estimates the necessary duration for water irrigation operations. The proposed design can be illustrated in Figure 2.3. The system uses environmental sensors including temperature, humidity, and soil moisture sensor to monitor the surrounding conditions, and the Arduino UNO sends these data to the cloud for processing. After processing, the Arduino UNO will receive instructions from the cloud and controls the DC motors. This will pump out suitable amount of water based on real time conditions. The Remote XY app was used as the cloud and displays all relevant data for the user.

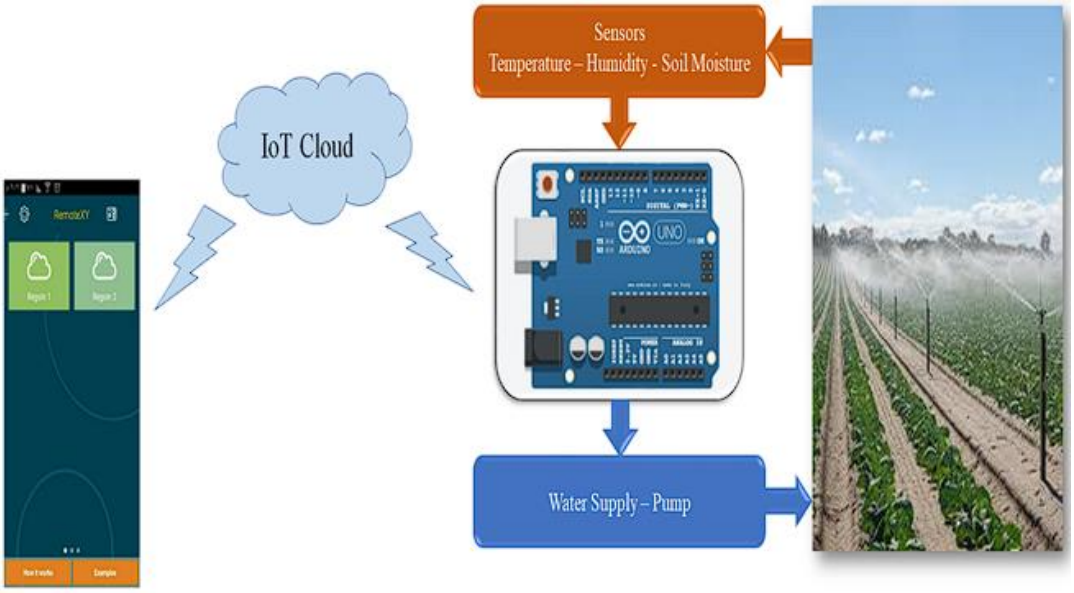


Figure 2.3: Proposed IoT-based Smart Water Irrigation System (Karar *et al.*, 2020).

CHAPTER 3

METHODOLOGY

3.1 Design Architecture

This project uses a Raspberry Pi 3 microcontroller as the computational brain of the system. The cold storage chamber will be built by utilizing a plastic storage container box which is insulated. This is to ensure the cool temperature inside the storage chamber can be maintained. The monitoring system of this project implements two sensors which is the CCS811 carbon dioxide sensor and the DHT22 temperature and humidity sensor. The sensors will be constantly monitoring the real time condition inside the storage chamber and sends the data back to the Raspberry Pi for processing. Furthermore, the cold storage chamber is also equipped with water misting and cooling system. For the water misting system, it is connected to two misting sprayer nozzles and uses a 12V water pump to pump water from the water tank. For the cooling system, a Peltier cooler assembly will be implemented to blow cold air inside the chamber. Both the Peltier assembly and water pump are connected to an external 12V power supply via a relay. The relay will provide a 12V DC to the Peltier module, fans, and water pump when it receives a data signal from the Raspberry Pi. Then, the Raspberry Pi is powered by its own 5V power adapter. The Raspberry Pi will first be connected to a wireless network, so that the air quality data can be sent to the Blynk platform and viewed from a smartphone. Furthermore, to control the cool environment inside the storage chamber, a threshold value 15℃ temperature and 70% humidity will be set. When the temperature rises above the threshold value, the Peltier cooler assembly will be activated to cool down the surrounding inside the storage chamber. When the humidity falls below 70%, the misting system will be activated, and water will be sprinkled from the top of the chamber. Figure 3.1 shows the block diagram of the cold storage chamber system.

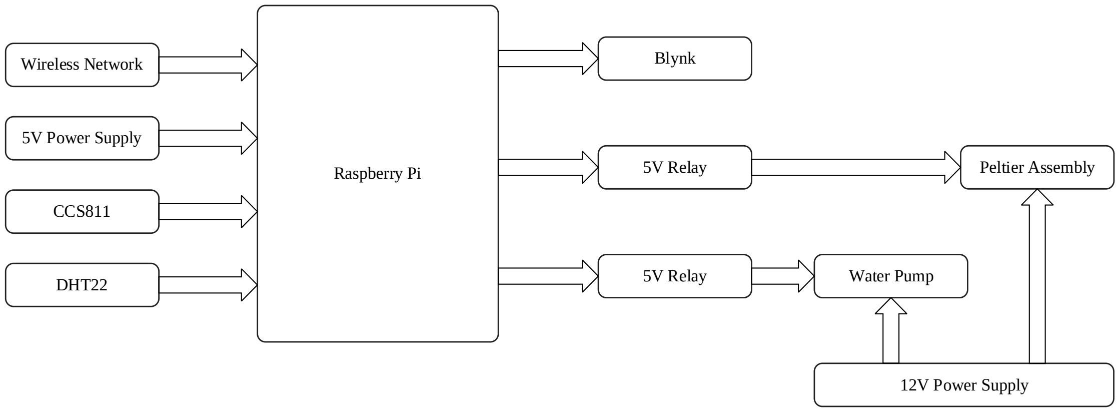


Figure 3.1: Block Diagram of Cold Storage Chamber System.

3.2 Project Management

At the start of the FYP1 semester in Table 3.1, the project title was studied and meeting with the supervisor for further explanation.

Table 3.1: Gantt Chart for FYP1.

Task Name	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project Title Selection														
Meeting with Supervisor														
Research														
Report Writing														
Prototype Building														
Presentation														

After the confirmation of title selection, meetings with the supervisor were carried out to do planning and discussion of the progress of the project. For research, report writing, and prototype building were carried out throughout almost the whole semester.

At the end of the semester, a presentation will be done to explain on the progress and future progress of the project. During the start of FYP2 semester in Table 3.2, the prototype of the chamber had been produced, and small modifications and refinements were made to the chamber. After the design had been finalized, the circuit hardware was installed on to the chamber and test runs were conducted. After the testing stage, the coding was also fine-tuned to match the desired specifications. Then, fresh produces were placed in the chamber to conduct the experiment. The results were recorded and analysed. Furthermore, the report was prepared after obtaining the results and a presentation was conducted. After finalizing the report, a hard copy will be printed and submitted along with other necessary documents.

Table 3.2: Gantt Chart for FYP2.

Task Name	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Prototype Refinements														
Installing Hardware														
Coding Refinements														
Testing the Chamber														
Conducting Experiment														
Analysis of Results														
Report Writing														
Poster Designing														
Presentation														
Document Submission														

3.3 Main Hardware and Software used for the Cold Storage Chamber Management

3.3.1 Raspberry Pi

Raspberry Pi is a small single board computer developed by the Raspberry Pi Foundation in collaboration with Broadcom. In this project, the Raspberry Pi 3 Model B+ will be used. The Raspberry Pi 3 Model B+ features a fast and power-efficient 1.4Ghz Broadcom BCM2837B0, 1GB LPDDR2 SDRAM, 300Mbps Gigabit Ethernet and Dual Band Wi-Fi (Raspberry Pi Ltd, 2023). Figure 3.2 shows the layout of the Raspberry Pi and its available ports. Moreover, the most important part of the Raspberry Pi is the 40 General Purpose Input Output (GPIO) pins that can be utilized to connect with other electronic components and devices. These pins are also individually addressable by software. Table 3.2 shows the specific function of each GPIO pin. Other than the specific function pins the rest are just normal GPIO pins with no special function. Figure 3.3 shows the pinout diagram of the Raspberry Pi. Furthermore, the official operating system of the Raspberry Pi (Raspbian OS) is based on Linux, and it can perform all basic tasks like a native computer. The OS also provides desktop environment and coding with Python can be done directly off the board itself.

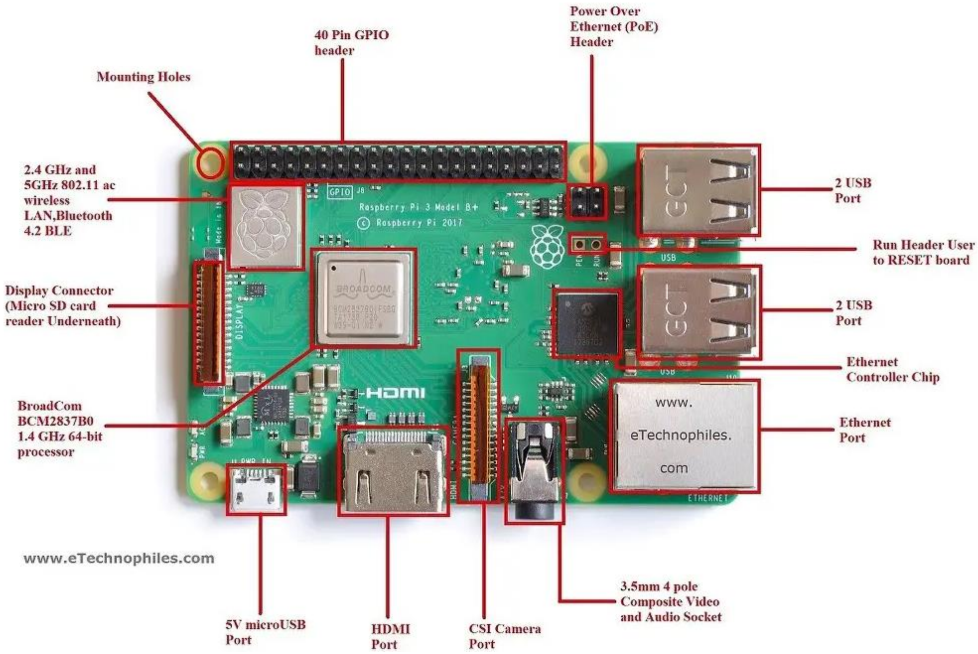


Figure 3.2: Diagram of a Raspberry Pi 3 Model B+ (Negi, 2024).

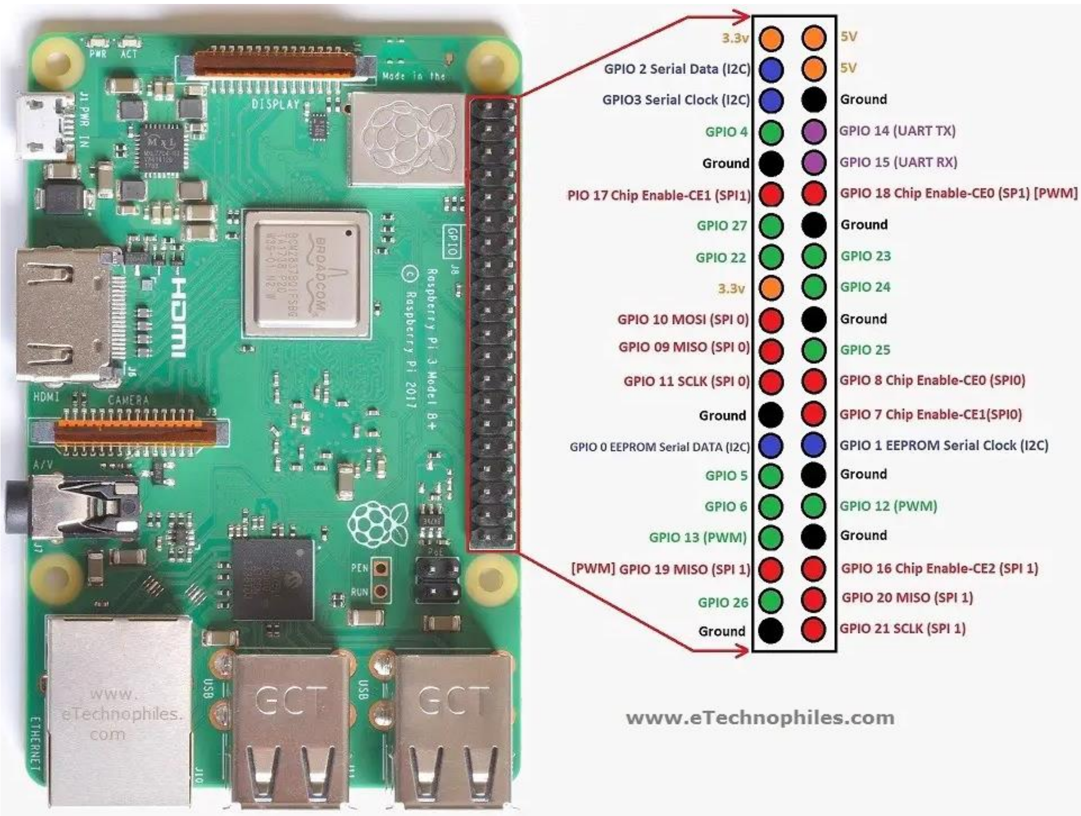


Figure 3.3: Pinout Labels of the Raspberry Pi (Negi, 2024).

Table 3.3: GPIO Pins with Specific Functions.

Purpose	Description	Pin Number
Power	Provide constant 3.3V or 5V DC to external devices	3.3V: 1 & 17 5V: 2 & 4 (Total 4 Pins)
Ground	Voltages measured with respect to the GND voltage	6, 9, 14, 20, 25, 30, 34, 39 (Total 8 Pins)
Pulse Code Modulation (PCM)	Digital representation of sampled analogue. (Can be used for high quality digital audio output for a DAC)	12: CLK (Clock) 35: FS (Frame-sync) 38: DIN (Input) 40: DOUT (Output)

Serial Peripheral Interface (SPI)	Four-wire serial bus, used for data transfer	19: MOSI (Master Out Slave In) 21: MISO (Master In Slave Out) 23: SCLK (Clock) 24: CE0 (Chip Select 0) 26: CE1 (Chip Select 1)
Inter-Integrated Circuit (I2C)	Two-wire communication with compatible external peripherals	Data: 3 (SDA) & 5 (SCL) EEPROM Data: 27 (SDA) & 28 (SCL)
Universal Asynchronous Receive/ Transmitter (UART)	TX pins used to Transmit serial data and RX pins used to receive serial data	Transmit/ Send: 3, 5, 7, 8, 11, 19, 23, 24, 26, 27, 31, 32, 36 Receive: 10, 21, 28, 29, 33 (Total 18 Pins)

3.3.2 DHT22 Module

The DHT22 is a low-cost digital temperature and humidity sensor. This sensor can be readily interfaced with any microcontroller, such as the Raspberry Pi or Arduino, to monitor humidity and temperature in real time. The sensor comes as a module which is shown in Figure 3.4. The DHT22 sensor is installed on the module alongside with a pull up resistor. Inside the sensor, there is a thermistor to measure temperature and a capacitive humidity sensor to measure humidity. The capacitive humidity sensor consists of two electrodes with a moisture holding substrate as a dielectric between them. As the humidity levels alter, it will change the capacitance value of the sensor. The IC inside the sensor will interpret the value and sends out a digital signal. While for the temperature part, the DHT22 uses a Negative Temperature Coefficient (NTC) thermistor to sense for temperature change. When the temperature increases, the resistance of the NTC thermistor decreases, which the sensor will detect the resistance value and translate it to degree Celsius. value and translate it to degree Celsius. The sensing temperature of the DHT22 sensors ranges from -40 to $80℃$ with a $\pm 0.5℃$ accuracy. While for the humidity, it ranges from 0 to 100% with a $\pm 2\%$ accuracy. Furthermore, the sampling rate of the sensor is $0.5Hz$ which gives one reading per two seconds (ELPROCUS, 2019).

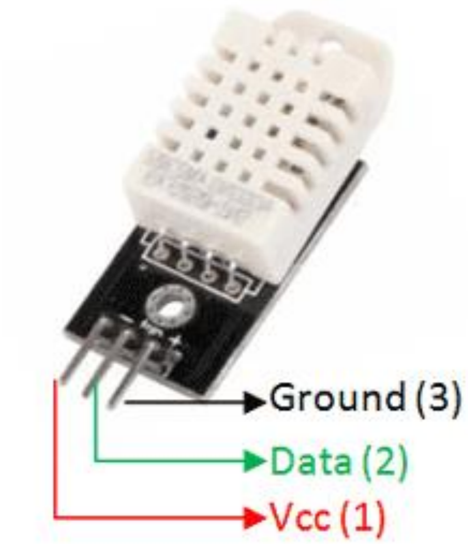


Figure 3.4: DHT22 Sensor Module with Pinout Labels (Components101, 2018).

3.3.3 CCS811 Module

The CCS811 sensor is used to monitor surrounding air quality, it can detect a wide range of Volatile Organic Compounds (VOCs) such as Alcohols, Aldehydes, Ketones, Organic Acids etc. This module has the CCS811 Metal Oxide Sensor (MOX) and a Microcontroller Unit (MCU) to detect VOCs. The module operates in such a way that the MCU reads the analogue voltage from the MOX and compares it with a $10K$ NTC thermistor that matches the equilibrium resistance of the CCS811, then it will return a Total Volatile Organic Compound (TVOC) reading and an Equivalent Carbon Dioxide (eCO_2) reading (Miller, Rembor, and Nelson, 2017). The front view of the module is shown in Figure 3.5. The sensor module has eight pins, but only five pins are connected to the Raspberry Pi, which is Power (V_{cc}), Ground (GND), Serial Clock (SCL), Serial Data (SDA) and Wake (WAK). It requires 3.3V to power the module and the digital signals will be sent through the Inter-Integrated Circuit (I^2C) protocol to the Raspberry Pi.

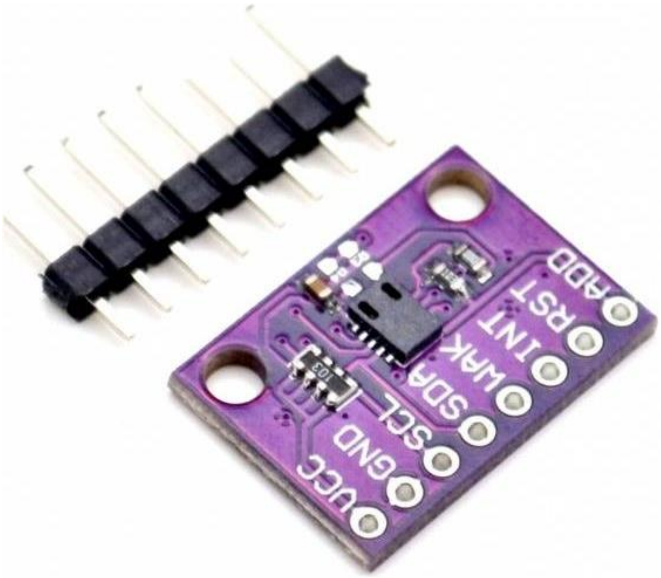


Figure 3.5: CCS811 Module Front View (Sainapse, n.d.).

3.3.4 Peltier Cooler Assembly

The Peltier cooler assembly has been used to blow cool air into the chamber and provide cool environment inside the chamber. The assembly was built with a Peltier module, heatsinks, and fans, respectively. The assembly was assembled in a way that the Peltier module has been sandwiched between a large heat-dissipating heatsink and a smaller cold-dissipating heatsink. The hot side of the Peltier module is in contact with the heat dissipating heatsink and a large 12V DC cooling fan has been attached to the larger heatsink to dissipate heat produced by the module to the surrounding. However, the module's cold side is in contact with the smaller cold dissipating heatsink, and a smaller 12V DC blower fan has been attached to the heatsink to blow air through the cold heatsink and produce a lower temperature air and circulating inside the chamber. Figure 3.6 shows the front and back view of the assembled Peltier cooler assembly. The Peltier module (Thermoelectric Cooler) is built with two types of semiconductor elements, which are sandwiched between two parallel plates and enclosed within copper substrates. When electricity is applied to the module, the junction of those semiconductors will either absorb or release heat, depending on the direction of the current flow. This is known as the “Peltier effect” and this effect will generate a cooling effect on one side and generation of heat on the opposite side (KYOCERA Corporation, n.d.).

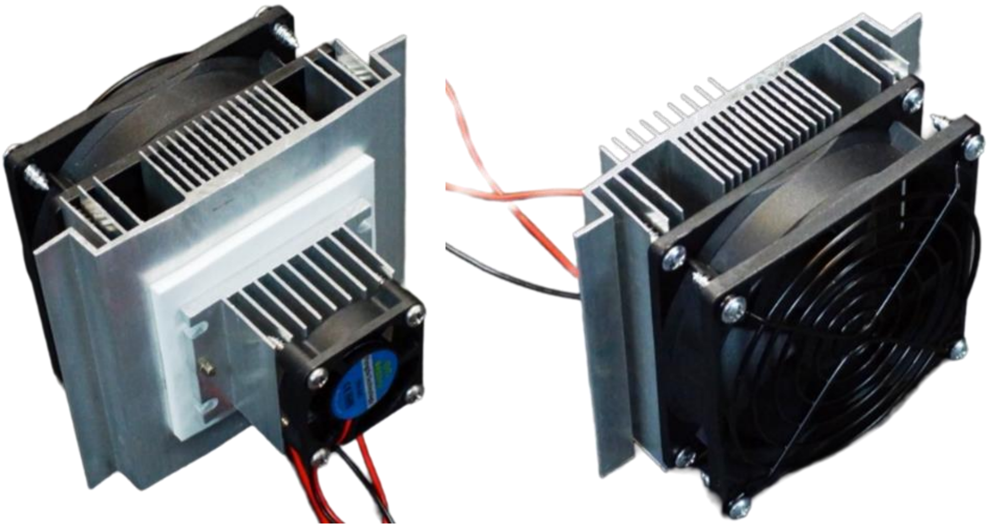


Figure 3.6: Peltier Cooler Assembly (ProtoSupplies, n.d.).

3.3.5 5V Relay

The 5V DC Relay is a commonly used switching device in electronic projects. In this project a four-channel relay has been used. This is because the project implements two water pumps, a Peltier module, and two DC fans. The two water pumps will use two channels (one channel for each pump), Peltier module will use one channel, and the two DC fans of the Peltier cooler assembly will use one channel (two fans wired to one channel). Figure 3.7 shows a four-channel relay with pinout labels. The four-channel relay has four single 5V relays alongside with some switching and isolating components. Each terminal block has six terminals, and each block is shared by two relays. A 5V relay is equipped with a coil that needs only a low voltage to activate. Once the coil receives this small triggering voltage, it generates a magnetic field that operates the relay’s mechanical switch. This switch is capable of handling much higher DC voltages, enabling the low-voltage control signal to effectively control or switch high-power circuits and devices (Components101, 2021).

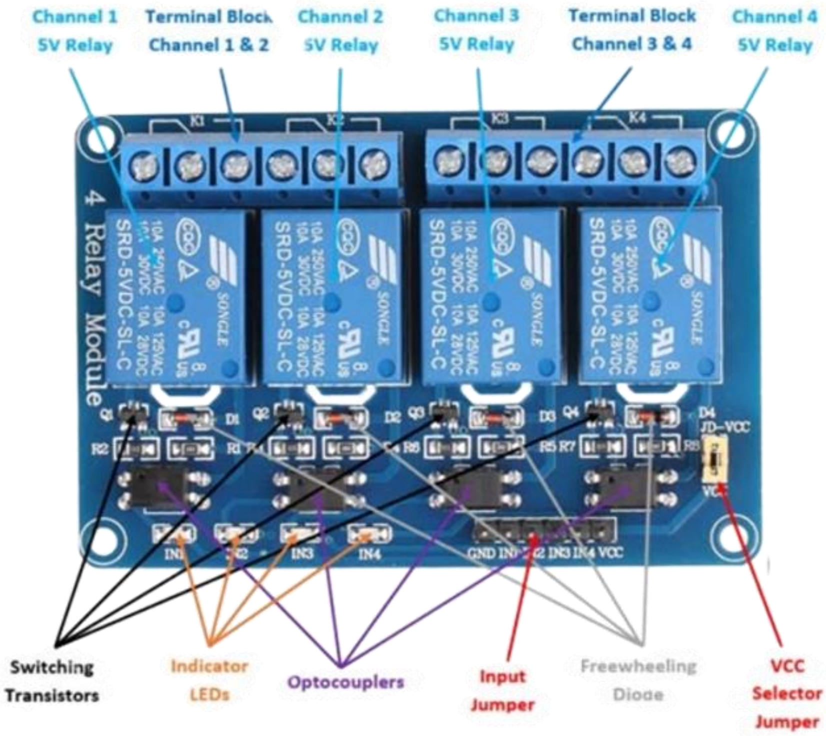


Figure 3.7: Four-channel 5V DC Relay with Pinout Labels (Components101, 2021).

3.3.6 Water Pump

The water pump used in this project is a 12V DC diaphragm pump. Its total power draw of this pump is about 6W, and the flow rate is $1.6L \pm 0.1L/min$. The pump has a maximum inlet water pressure of 0.3MPa and the pump will be connected to two misting heads with capillary tube. Two water pumps will be used in this project, one pump will be utilized to pump water through the misting nozzles and another pump will suck excess water out from the bottom hole of the chamber. Figure 3.8 shows the overview of the water pump.

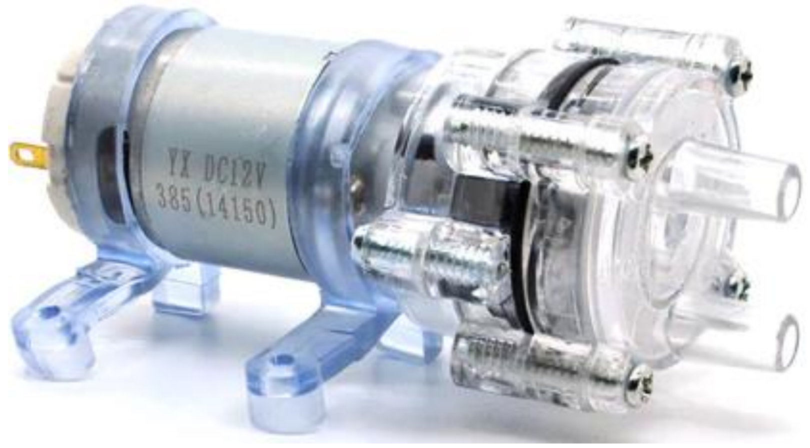


Figure 3.8: 12V DC Water Pump (Gikfun, n.d.).

3.3.7 150W Boost Converter

The 150W Boost Converter has been connected to the Peltier module only. This is to provide a higher current to flow to the module. As there are other 12V components connected to the power supply, the voltage will drop slightly, causing the current passing through the module to be limited. Given by $V = IR$, when the resistance remains constant, and voltage decreases, the current will also decrease. In this case, by using the boost converter to increase the voltage slightly, will allow more current to pass through the Peltier module. Figure 3.9 shows the pinout diagram of the boost converter. A boost converter is a type of switch mode converter that increases an input voltage. The circuit is built with an inductor, a semiconductor switch, a diode, and a capacitor. The operation of a boost converter consists of two primary switching stages which is the ON and OFF states. When the switch is ON, current from the input supply flows through the inductor and the switch, charging the inductor. When the switch is OFF, the inductor's current continues to flow, taking the only pathway available through the diode. This process stores energy in the inductor's magnetic field and transfers it to the capacitor in a way that allows the capacitor's voltage to increase beyond the voltage of the source that supplied energy to the inductor (Keim, 2023).

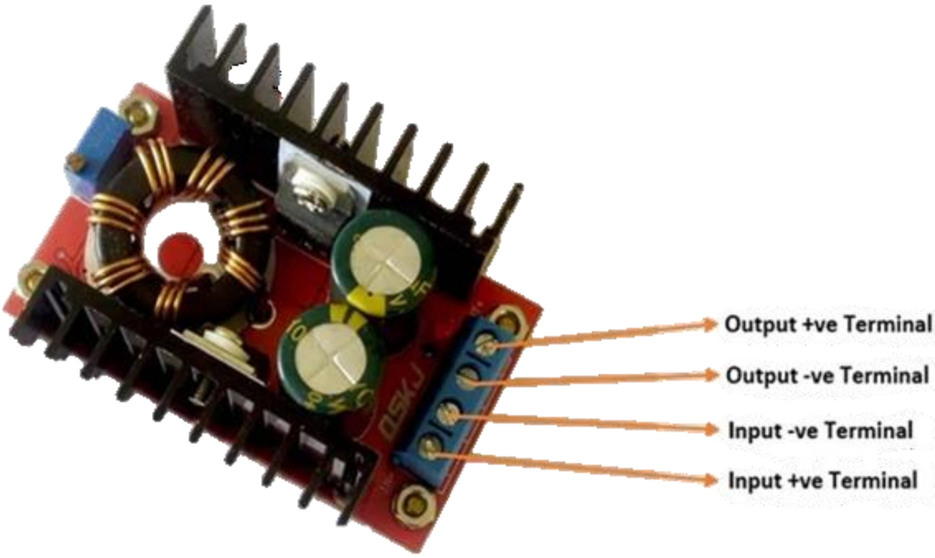


Figure 3.9: 150W Boost Converter Module (Quartz Components, n.d.).

3.3.8 12V Power Adapter

This adapter is used to provide stable 12V DC to the water pumps and Peltier cooler assembly. This type of adapter is chosen because, it is easy to setup and provides the highest portability. The power adapter is connected in series with relays to enable controlling of the 12V devices through the Raspberry Pi. The power adapter used can supply up to 10A, which is more than enough to power all 12V devices. Figure 3.10 shows the power adapter. The power adapter is also known as a power converter as it transforms the high voltage Alternating Current (AC) from the mains into a lower Direct Current (DC) voltage like 12V. It is a switching power supply which uses a bridge rectifier circuit to convert the sinusoidal AC in to pulsed DC voltage. Then, the pulsed DC voltage is smoothed with capacitor. After that, a chopper is used to transform the input voltage into a series of high-frequency pulses, by rapidly toggling the switch. This outputs a pseudo-AC pulse wave, and the voltage is regulated using a high-frequency transformer. Lastly, this voltage is being rectified and smoothed again with a diode and capacitor to produce a smooth and steady DC (Monolithic Power Systems, n.d.).

Figure 3.10: 12V DC Power Adapter with Power Cord.

3.3.9 Blynk

Blynk is used in a wide range of applications, from small IoT projects to millions of commercial connected products. It is a powerful Internet of Things (IoT) platform that allows users to connect their hardware to the cloud. It’s designed to control hardware remotely, store and display sensor data, and visualize the results. Blynk offers a full suite of software allowing to prototype, deploy, and remotely manage connected electronic devices at any scale. The main features of Blynk are device provisioning, data transfer, customized widget for easy data viewing and a comprehensive web console (Blynk.Documentation, 2024). In this project, Blynk is utilized to record and view data received from the sensors. This enables users to monitor the cold storage chamber remotely and able to monitor the status of it in real time. Users can view it with any device browser with internet access or through the Blynk app. Figure 3.12 shows the Blynk app monitoring the parameters inside the chamber.

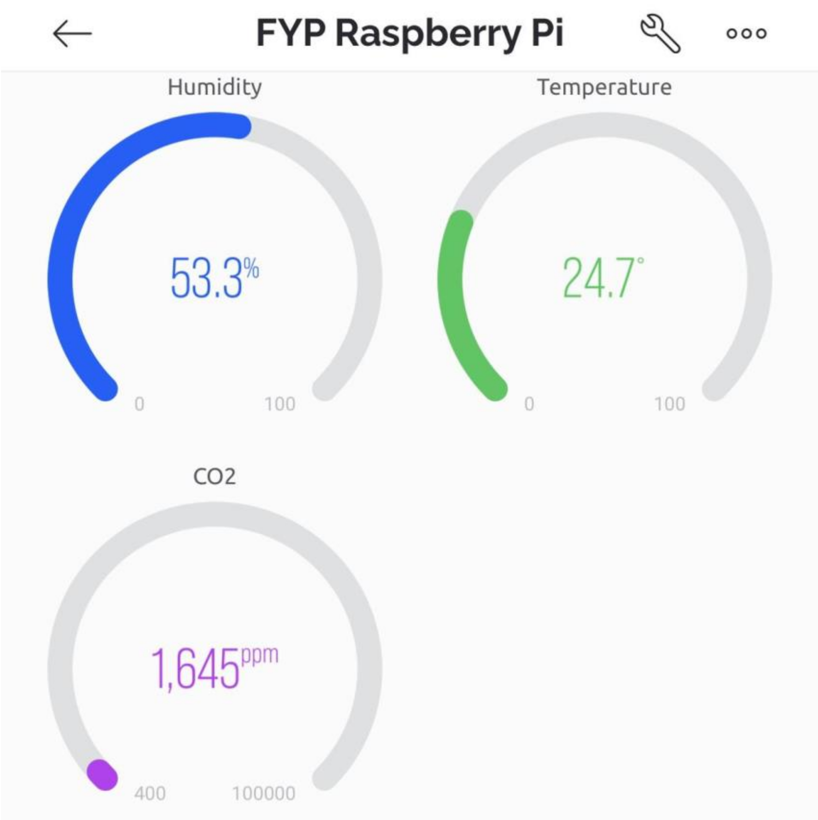


Figure 3.11: Blynk App showing the Humidity, Temperature and CO_2 Values Inside the Cold Storage Chamber in Real Time.

3.4 Circuit Connection of the System

The Raspberry Pi is connected to its 5V DC power adapter via USB Micro B connector. The DHT22 sensor is connected to the corresponding pins on the Raspberry Pi as shown in Table 3.4. The sensor will be connected to 5V power, and the output of the sensor will be sent to the Raspberry Pi through the GPIO 4 pin. For the CCS811 sensor, it is communicating with Raspberry Pi through I^2C, and this is established through a two-wire communication, which is the SCL and SDA pins. Other than that, another 3.3V and ground is required to power the sensor. The last pin to be connected is the WAK pin which is the enable pin to establish the I^2C connection. So, a total of five pins will be required to connect the sensor to the Raspberry Pi. The connection of the CCS811 to the Raspberry Pi is shown in Table 3.5. Furthermore, for the connection of the water pump and the Peltier cooling assembly, both are connected to the Raspberry Pi through the relay due to both components require external 12V DC to work. The connection of them is shown in Table 3.6. Pin 11 and 13 of the Raspberry Pi, which will be triggering channel 1 and 2 of the relay that is connected to both water pumps. While pin 15 and 18 will be triggering channel 3 and 4 of the relay which is connected to the Peltier module and both DC fans. The connection of the water pump and DC fans to the relay is by connecting the positive terminal of the pump to the middle COM port of the relay and the negative terminal of the pump to the negative of the external 12V power. Then the positive of the external 12V power will be connected to the normally open (NO) port of the relay. Furthermore, the connection for the Peltier module is slightly different with the others, as it connects to the boost converter first. The input positive and negative of the boost converter is connected to the 12V power adapter. Then the output is connecting to the Peltier module through the relay. The positive terminal of the module is connected to the middle COM port of the relay and the negative terminal of the module to the output negative of the converter. Then the output positive of the converter will be connected to the normally open (NO) port of the relay. Table 3.7 shows the connection of the boost converter with the power supply. With this configuration, the devices will stay off by default and will only turn on when a trigger signal is received. Figure 3.12 illustrates the connections between the relay, pump, Peltier assembly, and Raspberry Pi.

Table 3.4: Connection of DHT22 sensor to the Raspberry Pi.

DHT22	**Raspberry Pi**	
Pin	**Pin**	**Description**
V_{cc}	2	5V Power
Out	7	GPIO 4
GND	14	Ground

Table 3.5: Connection of CCS811 to the Raspberry Pi.

CCS811	**Raspberry Pi**	
Pin	**Pin**	**Description**
V_{cc}	1	3.3V Power
GND	6	Ground
SCL	5	GPIO 3 (I^2C Clock)
SDA	3	GPIO 2 (I^2C Data)
WAK	9	Ground
INT	No Connect	
RST	No Connect	
ADD	No Connect	

Table 3.6: Connection of Water Pump and Peltier Cooler Assembly to the Relay and Raspberry Pi.

Relay		Raspberry Pi	
Pin	**Description**	**Pin**	**Description**
GND	Ground	20	Ground
V_{cc}	Power	17	3.3V Power
In_1	Channel 1 Trigger	11	GPIO 17
In_2	Channel 2 Trigger	13	GPIO 27
In_3	Channel 3 Trigger	15	GPIO 22
In_4	Channel 4 Trigger	18	GPIO 24

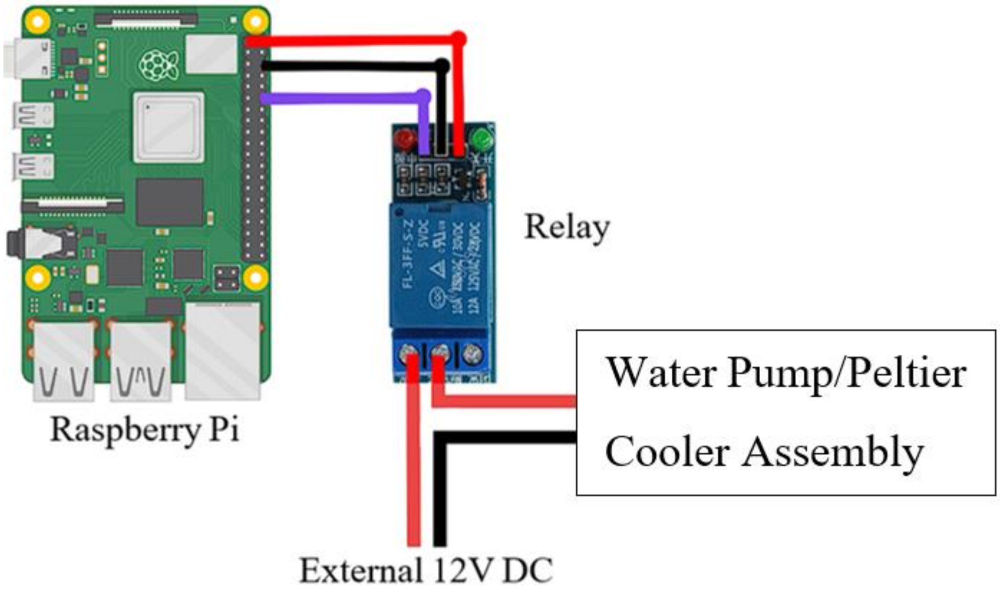


Figure 3.12: Connection between the Relay, Water Pump, Peltier Cooling Assembly and Raspberry Pi.

Table 3.7: Connection of Boost Converter to 12V Power Supply, Relay and Peltier Module.

Boost Converter	**12V Power Supply**
Description	**Description**
Input Positive Terminal	Positive Terminal
Input Negative Terminal	Negative Terminal
Output Negative Terminal	Negative Terminal of Peltier Module
Output Positive Terminal	Normally Open (NO) Port of Relay

3.5 Building the Chamber

The IoT Cold Storage Chamber has been built using a 90L (45.5cm × 67.5cm × 43.0cm) plastic storage box. In the interior surface of the storage box, insulations paddings were installed to minimize the thermal exchange from the inside of the chamber to the surrounding. 3mm thickness Expanded Polystyrene (EPS) Rigid insulation foam was used as insulation and another layer of aluminium foil foam were glued on topped of the EPS to provide waterproofing to the insulation layer. Acrylic double-sided tape and RTV neutral silicon glue was used to stick the EPS to the storage box and sticking the aluminium foil foam to the EPS. For the edges of the insulation, waterproof butyl tape was used to tape the insulation layer shut to the storage box. This tape used provides waterproofing and superior adhesion to ensure no moisture will enter the insulation layer. The same was also done to the lid of the storage box. For the base, it is designed to have a gentle slope to the centre for excess water to drain out. So, the water can be drained out from the fresh produces or the side of the chamber to collect at the centre of the base. A net has also been installed slightly higher than the base to suspend the fresh produces and prevent the produces from soaking in water. Figure 3.13 shows the insulation layer inside of the chamber, while Figure 3.14 shows the net suspending the vegetables.

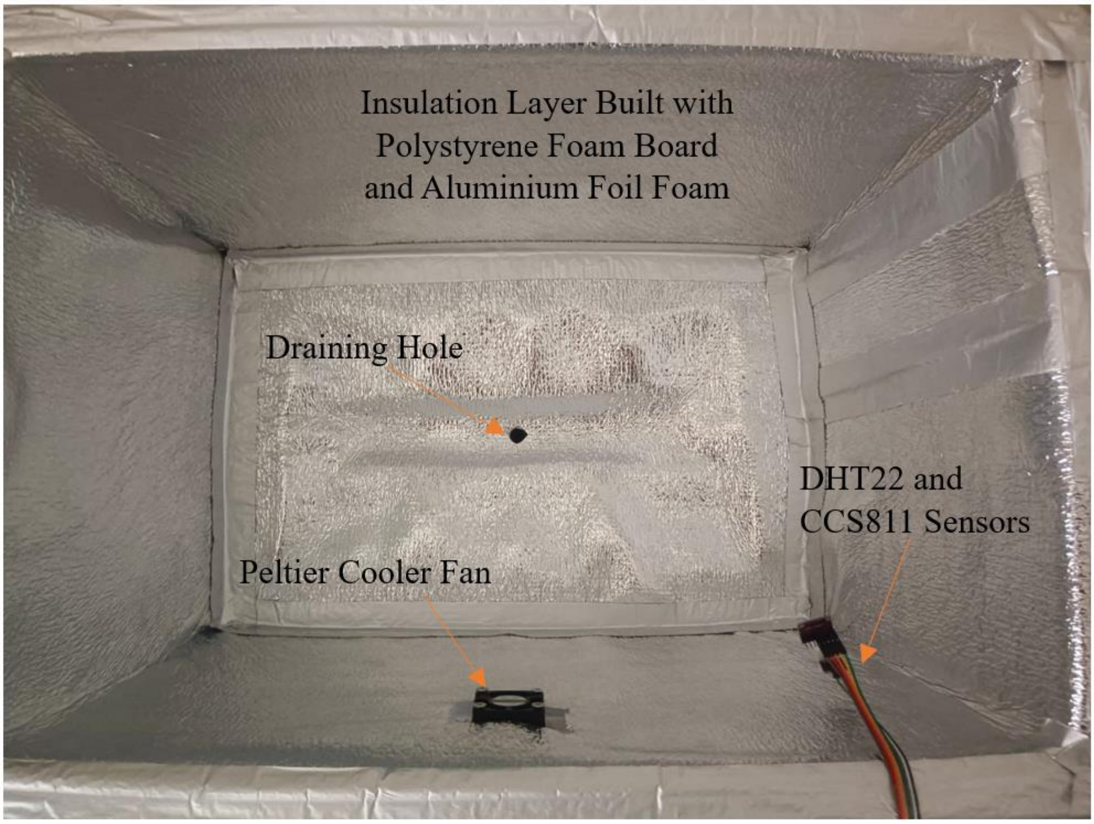


Figure 3.13: Interior of the IoT Cold Storage Chamber.

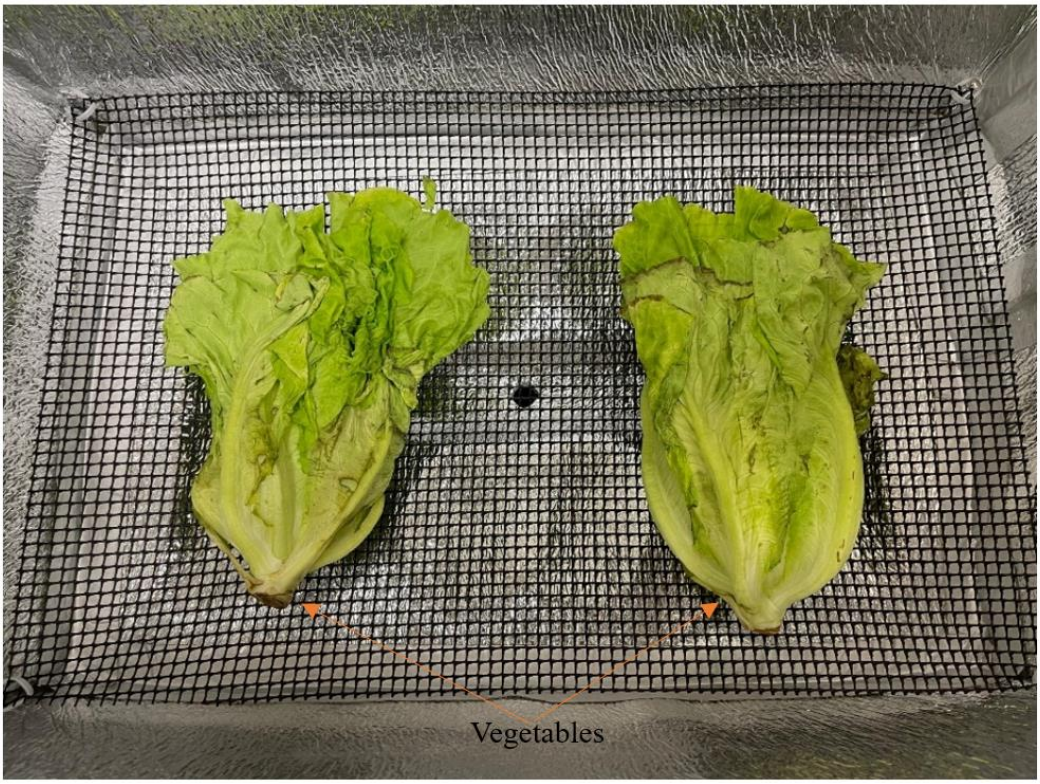


Figure 3.14: Net Installed Above the Base to Suspend Vegetables.

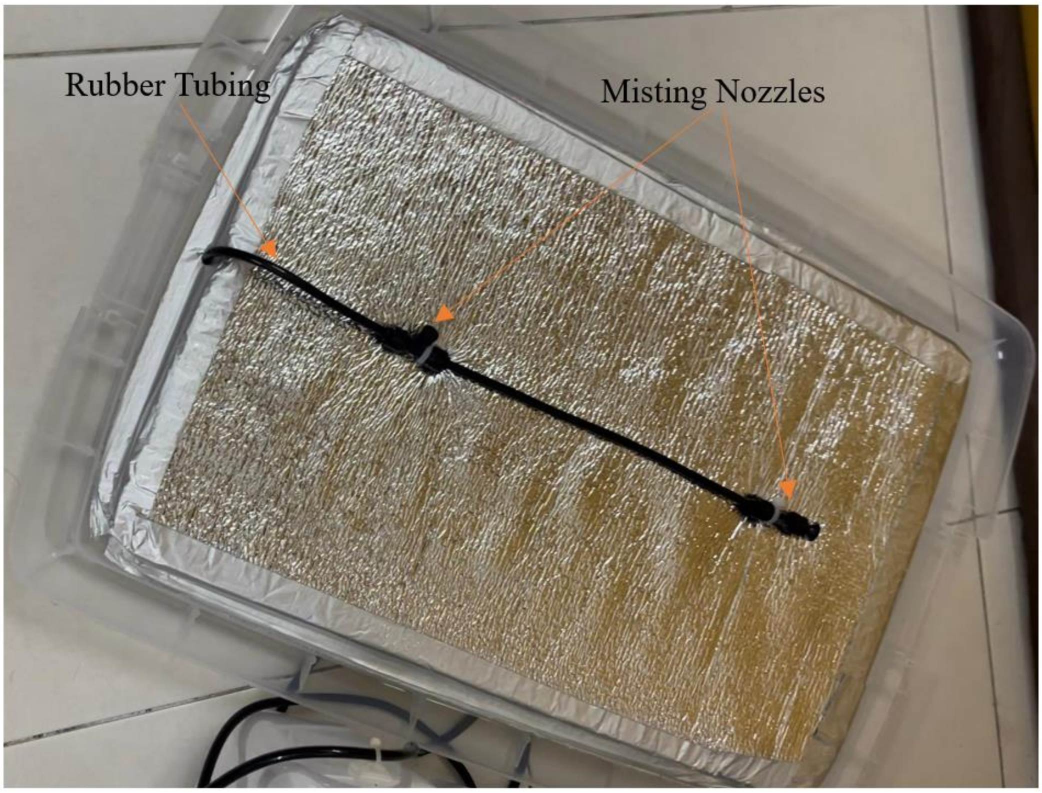


Figure 3.15: Misting Nozzles Installed on the Lid of the Storage Box.

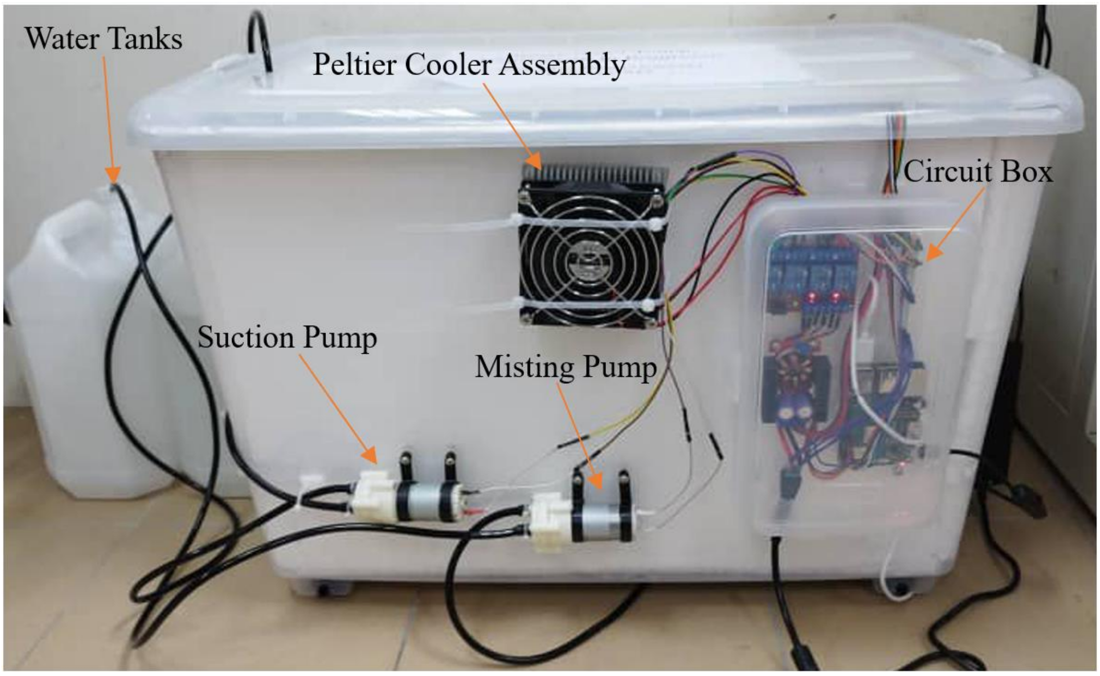


Figure 3.16: Completed Setup Installed on the Backside of the Chamber with Two Water Tanks Beside.

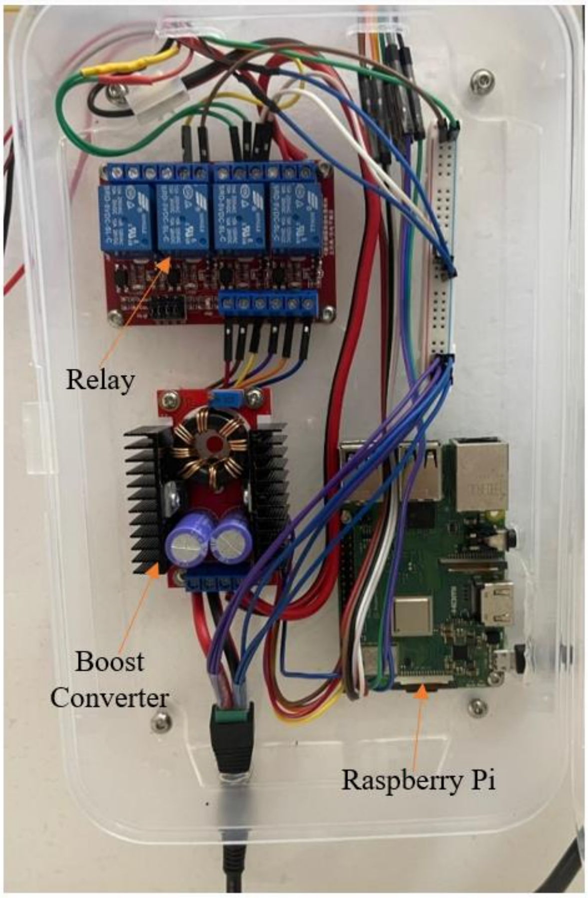


Figure 3.17: Zoomed in View of the Circuitry.

Furthermore, a suitable size hole has been cut at the back of the chamber, to install the Peltier cooler assembly through the chamber. The whole assembly was secured on to the chamber using thick cable ties. After that, two 0.3 mm low-pressure misting nozzles were installed on the lid of the chamber and a small hole was made to channel the rubber tubing from the top side of the lid. The tubing of the misting loop was secured on the lid using thinner cable ties. Figure 3.15 shows the misting loop installed on the lid of the storage box. Moreover, both pumps were screwed on the backside of the chamber and the rubber tubes of the misting loop were connected to the pump. While another pump was used to suck the excess water accumulated on the bottom of the chamber, so one end of the rubber tube was adhered to the bottom of the chamber and another end was connected to the suction pump. Silicon glue was used to seal the connecting part of the chamber and rubber tube. Then, two plastic water tank was prepared on beside the chamber, one acting as the water reservoir for the misting system, while another one is reservoir for collecting excess water.

Table 3.8: Materials and Components Used to Build the IoT Cold Storage Chamber.

No.	Material and Components
1	Raspberry Pi with Raspberry Pi OS installed
2	4-Channel 5V Relay
3	150W DC Boost Converter
4	Peltier Cooler Assembly (Peltier Module & Heatsink Assembly)
5	12V DC Pneumatic Diaphragm Water Pump
6	DHT22 Temperature and Humidity Sensor
7	CCS811 eCO2 Gas Sensor
8	12V/10A DC Power Adapter
9	Misting Nozzles and Rubber Tubing
10	Water Tank
11	90L Plastic Storage Box
12	Dupont Jumper Wires
13	Breadboard
14	14AWG Copper Wires
15	Cable Ties
16	DC Power Plug Jack Connector
17	Terminal Connector Block
18	Screw and Nuts
19	RTV Neutral Silicone Sealant (Multi-Purpose)
20	Waterproof Butyl Tape
21	Acrylic Double-Sided Tape
22	Expanded Polystyrene (EPS) Rigid Foam
23	Aluminium Foil Foam
24	PVC Square Net Mesh

Lastly, a suitable size plastic container acting as a junction box to store all the electronic components inside has been screwed on to the backside of the chamber. The plastic container will protect the Raspberry Pi, relays, and the boost converter from damaging. Figure 3.16 shows the completed circuitry installed on the backside of the chamber, and Figure 3.17 shows the zoomed in view of the circuitry. Finally, the chamber has been turned on to check on every component is working as intended, and to check for any water leaks from the bottom of the chamber. Another crucial step is to adjust the tuning knob of the boost converter to fine tune the voltage being supplied to the Peltier module. The exact voltage value may differ for each module, so trial and error must be implemented to find the best supply voltage, to allow suitable current flow through the module. The rated supply voltage for the module is 12V but exceeding it slightly can also work safely. If the module is over voltage for an extended period, will cause permanent damage to the module. After trial and error, it is found that by allowing around 4.08-4.15A of current flowing through the module will produce the best temperature delta, as per reading from measuring the cold side of the module. If allowing more current higher than this value will cause the cold side temperature to increase as more current passing through it generates too much heat and eventually heating up the cold side of the module too much. Table 3.8 shows the complete list of materials used to build the chamber.

3.6 Program Controlling the Chamber

The program of this project has been designed in Python language. Figure 3.18 illustrates the flow the program. The code can be modified directly in the Raspberry Pi OS, by using the built-in text editor such as Thonny and Geany. Figure 3.19 shows the code being edited in Geany. The ability of this program is to function as PID system to manipulate the environment inside the chamber. So, the program will constantly compare the real-time sensor values with the preset value to ensure certain parameters can be achieved. Figure 3.20 shows the program running in the terminal window. The program will trigger the cooling or misting system to compensate for the parameter that defers with preset value. Moreover, the program is also able to manage errors, such as sensor errors and restarting the program automatically after facing errors. Other than that, the program can log the sensor data into Excel files stored locally, just for a precautionary step to prevent any data loss. Figure 3.21 shows the content logged in the Excel file. The code developed has been attached in the Appendix section. At the start of the code, all necessary modules, and libraries, including those for sensor communication (DHT22 and CCS811), GPIO control, Blynk integration, and data storage (Excel) has been imported. The Blynk authentication token ($BLYNK\_AUTH\_TOKEN$) has been set to enable communication to Blynk. Figure 3.22 shows the web console view of Blynk. The GPIO pins are configured for the relays, to enable the control of relays in different channels. After that, an infinite loop will be executed collects sensor data and performs preset actions. The loop will be constantly reading temperature and humidity from the DHT22 sensor and CO_2 and TVOC values from the CCS811. These values will be sent to the Blynk database and recorded locally in an Excel file sorted based on date and time. With these data values, the program will control the relays to activate the water pump if humidity is below 70% or activate the Peltier cooler assembly based on temperature (above 15°C activates, below deactivates). For the misting system logic, it keeps tracks of the time interval of misting to enable periodic watering and cooldown intervals. This is to ensure the chamber will not be too humid and prevent over misting. Lastly at the end of the loop, there are error handling and termination commands. The command $OSError$ will manage errors such as DHT22 data not ready or a full buffer was not returned. For keyboard interrupt ($Ctrl + C$), the program will clean up GPIO pins (sends a low signal through trigger pin). All the relevant message produced by the program will be written to the Excel sheet, such as "Water Misting," Peltier Cooler Activated/Deactivated" or "DHT22 Error". These messages will be recorded with date and time, so it can be easily traced.

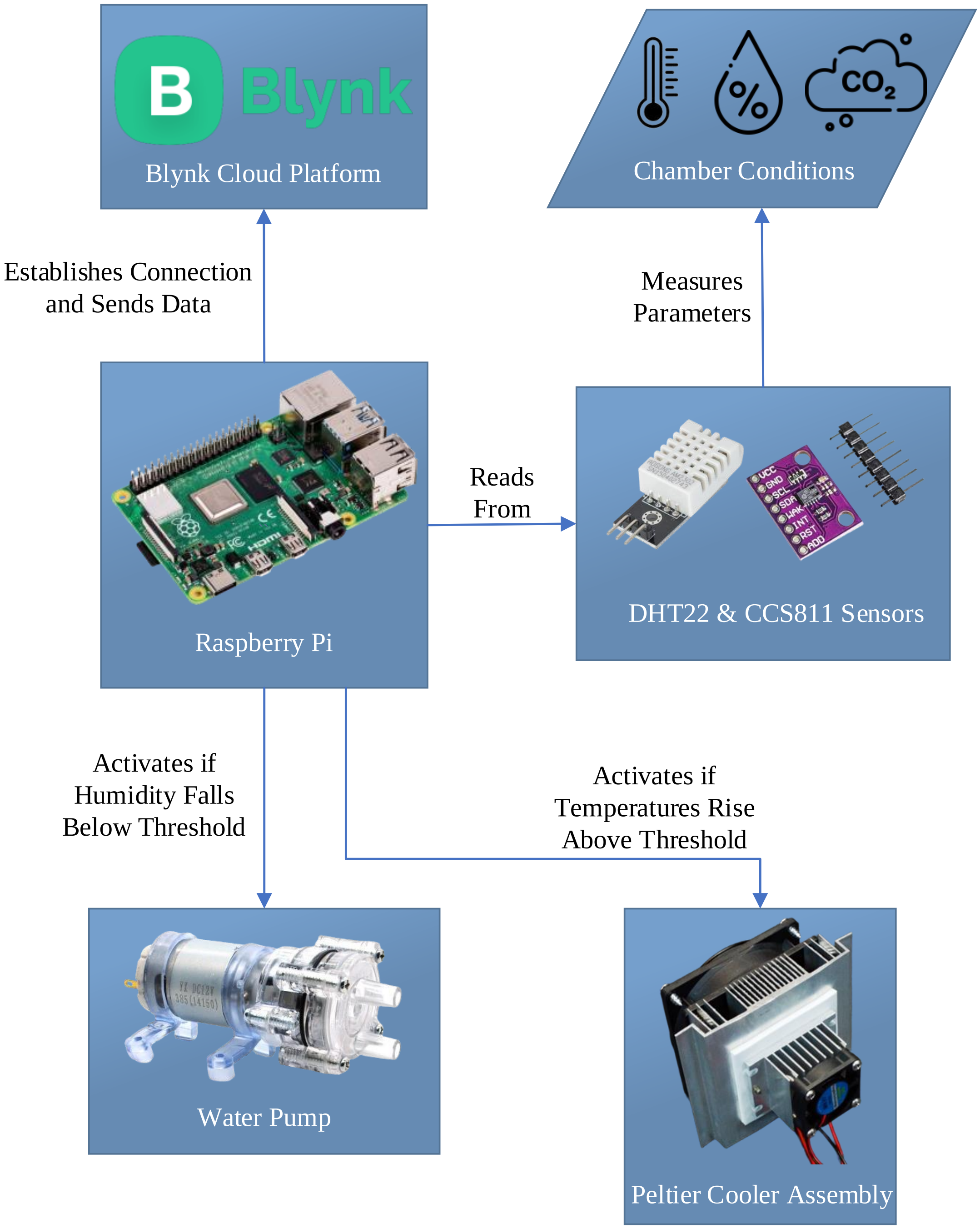


Figure 3.18: Flow of the Program.

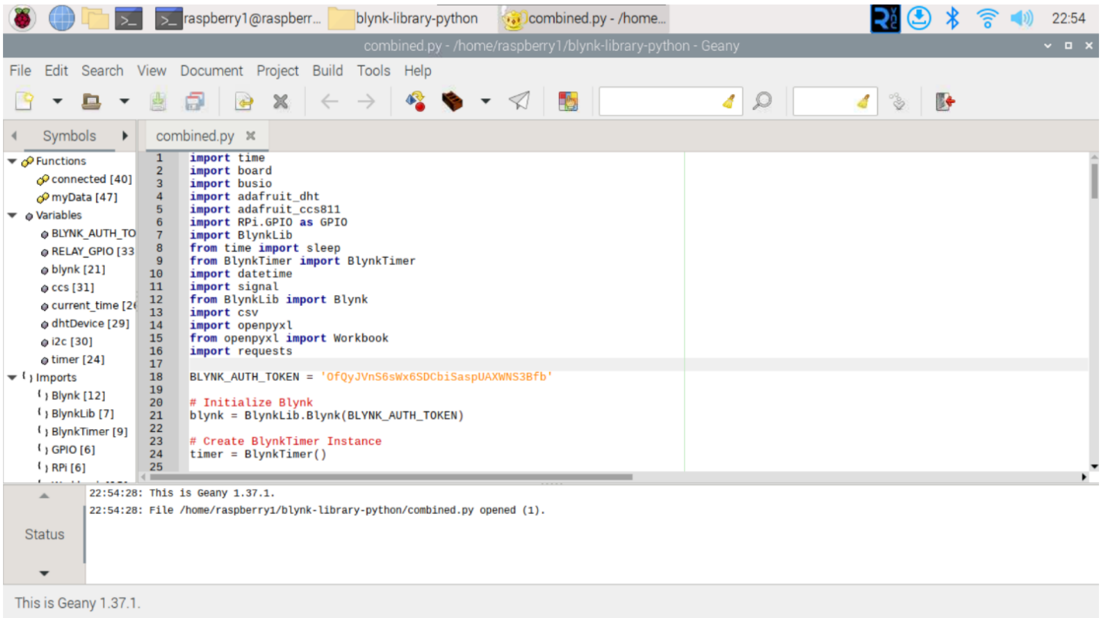


Figure 3.19: Code Editing in Geany.

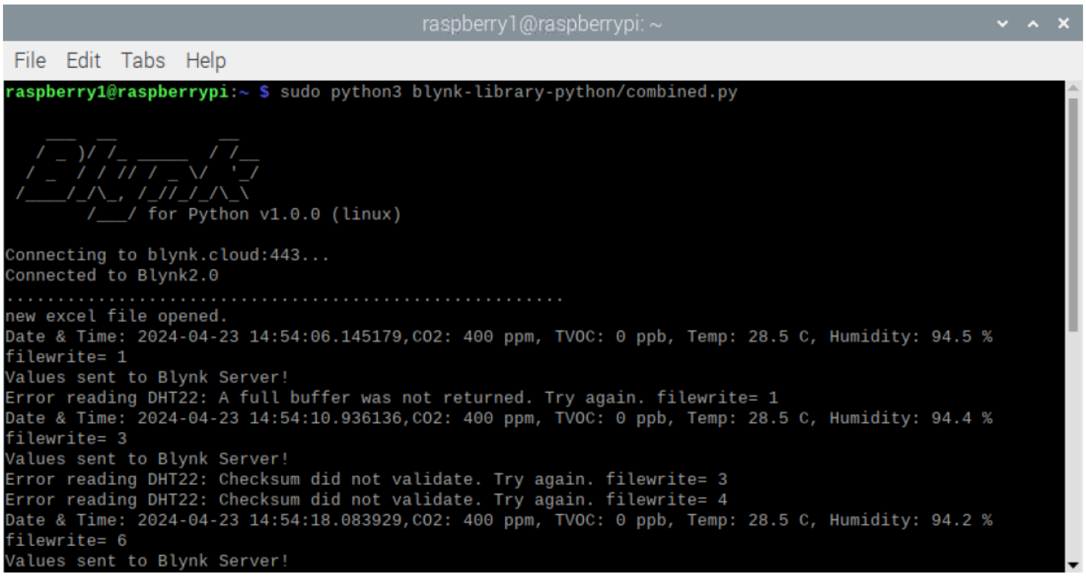
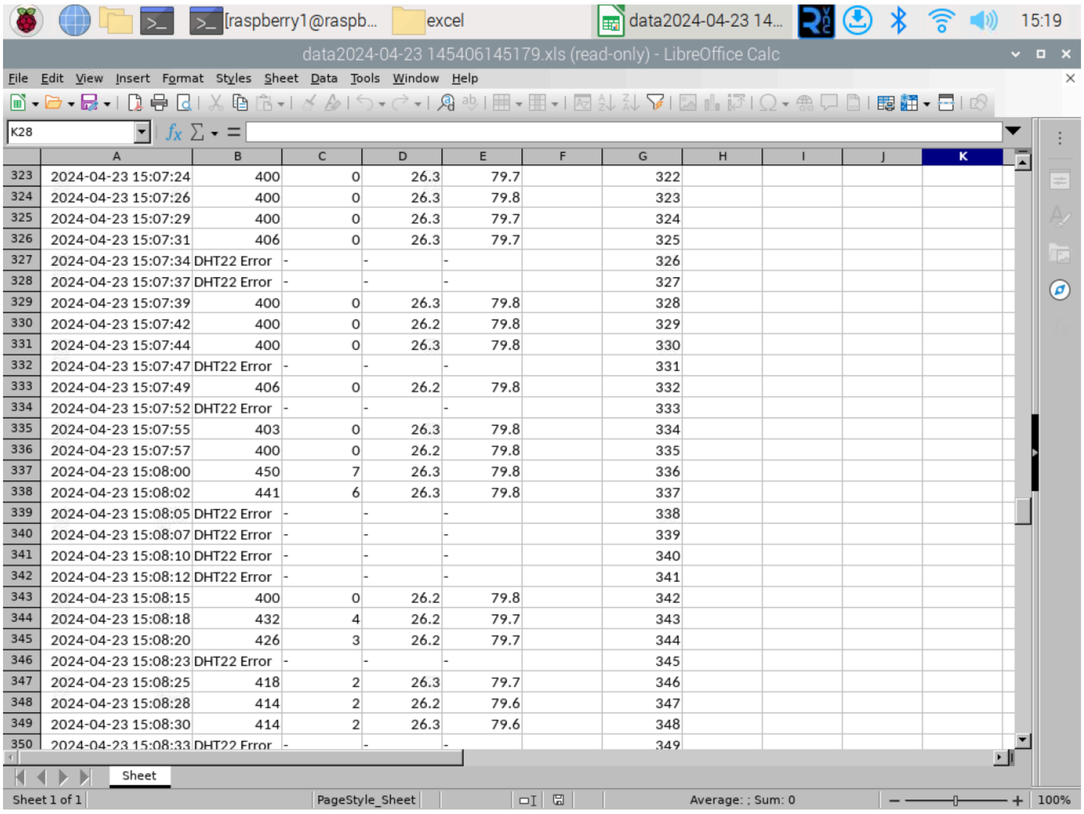


Figure 3.20: Terminal View of the Program Running.



	A	B	C	D	E	F	G
323	2024-04-23 15:07:24	400	0	26.3	79.7		322
324	2024-04-23 15:07:26	400	0	26.3	79.8		323
325	2024-04-23 15:07:29	400	0	26.3	79.7		324
326	2024-04-23 15:07:31	406	0	26.3	79.7		325
327	2024-04-23 15:07:34	DHT22 Error	-	-	-		326
328	2024-04-23 15:07:37	DHT22 Error	-	-	-		327
329	2024-04-23 15:07:39	400	0	26.3	79.8		328
330	2024-04-23 15:07:42	400	0	26.2	79.8		329
331	2024-04-23 15:07:44	400	0	26.3	79.8		330
332	2024-04-23 15:07:47	DHT22 Error	-	-	-		331
333	2024-04-23 15:07:49	406	0	26.2	79.8		332
334	2024-04-23 15:07:52	DHT22 Error	-	-	-		333
335	2024-04-23 15:07:55	403	0	26.3	79.8		334
336	2024-04-23 15:07:57	400	0	26.2	79.8		335
337	2024-04-23 15:08:00	450	7	26.3	79.8		336
338	2024-04-23 15:08:02	441	6	26.3	79.8		337
339	2024-04-23 15:08:05	DHT22 Error	-	-	-		338
340	2024-04-23 15:08:07	DHT22 Error	-	-	-		339
341	2024-04-23 15:08:10	DHT22 Error	-	-	-		340
342	2024-04-23 15:08:12	DHT22 Error	-	-	-		341
343	2024-04-23 15:08:15	400	0	26.2	79.8		342
344	2024-04-23 15:08:18	432	4	26.2	79.7		343
345	2024-04-23 15:08:20	426	3	26.2	79.7		344
346	2024-04-23 15:08:23	DHT22 Error	-	-	-		345
347	2024-04-23 15:08:25	418	2	26.3	79.7		346
348	2024-04-23 15:08:28	414	2	26.2	79.6		347
349	2024-04-23 15:08:30	414	2	26.3	79.6		348
350	2024-04-23 15:08:33	DHT22 Error	-	-	-		349

Figure 3.21: Data Logged in the Excel File.

Figure 3.22: Web View Console of Blynk

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Experiment Results

The experiment has been conducted by using butterhead lettuces. A total of four butterhead lettuces were studied, two were placed in room temperature and another two were place inside of the chamber. The lettuces placed in the chamber lasted for 15 days, while the ones placed room temperature only lasted 10 days. By comparing the lettuces in Figures 4.3 and 4.4, which were the tenth day of experiment, the lettuces in room temperature had dried up and started moulding at the bottom part. While for the lettuces in the chamber, only the most outer part has signs of drying up, the inner part still observed fresh and moisture, and no moulding were found. Based on Figure 4.5, at the fifteen days of experiment, the lettuces in the chamber started moulding, but comparing to the last day of the lettuces in room temperature, the overall appearance of the ones in the chamber are still better than the ones in room temperature at the tenth day, in terms of appearance and moisture.

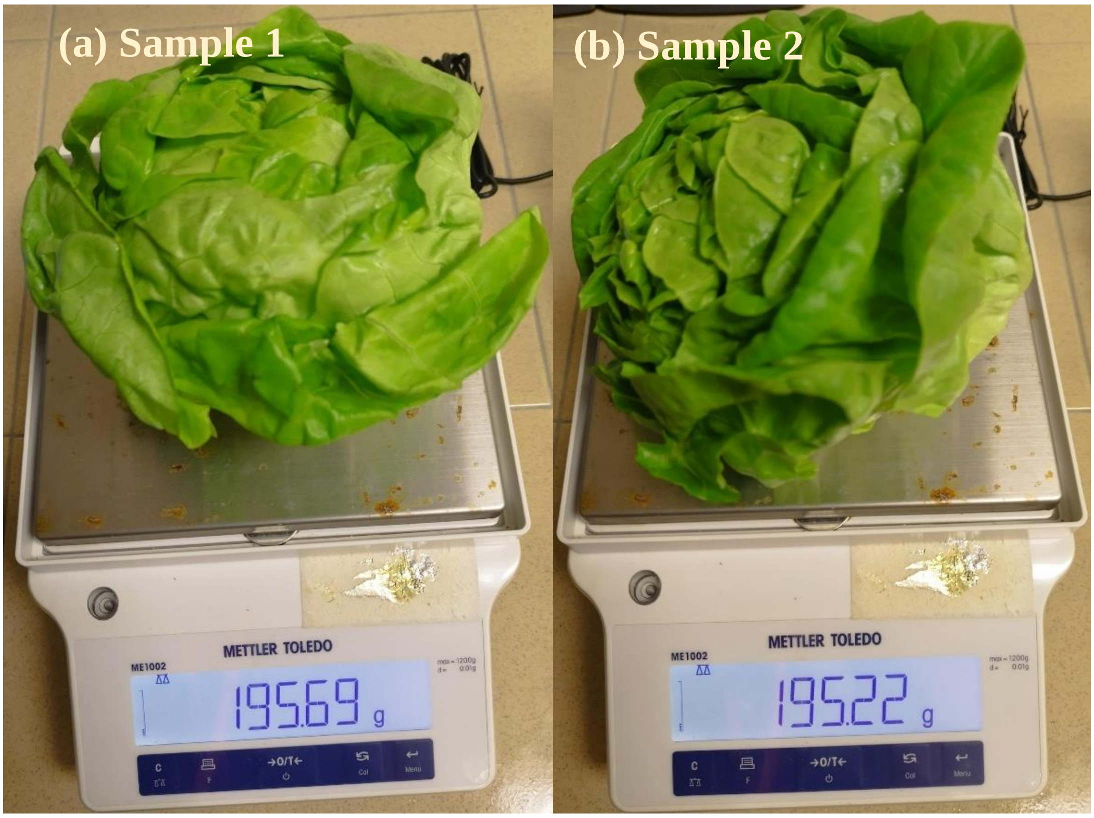


Figure 4.1: Butterhead Lettuces Place in the Chamber (Day 1).

Figure 4.2: Butterhead Lettuces Place in Room Temperature (Day 1).

Figure 4.3: Butterhead Lettuces Place in the Chamber (Day 10).

Figure 4.4: Butterhead Lettuces Place in Room Temperature (Day 10).

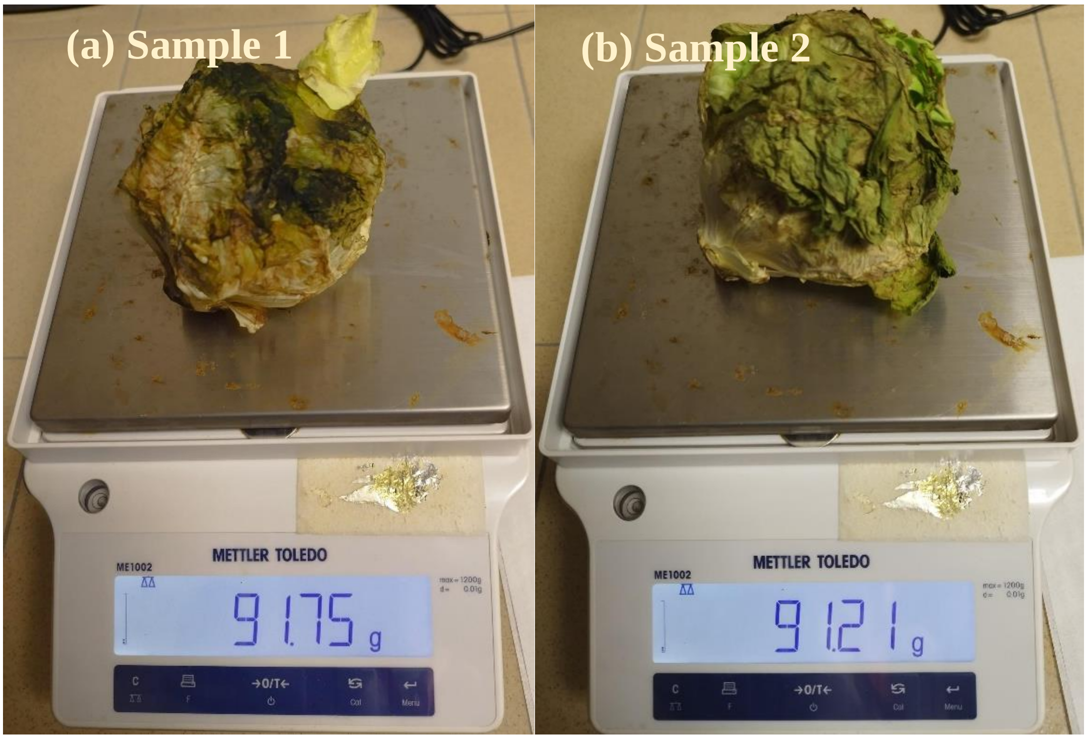


Figure 4.5: Butterhead Lettuces Place in the Chamber (Day 15).

4.2 Analysis of Temperature and Humidity Inside the Chamber

The temperature and humidity inside the cold storage chamber have been measured every two seconds and averaged per day. The program acting as a control system maintained the temperature and humidity around the optimized range as planned in the study. The surrounding temperature and humidity of the studied chamber were fixed at 27°C and 55%, respectively. Table 4.1 and Figure 4.6 present the average temperature and humidity each day throughout the 15 days of data collection. Based on the obtained results, it can be observed that the temperature of the chamber ranges between 21°C to 23°C, while the humidity ranges between 68% to 81%. The average temperature of the studied cool storage system during this period is 21.967°C, and humidity is 72.202%, respectively. According to diagram in Figure 4.6, at the ninth day, there is a humidity spike, hitting a value of 80.495%. The results are verified, and no errors were found. This might be caused by the program having entered a loop and causing water to mist too frequently (program bug). Another reason is the DHT22 sensor has water droplets sprinkled onto it by accident.

Table 4.1: Average Temperature and Humidity of the Studied Chamber in 15 Days.

Day	Temperature (°C)	Humidity (%)
1	21.729	72.635
2	22.391	70.416
3	21.901	68.770
4	22.559	71.845
5	22.066	74.641
6	22.346	72.684
7	22.263	69.676
8	22.026	69.856
9	21.450	80.495
10	21.292	72.997
11	22.105	71.613
12	22.213	71.103
13	21.669	74.010
14	21.821	71.172
15	21.677	71.114

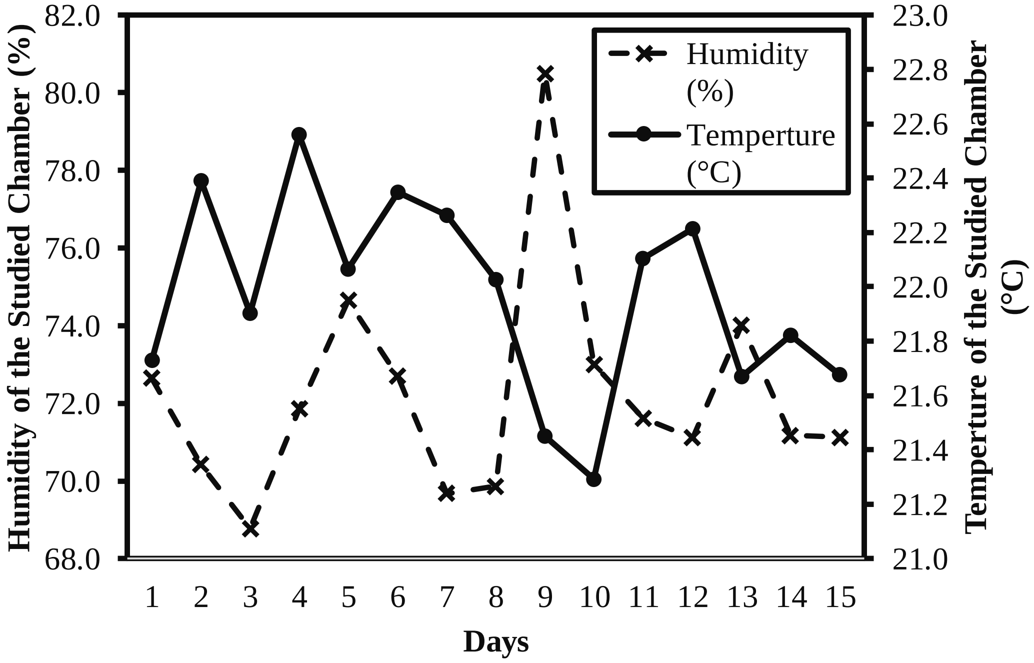


Figure 4.6: Average Temperature and Humidity of the Studied Chamber in 15 Days.

4.3 Analysis of Weight Loss and Appearance Quality of Butterhead Lettuces between the Chamber and Room Temperature

In this study, the freshness of the butterhead lettuces had been prolonged as compared to those stored under room conditions. Tables 4.2 and 4.3 show the appearance rating and description rating of the lettuce, respectively. The weight losses and appearances of the butterhead lettuces were recorded each day until the appearance rating of the butterhead lettuces dropped to one, which indicates serious quality deterioration and presence of mould. In Table 4.2, the relative weight losses (RWLs) of two butterhead lettuces in the studied chamber system are tabulated. The results presented that the cool storage system can consistently slow down the RWLs of butterhead lettuces. For example, the averaged RWL of butterhead lettuces on the tenth day is -31.4960% of the average initial weight. On the fifteenth day, the average RWL is -53.1990% of the initial weight. Moreover, the appearance rating of the butterhead lettuces dropped to one after fifteenth day. In Figure 4.7, the plot diagram demonstrates the average RWLs of butterhead lettuces.

Table 4.2: Rating and Description of the Appearance of Lactuca Sativa Throughout Storage Duration (Belisle *et al.*, 2021).

Rating	Description of the Appearance
5: Excellent	The lettuce is fresh, turgid with greenish bright colour, zero defect.
4: Good	The lettuce is well grounded and fresh, turgid, and bright greenish colour. The leaves were slightly browning or defect with size less than 0.5 cm.
3: Fair	The lettuce is slightly wilt, lightly browning, and the lettuce leaves were dull greenish. The defect size is more than 0.5 cm. The lettuce was at the lower limit of the marketability.
2: Slightly defect	The lettuce is wilt, moderately browning of colours, the appearance is seriously defective with more than 1 cm. The lettuce is not marketable.
1: Extremely defect	The lettuce is seriously deteriorating, wilted and seriously browning and yellowing leaves. The lettuce is not marketable.

Table 4.3: 5-point Hedonic Scale of Vegetable's Appearance Qualities.

5-point Hedonic Scale	Vegetable's Appearance Qualities
1	Extremely defect
2	Slightly defect
3	Fair
4	Good
5	Excellent

Table 4.4: Relative Weight Loss of Butterhead Lettuces in the Cool Storage Chamber.

Days	Sample 1	Sample 2	Average
1	0.0000	0.0000	0.0000
2	-6.9153	-6.0351	-6.4752
3	-13.1544	-12.1570	-12.6557
4	-17.2165	-16.5721	-16.8943
5	-20.2899	-17.2620	-18.7760
6	-23.6759	-18.0081	-20.8420
7	-28.0145	-23.3482	-25.6814
8	-32.7323	-28.6065	-30.6694
9	-33.8592	-29.1328	-31.4960
10	-37.0249	-32.0149	-34.5199
11	-40.4262	-37.4521	-38.9391
12	-44.0938	-41.4431	-42.7685
13	-45.6664	-42.2863	-43.9763
14	-49.7234	-47.3044	-48.5139
15	-53.2784	-53.1197	-53.1990

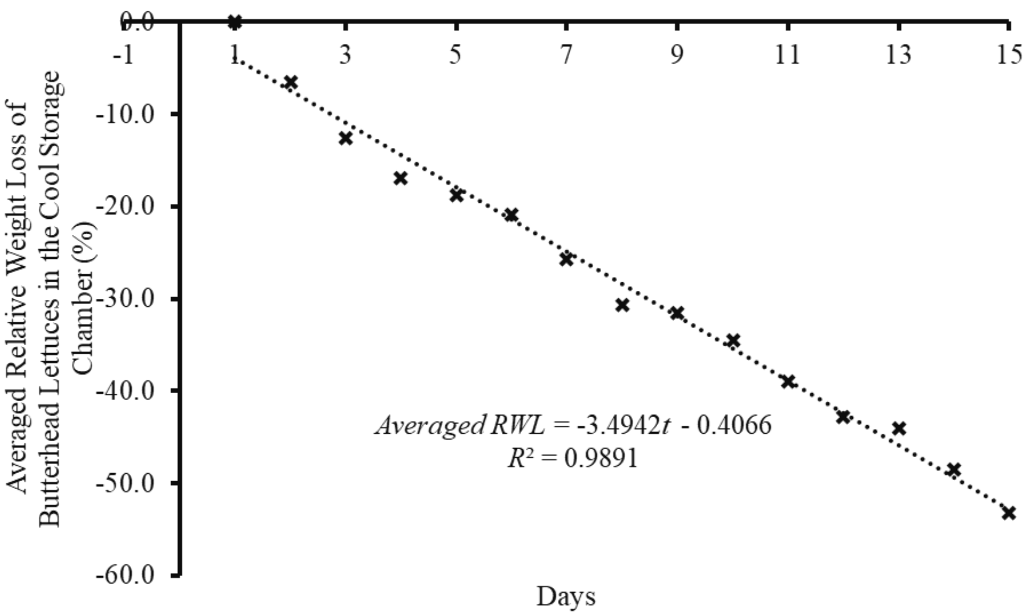


Figure 4.7: Averaged Relative Weight Losses of Butterhead Lettuces in the Cool Storage Chamber.

Table 4.5: Relative Weight Loss of Butterhead Lettuces Stored in the Room Conditions.

Days	Sample 1	Sample 2	Average
1	0.0000	0.0000	0.0000
2	-13.4565	-13.4181	-13.4373
3	-22.7006	-23.1574	-22.9290
4	-30.8698	-32.7491	-31.8094
5	-39.0738	-39.5288	-39.3013
6	-46.8363	-45.4160	-46.1262
7	-54.6569	-51.6949	-53.1759
8	-61.4491	-57.7491	-59.5991
9	-66.8003	-62.2047	-64.5025
10	-71.2916	-65.8513	-68.5715

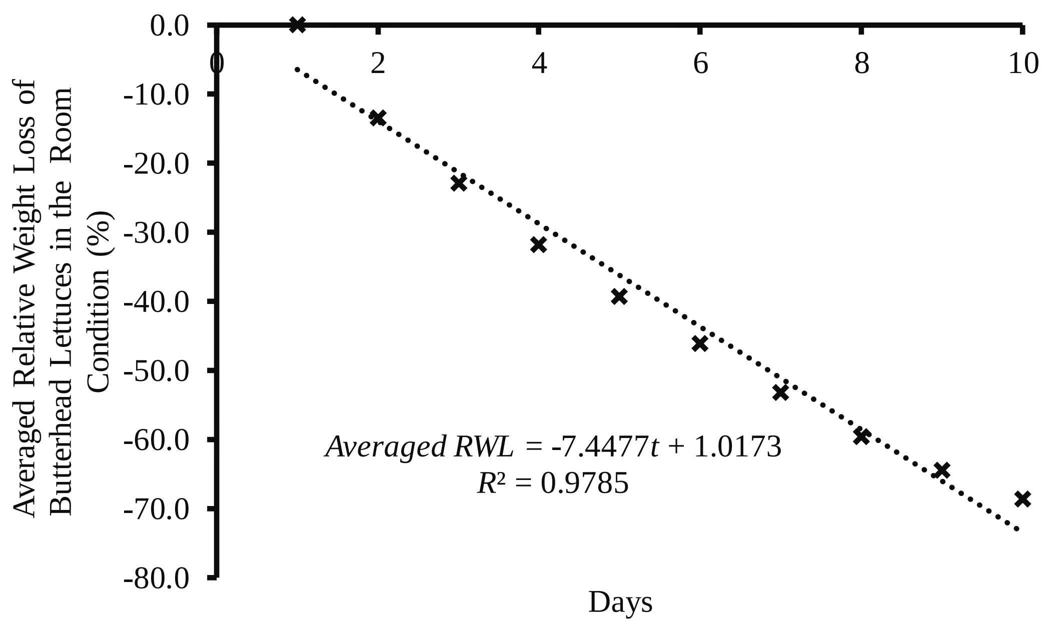


Figure 4.8: Averaged Relative Weight Losses of Butterhead Lettuces Stored in the Room Conditions.

Table 4.6: Appearance Quality of Butterhead Lettuces.

Day	Cold Storage		Room Condition	
	Sample 1	**Sample 2**	**Sample 1**	**Sample 2**
1	5	5	5	5
2	5	5	5	5
3	5	5	4	4
4	5	5	4	4
5	5	5	4	4
6	4	4	3	3
7	4	4	3	3
8	4	4	2	3
9	3	3	2	2
10	3	3	1	1
11	2	2	-	-
12	2	2	-	-
13	2	2	-	-
14	2	2	-	-
15	1	1	-	-

The linear best fit line in Figure 4.7 was found with the equation of $Averaged\ RWL = -3.4942t - 0.4066$, and coefficient of determination of $R^2 = 0.9891$, respectively. Moreover, the finding of averaged RWLs of butterhead lettuces in room temperature were observed in Table 4.5 and Figure 4.8. Under room conditions, the averaged RWL of butterhead lettuces on the tenth day is -69.6480% of the average initial weight. The butterhead lettuces in the room temperature were only in excellent condition for the first two days and gradually deteriorated until they were extremely defective on the tenth day. On the other hand, the appearance quality of butterhead lettuces in the studied chamber and stored under room conditions are presented in Table 4.6. From the table, the results presented that the cold storage could maintain excellent quality with a rating of five for five days and above three (fair) for ten days and the butterhead lettuces were only extremely defective on the fifteenth day. The weight losses of butterhead lettuces in cold storage did not exceed 10g per day except for the first two days. The misting system had decreased the water losses of the butterhead lettuces. Meanwhile, drastic weight losses were observed on the butterhead lettuces under room conditions. Significant differences in weight loss and appearance quality have been observed between the storage of butterhead lettuces in room temperature and cold storage chambers.

4.4 Field Study

A field study was conducted by visiting Ah Peng Orchard located in Teluk Anson. The purpose of this study was to learn more about the method and process of transporting the fresh produce from the farm to other places. According to Mr. Tan Chao Chin, the owner of the orchard, they are utilizing large plastic basket to store the fresh produces. Then, layers of newspaper were used as a bedding in the interior of the basket, to protect the fresh produces inside and to provide some insulation. Figure 4.9 and 4.10 shows the plastic basket used to transport the fresh produces. Other than that, shown in Figure 4.11 polystyrene boxes were also used to store the fresh produces. From here, it is known that farmers are still using the traditional method to store their fresh produce. This traditional storing method may cause damage to fresh produce, in terms of freshness and appearance. This is the due to the polystyrene boxes stacking too high during transportation or storing, causing the boxes in lower levels to collapse, and crushed. Moreover, constantly using newspapers will produce more wastage and harming the environment. So, this project is proposing an innovative way to store and transport the fresh produces.

Figure 4.9: Large Plastic Baskets Used to Store Fresh Produces.

Figure 4.10: Large Plastic Baskets with Newspaper Beddings.

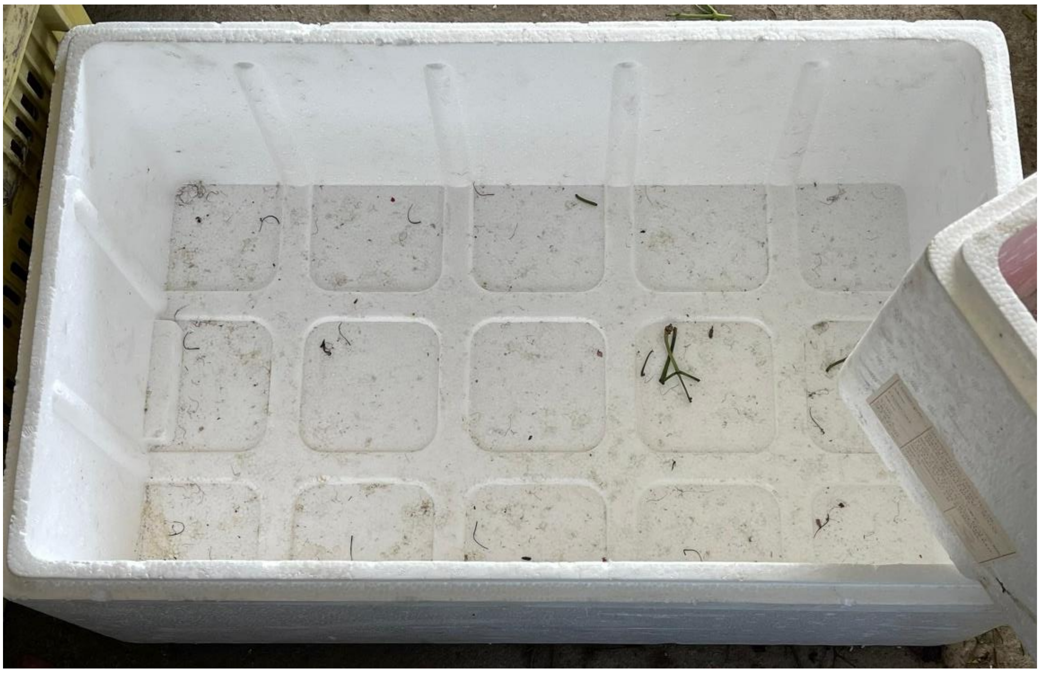


Figure 4.11: Polystyrene Boxes Used to Store Fresh Produces.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This research illustrates how post-harvest procedures particularly the storing stage can be improved by integrating diverse agricultural innovations. As a result, an innovative solution has been proposed to address post-harvest losses, namely the Internet of Things (IoT) cold storage chamber. This sustainable, energy-efficient, user-friendly, and effective solution is suitable for small-scale farmers and domestic logistics purposes. This concept utilizes real-time data collection and monitoring with DHT22 and CCS811 sensors, addressing the limitations of traditional storage methods and presenting a pathway to a more efficient, robust, and sustainable approach to agriculture. Moreover, this real-world application analysis emphasizes the tangible benefits of incorporating IoT into post-harvest storage. Beyond its technological aspects, this solution offers advantages to farmers and plays a crucial role in reducing food waste, aligning with broader goals of efficiency and environmental awareness. Looking forward, the IoT cold storage chamber illustrates the merging of agriculture and technology, ensuring a more reliable and healthy food supply, and signalling a shift towards a future where agriculture harmoniously coexists with our evolving environment. Such forward-thinking ideas are vital in assisting the community to navigate through changing global demands, ensuring the agricultural sector will not only survive but thrives in facing of challenges. Consequently, a cold storage chamber with a temperature and humidity control system has been designed and evaluated. The cold storage chamber successfully extended the freshness of vegetables by maintaining an optimal internal temperature and humidity for vegetable storage. The noticeable differences in weight loss and appearance quality observed in two sets of butterhead lettuce samples demonstrate the effectiveness of the cold storage chamber in preserving vegetable quality, and it could also be applicable to other horticultural crops and fruits.

5.2 Recommendations

The first recommendation is changing the material used for insulation. Currently, the insulation is using conventional Styrofoam (EPS). This is due to pricing and availability concerns. The best material to be used in this case would be the Extruded Polystyrene Foam (XPS). This material has more superior thermal insulating properties compared to conventional Styrofoam. With this material, it is believed that it will contribute a lot in maintaining a lower temperature inside the chamber. Furthermore, the chamber can add in a filter system. By adding a filter system, the excess water drained out from the chamber can be recycled and reused and saving some water in the process. This will also reduce the number of changing of water manually. After that, image processing can be implemented in the system. This can be made possible by adding a camera and some lighting on the top interior of the chamber. With a camera inside the chamber, the system can automatically capture pictures of the fresh produces and determine the condition and appearance quality through image processing. This function can reduce the number of users manually opening the chamber to assess the quality of the fresh produce. Moreover, the MCU of the system can be changed to a more budget friendly board, such as the ESP32. With this change, it will reduce the overall cost of building the system significantly. Through implementing this, the extra conserved budget may be utilized to implement more accurate sensors or maybe purchase higher quality materials in other sections of the build. For the last recommendation is to integrate a portable power storing device for the entire system. This will maximize the mobility of the project as it is able to operate for a certain amount of time on battery. Power storing device such as mobile power banks, 18650 battery packs, or small size 12V lithium batteries can be modified and repurposed to suit the design of this project.

REFERENCES

ADRTECHINDIA (2024). *Cold Storage Temperature Monitoring Systems*. [online] Medium. Available at: https://medium.com/@ADRTECHINDIA/cold-storage-temperature-monitoring-systems-5e9cab35e111 [Accessed 16 Apr. 2024].

Ahmad, M.S. and Siddiqui, M.W. (2015). Factors Affecting Postharvest Quality of Fresh Fruits. *Postharvest Quality Assurance of Fruits*, pp.7–32. https://doi.org/10.1007/978-3-319-21197-8_2.

Blynk.Documentation (2024). *Introduction*. [online] Blynk.io. Available at: https://docs.blynk.io/en.

Cil, A.Y., Abdurahman, D. and Cil, I. (2022). Internet of Things enabled real time cold chain monitoring in a container port. *Journal of Shipping and Trade*, 7(1). https://doi.org/10.1186/s41072-022-00110-z.

Components101 (2018). *DHT22 Sensor Pinout, Specs, Equivalents, Circuit & Datasheet*. [online] Components101. Available at: https://components101.com/sensors/dht22-pinout-specs-datasheet.

Components101 (2021). *5V Four-Channel Relay Module - Pin Diagram, Specifications, Applications, Working*. [online] Components101. Available at: https://components101.com/switches/5v-four-channel-relay-module-pinout-features-applications-working-datasheet.

Darji, C.P. (2021). IoT Based Sensor for Humidity and Temperature Measurement in Smart HVAC Systems. *International Journal of Recent Technology and Engineering (IJRTE)*, 9(5), pp.42–44. https://doi.org/10.35940/ijrte.e5125.019521.

ELPROCUS (2019). *DHT22 – Pin Diagram, Circuit and Its Applications*. [online] ElProCus - Electronic Projects for Engineering Students. Available at: https://www.elprocus.com/dht22-pin-diagram-circuit-and-its-applications/.

Food and Agriculture Organization (FAO) (2009). *How to feed the world in 2050*. [online] Available at: https://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf.

Friha, O., Ferrag, M.A., Shu, L., Maglaras, L. and Wang, X. (2021). Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies. *IEEE/CAA Journal of Automatica Sinica*, 8(4), pp.718–752. https://doi.org/10.1109/jas.2021.1003925.

Gast, K.L.B. (2019). *Storage Conditions: Fruits and Vegetables - Cooperative Extension Publications - University of Maine Cooperative Extension*. [online] Cooperative Extension Publications. Available at: https://extension.umaine.edu/publications/4135e/.

Gikfun (n.d.). *Gikfun DC 6V - 12V Micro Self-priming Diaphragm Pump R385 Suitable for*. [online] Gikfun. Available at: https://gikfun.com/collections/water-pump/products/gikfun-dc-6v-12v-micro-self-priming-diaphragm-pump-r385-suitable-for-water-dispenser-cooler-small-pumping-projects [Accessed 20 Apr. 2024].

Karar, M.E., Al-Rasheed, M.F., Al-Rasheed, A.F. and Reyad, O. (2020). IoT and Neural Network-Based Water Pumping Control System For Smart Irrigation. *Information Sciences Letters*, 9(2), pp.107–112. https://doi.org/10.18576/isl/090207.

Keim, R. (2023). *Understanding the Operation of a Boost Converter*. [online] All About Circuits. Available at: https://www.allaboutcircuits.com/technical-articles/understanding-the-operation-of-a-boost-converter/.

Kinhal, V. (2024). *Carbon Dioxide Technology in Food Preservation: Extending Shelf Life and Ensuring Safety*. [online] Felix Instruments. Available at: https://felixinstruments.com/blog/carbon-dioxide-technology-in-food-preservation-extending-shelf-life-and-ensuring-safety/ [Accessed 22 Apr. 2024].

KYOCERA Corporation (n.d.). *Peltier module (Thermoelectric module)*. [online] KYOCERA GROUP GLOBAL SITE. Available at: https://global.kyocera.com/prdct/ecd/peltier/.

Miller, D., Rembor, K. and Nelson, C. (2017). *Adafruit CCS811 Air Quality Sensor*. [online] Adafruit Learning System. Available at: https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor/overview.

Mohammed, M., Riad, K. and Alqahtani, N. (2022). Design of a Smart IoT-Based Control System for Remotely Managing Cold Storage Facilities. *Sensors*, 22(13), p.4680. https://doi.org/10.3390/s22134680.

Monolithic Power Systems (n.d.). *Understanding AC/DC Power Supplies*. [online] MPS. Available at: https://www.matsusada.com/column/linear_switching_difference.html.

Naik, H.R., Amin, T. and Mahdi, S.S. (2022). Post-harvest Management and Value Addition of Food Crops. *Springer eBooks*, pp.131–146. https://doi.org/10.1007/978-3-031-09218-3_11.

Negi, A. (2024). *Raspberry Pi 3 B+ Pinout with GPIO functions, Schematic and Specs in detail*. [online] eTechnophiles. Available at: https://www.etechnophiles.com/raspberry-pi-3-b-pinout-with-gpio-functions-schematic-and-specs-in-detail/.

ProtoSupplies (n.d.). *Thermoelectric Peltier TEC1-12706 Cooler Kit*. [online] ProtoSupplies. Available at: https://protosupplies.com/product/thermoelectric-peltier-tec1-12706-cooler-kit/ [Accessed 18 Apr. 2024].

Quartz Components (n.d.). *150W 6A DC-DC Step-Up Boost Converter*. [online] QuartzComponents. Available at: https://quartzcomponents.com/products/150w-dc-dc-step-up-boost-converter-10-32v-to-12-35v-6a-adjustable-power-supply-module [Accessed 19 Apr. 2024].

Raspberry Pi Ltd (2023). *Raspberry Pi 3 Model B+*. [online] Available at: https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf [Accessed 20 Apr. 2024].

Sainapse (n.d.). *CCS811 Air Quality Gas Sensor Module For VCO and eCO2*. [online] Sainapse. Available at: https://sainapse.com.my/sensor/ccs811-air-quality-gas-sensor-module-for-vco-and-eco2 [Accessed 18 Apr. 2024].

Sensitech (n.d.). *Relative Humidity: The Key to Fresh Produce?* [online] Sensitech. Available at: https://www.sensitech.com/en/blog/blog-articles/blog-relative-humidity-key-to-fresh-produce/ [Accessed 25 Apr. 2024].

Srivastava, A. and Gulati, A. (2016). iTrack: IoT framework for Smart Food Monitoring System. *International Journal of Computer Applications*, 148(12), pp.1–4. https://doi.org/10.5120/ijca2016911314.

Venkateshwari, P. and Subramaniam, S. (2022). A Survey and Challenges: Embedded System on IoT. *EAI/Springer Innovations in Communication and Computing*, pp.113–129. https://doi.org/10.1007/978-3-031-04524-0_7.

APPENDICES

APPENDIX A: Program Python Code

```
import time
import board
import busio
import adafruit_dht
import adafruit_ccs811
import RPi.GPIO as GPIO
import BlynkLib
from time import sleep
from BlynkTimer import BlynkTimer
import datetime
import signal
from BlynkLib import Blynk
import csv
import openpyxl
from openpyxl import Workbook
import requests
import os
import sys

BLYNK_AUTH_TOKEN = 'CHANGE TO YOUR KEY'

# Initialize Blynk
blynk = BlynkLib.Blynk(BLYNK_AUTH_TOKEN)
```

```
# Create BlynkTimer Instance
timer = BlynkTimer()

current_time = datetime.datetime.now()

# function to sync the data from virtual pins
dhtDevice = adafruit_dht.DHT22(board.D4, use_pulseio=False)
i2c = busio.I2C(board.SCL, board.SDA)
ccs = adafruit_ccs811.CCS811(i2c)

RELAY_GPIO = [17, 27, 22, 24]
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
for relay_pin in RELAY_GPIO:
        GPIO.setup(relay_pin, GPIO.OUT)

@blynk.on("connected")
def connected():
        print("Connected to Blynk2.0")
        print("......................................................")

while not ccs.data_ready:
    pass

def myData():
        interval = 0
        set_interval = 900
        pump = 0
        routine = 0
        pcount = 0
        set_pcount = 900
        filewrite = 1800
        starter = 0
```

```
while True:
    try:
        temperature_c = dhtDevice.temperature
        humidity = dhtDevice.humidity
        current_time = datetime.datetime.now()


        if filewrite >= 1800:
            filewrite = 0

            #Create a new Excel workbook
            workbook = Workbook()

            # Create a new Excel sheet
            sheet = workbook.active
            excelname = str(current_time)
            excelname = excelname.replace(':','')
            excelname = excelname.replace('.','')
            excel_file_path = "/home/raspberry1/blynk-library-python/excel/data"+excelname+".xls"

            sheet.cell(row=1,column=1,value="Date & Time")
            sheet.cell(row=1,column=2,value="CO2(ppm)")
            sheet.cell(row=1,column=3,value="TVOC(pbb)")

    sheet.cell(row=1,column=4,value="temperature(Celsius)")
            sheet.cell(row=1,column=5,value="Humidity(%)")
            sheet.cell(row=1,column=7,value="FileWrite")
            sheet.cell(row=1,column=8,value="new file opened")
            print("new excel file opened.")
            row =2


        filewrite += 1
```

```
        if starter == 0 and humidity is not None and humidity <50:
            GPIO.output(27,1)
            time.sleep(0.4)
            GPIO.output(27,0)
            starter = 1
            sheet.cell(row=row,column=6,value="water pump activated")

        if temperature_c is not None and temperature_c > 10:
            #If temperature is above 10 degrees, activate the relays
            for relay_pin in [24, 22]:
                GPIO.output(relay_pin, 1)
        else:
            #If temperature is 10 degrees or below, deactivate the relays
            for relay_pin in [24, 22]:
                GPIO.output(relay_pin, 0)

        #Check humidity and cooldown timer for pump control
        #BLYNK_WRITE

        if humidity is not None and humidity < 85 and interval == set_interval:
            #If humidity is below 85 and cooldown interval has passed, trigger the water pump
            GPIO.output(27, 1)
            time.sleep(0.4)
            GPIO.output(27, 0)
            interval = 0
            pump = 1
            sheet.cell(row=row,column=6,value="water pump activated")

        if routine == 3600:
```

```
                GPIO.output(27,1)
                time.sleep(0.4)
                GPIO.output(27,0)
                interval = 0
                pump = 1
                routine = 0
                sheet.cell(row=row,column=6,value="water pump activated")

            if pump == 1 and pcount == 900:
                GPIO.output(17,1)
                time.sleep(0.5)
                GPIO.output(17,0)

                pump = 0

            if interval != set_interval:
                interval = interval + 1
            else:
                interval = set_interval

            if pcount != set_pcount:
                pcount = pcount + 1
            else:
                pcount = set_pcount

            routine = routine + 1

            if ccs.data_ready:
                temp = temperature_c
                print("Date & Time: {},CO2: {} ppm, TVOC: {} ppb, Temp: {} C, Humidity: {} %".format(current_time, ccs.eco2, ccs.tvoc, temp, humidity))
                print("filewrite=",filewrite)
```

```
blynk.virtual_write(0, humidity)
blynk.virtual_write(1, temp)
blynk.virtual_write(2, ccs.eco2)
data = [current_time,ccs.eco2,ccs.tvoc,temp,humidity]

# Prepare the data to send to ThingSpeak
#thingspeak_data = {
#'field1': temperature_c,  # Assuming temperature_c is in Celsius
#'field2': humidity,
#'field3': ccs.eco2,}

# Send data to ThingSpeak
#response = requests.post(thingspeak_url, data=thingspeak_data)

# Check if the request was successful (HTTP status code 200)
#if response.status_code == 200:
    #print('Data sent to ThingSpeak successfully!')
#else:
    #print(f'Error sending data to ThingSpeak. Status code: {response.status_code}')
#thingspeak causes extra problem to the coding
# Write data to the Excel sheet
current_time = datetime.datetime.now()
sheet.cell(row=row,column=1,value=current_time)
sheet.cell(row=row,column=2,value=ccs.eco2)
sheet.cell(row=row,column=3,value=ccs.tvoc)
sheet.cell(row=row,column=4,value=temp)
sheet.cell(row=row,column=5,value=humidity)
sheet.cell(row=row,column=7,value=filewrite)
row+=1
```

```
            # Save the workbook to an Excel file
            workbook.save(excel_file_path)

            print("Values sent to Blynk Server!")
            time.sleep(2)
        #else:
        #   print("Error: Data not ready")


    except RuntimeError as e:
        print("Error reading DHT22:", e,"filewrite=",filewrite)
        if interval != set_interval:
            interval = interval + 1
        else:
            interval = set_interval

        filewrite+=1

        if pcount != set_pcount:
            pcount = pcount + 1
        else:
            pcount = set_pcount

        routine = routine + 1
        # Write data to the Excel sheet

        current_time = datetime.datetime.now()
        sheet.cell(row=row,column=1,value=current_time)
        sheet.cell(row=row,column=2,value="DHT22 Error")
        sheet.cell(row=row,column=3,value="-")
        sheet.cell(row=row,column=4,value="-")
        sheet.cell(row=row,column=5,value="-")
        sheet.cell(row=row,column=7,value=filewrite)
        row+=1
```

```
            # Save the workbook to an Excel file
            workbook.save(excel_file_path)

            time.sleep(2)

        except OSError as e:
            print("OSError:", e)
            #restart = 0
            time.sleep(2)
            print("restarting the loop")
            GPIO.output(17,0)
            GPIO.output(27,0)
            GPIO.output(24,1)
            GPIO.output(22,1)
            #while restart < 100:
            #       print("Restarting the loop...")
            #       time.sleep(2)
            #       restart += 1
            #os.execl("/home/raspberry1/blynk-library-
python/combined.py","/home/raspberry1/blynk-library-
python/combined.py",*sys.argv)
            #if restart == 100:
            #       exit(0)


        except KeyboardInterrupt:
            GPIO.output(17,0)
            GPIO.output(27,0)
            GPIO.output(24,0)
            GPIO.output(22,0)
            GPIO.cleanup()
            sheet.cell(row=row,column=1,value=current_time)
            sheet.cell(row=row,column=2,value="keyboard interrupt")
```

```
                    sheet.cell(row=row,column=3,value="-")
                    sheet.cell(row=row,column=4,value="-")
                    sheet.cell(row=row,column=5,value="-")
                    sheet.cell(row=row,column=7,value=filewrite)
                    row+=1

                    # Save the workbook to an Excel file
                    workbook.save(excel_file_path)
                    exit(0)

timer.set_interval(2, myData)

while True:
    blynk.run()
    timer.run()
```