



Master Thesis

The controller for the temperature chamber with Peltier cell

Study programme:

N0714A150003 Mechatronics

Author:

Pavan Lakkamaneni Madhusudhana Rao

Supervisor:

Ing. Miroslav Novak, Ph.D.

Institute of Mechatronics and Computer Engineering

Liberec 2023



Master Thesis Assignment Form

The controller for the temperature chamber with Peltier cell

<i>Name and surname:</i>	Pavan Lakkamaneni Madhusudhana Rao
<i>Identification number:</i>	{Identification number}
<i>Study programme:</i>	N0714A150003 Mechatronics
<i>Assigning department:</i>	Institute of Mechatronics and Computer Engineering
<i>Academic year:</i>	2023/2023

Rules of Elaboration:

1. Do a literature search on Peltier cell control, focusing on the hardware used in the open source projects and temperature chamber control algorithms to maintain a constant set temperature.
2. Design the schematics of the temperature controller. We assume control by a microcontroller or a development board, e.g., Arduino, STM32... The regulator must contain a power circuit to drive the Peltier cell, temperature sensors, a user interface, a data logger, and a communication interface.
3. Design and implement printed circuit boards. Verify the thermal design of power components. Measure the basic parameters: power circuit switching and the precision of temperature measurement.
4. Program controller firmware containing a temperature controller, data logger, and communication with a PC.
5. When programming the temperature controller, it is necessary to select an appropriate controller structure and tune its parameters. The temperature control must feature an appropriate response to temperature setpoint changes (no overshoot) and low sensitivity to ambient temperature changes (i.e., good disturbance rejection). You may consider using the setpoint weighting technique for this purpose. Analyze whether the Peltier cell nonlinearity requires the use of the gain scheduling technique or at least the switching between heating/cooling sets of controller parameters. Check the functionality of the controller on the finished temperature chamber.

Scope of Graphic Work: by appropriate documentation
Scope of Report: 40–50 pages
Thesis Form: printed/electronic
Thesis Language: English

List of Specialised Literature:

- [1] Anon., 2010. MAX1978/MAX1979 Integrated Temperature Controllers for Peltier Modules [online]. 19?2490; Rev 3; 3/10. B.m.: Maxim Integrated. Available from: <https://datasheets.maximintegrated.com/en/ds/MAX1978-MAX1979.pdf>
- [2] PANG, Du-Yeol, Won-Suk JEON, Kwang-Hoon CHOI, Tae-Kyu KWON, Nam-Gyun KIM a Seong-Cheol LEE, [b.r.]. Temperature Control using Peltier Element by PWM Method. 5.
- [3] SCHILLER, Bobby, 2020. Peltier Cooling for Test Modules [online]. C++. 23. květen 2020. [vid. 2022-09-27]. Available from: <https://github.com/bobby-schiller/peltier-control>
- [4] Karl J. Aström and Tore Hägglund, PID Controllers: Theory, Design, and Tuning, Second Edition, International Society of Automation 1995, ISBN: 978-1-55617-516-9

Thesis Supervisor: Ing. Miroslav Novak, Ph.D.
Institute of Mechatronics and Computer Engineering

Date of Thesis Assignment: January 26, 2023
Date of Thesis Submission: May 22, 2023

Prof. Ing. Zdeněk Plíva, Ph.D.
Dean

S.S.

Doc. Ing. Josef Černohorský, Ph.D
Head of institute

Liberec January 26, 2023

Declaration

I hereby certify, I, myself, have written my master thesis as an original and primary work using the literature listed below and consulting it with my thesis supervisor and my thesis counsellor.

I acknowledge that my master thesis is fully governed by Act No. 121/2000 Coll., the Copyright Act, in particular Article 60 – School Work.

I acknowledge that the Technical University of Liberec does not infringe my copyrights by using my master thesis for internal purposes of the Technical University of Liberec.

I am aware of my obligation to inform the Technical University of Liberec on having used or granted license to use the results of my master thesis; in such a case the Technical University of Liberec may require reimbursement of the costs incurred for creating the result up to their actual amount.

At the same time, I honestly declare that the text of the printed version of my master thesis is identical with the text of the electronic version uploaded into the IS STAG.

I acknowledge that the Technical University of Liberec will make my master thesis public in accordance with paragraph 47b of Act No. 111/1998 Coll., on Higher Education Institutions and on Amendment to Other Acts (the Higher Education Act), as amended.

I am aware of the consequences which may under the Higher Education Act result from a breach of this declaration.

August 28, 2023

Pavan Lakkamaneni Madhusudhana Rao

The controller for the temperature chamber with Peltier cell

Abstract

The work describes one of the ways of controlling the solid-state heat or cold generator called as Peltier module by designing a dual H-Bridge circuit and controlling the electric properties for the Peltier module by signalling suitable MOSFET driver with the help of readily available microcontroller to fulfil the generated temperatures as per the expectations inside the thermally insulated box. Firstly, the requirements of the cooling and heating capacities to be achieved is decided and the taken into considerations to choose the proper electrical hardware and constructing a PCB for integrating the components into it, following with a suitable circuit configuration for changing the electric parameters to the Peltier module for applying heat or removing heat on same side to maintain the prescribed temperature by the user. The change in electric parameters is decided by the suitable readily available commercially microcontroller by generating PWM signals to the respective digital pins for the MOSFET drivers. The digital temperature sensors DS18B20+ are used to get feed-back on the generated temperatures from the enclosed box. The suitable user interface is been de-signed for setting up the required temperature inside the box and get data on the display. While the main parameters generated temperatures and set temperatures are transferred to the Arduino cloud for data logging and also to manipulate the required temperatures remotely. In the present work the Peltier modules are used to apply heat or remove heat in the same thermally insulated box by controlling the direction and the flow of the current as per the requirements.

Keywords

Peltier effect, Seebeck effect, Thermoelectric, Controller, MOSFET, H-Bridge, Driver, PCB, DC/DC Convertor, Sensor, UART, PID, Algorithm.

Acknowledgements

To all my technical guides and motivators, I would like to share my deepest gratitude for supporting me to complete the thesis.

Firstly, I would like to heartily thank my supervisor, **Ing. Miroslav Novak, Ph.D.**, for supporting and guiding me throughout the process of fine-tuning the approaches to complete this thesis. His continuous dedication to sharing his extensive knowledge of fabricating the project has been invaluable. His insights and constructive feedback have been instrumental in shaping the outcome of this work, and his mentorship has truly been a cornerstone of my academic journey.

I am deeply appreciative of the opportunity to express my gratitude to doc. **Dr. M.Sc. Ing. Jaroslav Hlava**. His exceptional guidance and expertise have been pivotal in my academic journey. The dedication to teaching has allowed me to delve into the intricate realms of Real-Time Operating Systems (RTOS) and control systems. His insightful approach, paired with his practical wisdom, has not only expanded my understanding but also equipped me with valuable skills that will undoubtedly shape my future endeavours.

Further, I would like to extend my heartfelt gratitude to **Ing. Lukas Hubka, Ph.D.**, for his invaluable role in helping me to initialize the thought of final outcomes and expectations with clear understandings. His expertise, insightful discussions, and guidance have played a pivotal role in shaping the direction and depth of this research.

In closing, I want to express my deep appreciation to the **Technical University of Liberec (TUL)**, specifically the Faculty of Mechatronics, Informatics, and Interdisciplinary Studies, for their unwavering support. Their provision of vital resources was fundamental in bringing my vision to life through a functional prototype. This hands-on experience not only enriched my understanding of practical problem-solving approaches but also enabled me to grasp the intricacies of turning theoretical knowledge into tangible solutions. TUL's backing has been a fundamental pillar, enabling me to merge academic learning with real-world application successfully.

Contents

- Motivation And Scope..... 12**
- 1 Introduction 12**
 - 1.1 Overview of Work 12
 - 1.2 State of Art..... 13
- 2 Brief Overview of Peltier Module 15**
 - 2.1 Thermoelectric Materials:..... 15
 - 2.2 Thermoelectric Effect 16
 - 2.2.1 Seebeck Effect 16
 - 2.2.2 Peltier Effect 18
 - 2.2.3 Thomson Effect..... 19
 - 2.3 Working Principle of Peltier Module..... 20
 - 2.4 Overview of Peltier Module 21
 - 2.5 Choosing of Peltier Module..... 23
- Thermoelectric Controller Review 24**
 - 2.6 Classification of Controllers..... 24
 - 2.7 Hardware Overview on available controllers..... 26
 - 2.7.1 LT8722 29
 - 2.7.2 MAX1978/MAX1979..... 30
 - 2.7.3 TC-48-20..... 33
 - 2.7.4 TEC-8A-24V-PID-HC-RS232 34
 - 2.8 General Assumptions on Thermoelectric controllers..... 36
- 3 Ideology and approaches for developing and fabricating the Thermoelectric Controller. 37**
 - 3.1 Design of power circuit. 38
 - 3.2 Schematics of the power circuit..... 39
 - 3.3 Interfacing components 42
 - 3.3.1 Interfacing ESP32 with DS18B20+ 42
 - 3.3.2 Interfacing Arduino Uno and ESP32 44
 - 3.3.3 User Interface..... 48
- 4 Control Algorithm 54**
 - 4.1 PWM signal design..... 56
 - 4.2 Algorithm behaviour..... 58
 - 4.3 Test Setup..... 63
 - 4.4 PI Tuning 64

5	Results and conclusion.....	68
5.1	Conclusion	72
6	Citations	73
7	List of Supplement Materials.....	78

List of images

Image 1: Seebeck Effect.....	17
Image 2: Peltier Effect.....	18
Image 3: Thomson Effect.....	19
Image 4: Working of Peltier Module	20
Image 5: Construction of Peltier Module.....	22
Image 6: LT8722 (a) Schematics, (b) Demonstration Circuit Front Side, (c) Demonstration Circuit back side.....	29
Image 7: MAX1978/MAX1979 schematics	30
Image 8: (a) DAC Input User Interface, (b) Monitoring Interface.....	32
Image 9: (a) MP-3193, (b) OEM Design, (c) Hardware.....	33
Image 10: TEC -8A-24V-PID-HC-RS232 (a) Front Side, (b) Back Side.....	34
Image 11: TEC -8A-24V-PID-HC-RS232 Connections.....	35
Image 12: System Schematics.....	37
Image 13: Schematic layout of Power circuit.....	39
Image 14: PCB (a) Top Layer, (b) Bottom Layer.....	41
Image 15: PCB board (a) Front side (b) Back side	41
Image 16: DS18B20+ (a) Pinout, (b) Normal Mode Connection.....	42
Image 17: Output readings from Temperature sensor.....	43
Image 18: UART wire connection schematics.....	45
Image 19: Output results of UART communication	48
Image 20: User interface design.....	48
Image 21: Output LCD display.....	54
Image 22: Control connections.....	55
Image 23: PWM Signals	57
Image 24: Control Flowchart.....	59
Image 25: (a) 3D Printed Box, (b) Integrated Peltier module and Heatsink.....	63
Image 26: (a) Test Setup, (b) Heat Sink.....	64
Image 27: (a) Sensor integration for testing, (b) Hardware Integration.....	64
Image 28: Thermal camera images (a) External side heatsink, (b) Internal side heatsink, (c) Power circuit & (d) Internal side heatsink.....	72

List of graphs

Graph 1: Setpoint temperature(t_{set}), PWM, Generated Temperature(T_1) VS time.....	66
Graph 2: Setpoint temperature(t_{set}), PWM, Generated Temperature(T_1) VS time.....	67
Graph 3: Error in regulation.....	67
Graph 4: Temperature VS PWM (a) -98% to 98%, (b) 98% to -98%.....	68
Graph 5: T_1 , ($T_1 - T_3$) VS PWM (a) -98% to 98% duty cycle, (b) 98% to -98% duty cycle. ...	69
Graph 6: Function fit.....	70
Graph 7: Dynamic behaviour of the box.....	70

List of tables

Table 1: Signalling pin layout 56

List of sourc

Source Code 1: DS18B20+ interface with ESP32..... 43
Source Code 2: UART code for Arduino..... 46
Source Code 3: UART code for ESP32. 47
Source Code 4: Code for Display UI with UART communication..... 53
Source Code 5: Control Algorithm code..... 62
Source Code 6: Multicore processing code..... 63

List of abbreviations

Abbreviation	Description
TEC	Thermoelectric Coolers
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
DC	Direct Current
ΔV	Potential Difference
EMF	Electromotive Force
Bi_2Te_3	Bismuth Telluride
PbTe	Lead Telluride
SiGe	Silicon Germanium
CoSb_3	Skutterudite
KE	Kinetic Energy(J)
PE	Potential Energy(J)

Q	Heat Transfer(J)
T_h	Temperature at hot end($^{\circ}\text{C}$)
T_c	Temperature at cold end($^{\circ}\text{C}$)
S	Seebeck Coefficient($\mu\text{V}/\text{K}$)
I	Current(A)
π_{AB}	Peltier Coefficient (Joules/Coulomb)
E^l	Effective Field
σ	Thomson Coefficient
V	Voltage
P	Power(W)
QL	Heat rejected at low temperature point(J)
QH	Heat pumped at high temperature point(J)
COP	Coefficient of Performance
K_t	Thermal conductivity ($\text{W}^{\circ}\text{C}^{-1}$)
R	Total resistance(ohm)
Z	Figure of merit
λ	Thermal Conductivity ($\text{Wm}^{-1}\text{K}^{-1}$)
I_{max}	Maximum operational Current(A)
Q_{max}	Maximum heat generated(J)
V_{max}	Maximum operational Voltage(V)
ΔT	Temperature difference($^{\circ}\text{C}$)

Motivation And Scope

The manner vaccines are kept is crucial to ensuring their efficacy. In order for immunizations to continue to function as intended in preventing disease, their efficacy and safety must be maintained through appropriate storage. The vaccinations need to be kept at the right temperature in order to prevent deterioration. Most vaccinations must be kept between 2 °C and 8 °C during storage. Exposure to extreme temperatures can decrease the efficiency of vaccinations, limit their efficacy, or leave them useless. While this applies to many more medicines, for example insulin would not be effective if it's not stored in between 2-8 °C (36-46 °F) to ensure it's not been degraded [1]. While this approach can also be applied in many other applications further like, organ storing till it's been implanted to the receiver and also in food tech or food delivery industries for maintaining optimum temperatures till it's been consumed by the consumers. This project also helps further to maintain temperatures required for long transportations until powered on.

1 Introduction

The modern discovery of thermoelectric materials which exhibits the phenomenon of generating heat or degenerating heat from the surroundings while applying voltage across it which is taking lead in the industry of refrigeration and heating. This device is the main to get inspiration for using in this thesis for applying heat or reducing heat in the closed thermally insulated chamber, which helps to maintain the environment around the things (medicines) as per the prescribed temperature for a long time to reduce losing its ideal form. Controlling the Peltier module is the main motivation to achieve temperatures in the range -5 to 45 °C.

1.1 Overview of Work

The course of the thesis describes the development of thermoelectric controller by choosing suitable Peltier Module, with a circuit schematic suitable for working in the required voltages and current continuously along with a feedback control by using digital sensors for maintaining the required temperatures as prescribed for the things.

Firstly, the hardware literature review for the present readily available thermoelectric controllers is studied and compared the applications for understanding the parameters to be choose for driving the Peltier module for the applications i.e., mainly focusing for controlling the current and voltages that is applied for the Peltier module.

Further, the circuit is designed to drive the current bidirectional to the Peltier module and designed to with suitable components supporting to control the Peltier module. The required active area to be impacted with change in temperature is considered to find the total power to be achieved to add heat or absorb heat inside the enclosed box. Further by choosing the suitable readily available microcontroller to apply the control logic for achieving the desired temperature and maintaining it. The suitable components like MOSFETS, MOSFET drivers and connectors with suitable DC/DC convertors are picked by the commercially available websites.

After plotting the circuit schematics and analysing, the PCB is designed by placing the respective components in place efficiently to integrate inside the components slot in the enclosed box. As suitable for controlling the Peltier module indirectly by MOSFET drivers, ESP32 is selected and which further used to upload the important parameters to the cloud for data logging.

Lastly, the firmware is developed using Arduino platform for ESP32 based development board for successfully obtaining the user interface for the user to set the temperature, singling the MOSFET drivers for maintaining the input temperature by PWM and data logging to trace the temperatures by uploading to the Arduino cloud where interface from user is also possible remotely.

1.2 State of Art

Peltier modules are also called as thermoelectric modules which works on the phenomenon Peltier effect to transfer heat from one side of the module to another. Now a days these modules are used for heating or cooling applications irrespective of size of material to be maintained under temperature. There are many fields getting benefited by these modules and some of are electronics, medical and industrial processes.

Modern Peltier module design integrates advancements in material science, manufacturing techniques, and design. Some significant improvements include:

- High- performance thermoelectric materials – New thermoelectric materials are being developed for increase in the performance means generating large amount of temperature difference across the sides of the module in the given electric power [2]. Some examples of materials promising for performances are skutterudites, half-Heuslers [3], and topological insulators.
- Improved device design – Increasing of computational modelling and simulating caused to redesign and optimize the geometry and configure the Peltier module for

precises applications, some designs include of design of the conducting material in the junction of semiconductor which does not make both type semiconductor be in contact on the ceramic plate[4]. There are many circular [5] and flexible thermoelectric module [6] available for usages.

- Integration with other cooling techniques - For high heating elements which cannot be cooled using cooling liquids, the high cooling Peltier modules are used where to cool Peltier modules on other side liquid heat exchangers are designed accordingly [7].
- Miniaturization of Peltier Modules – Miniaturizing of Peltier modules are the present emerging area of research, with huge potential in performance of portable electronic devices like processors, medical implants and other small-scale systems. The researchers are developing new fabrication techniques and materials which can be utilized in manufacturing of miniature size of modules for specific applications [8].
- Advanced control algorithms- There are many researchers developing the algorithms to control the thermoelectric materials precisely for hard real-time applications in cooling and maintaining the system under temperatures [9].
- Environmental sustainability – Extensive research is undergoing in the development of Thermoelectric components to reduce the impact on environment while manufacturing these modules and also use of biodegradable materials to minimize the footprint of Peltier Modules [10].
- Integration with renewable energy sources – Peltier modules can be integrated with self- sustainable cooling systems where they can be planted in remote areas or off-grid locations where traditional cooling methods are not available or practical [11].

In summarizing, the thermoelectric modules are focused on improving performance, efficiency, reliability and sustainability, along with exploring new applications and integrating with new technologies. The advancements are likely to lead to the development of more efficient, compact and versatile Peltier modules that can meet the growing demand for thermal management solutions in various industries.

Theory section

2 Brief Overview of Peltier Module

2.1 Thermoelectric Materials:

The materials that can induce change in temperature in response to the applied voltage difference are called as Thermoelectric materials. This behaviour in these materials is called as Seebeck Effect which explains the connection between the flow of heat and the flow of electrons internally to the materials. The industry manufactured Thermoelectric modules can be directly used in power production of inducing heat or cold as per the requirements. It is observed that the electrons inside these materials would flow from hotter side to the colder side. Inversely, applying an electric voltage to the device, a temperature device can be created, which can be used for cooling or heating application.

By the advancement of doping in semiconductors, it is obtained that the thermoelectric materials can be manufactured by doping n-type or p-type elements for the requirement. These induced thermoelectric materials are considered by parameters depends on electronic band structure and also the thermal conductivity. Materials with higher Seebeck coefficient with low electrical resistance and high thermal resistance are considered as ideal for thermoelectric materials. Some of the thermoelectric materials used are as follows.

- Bismuth telluride (Bi_2Te_3)

Bismuth telluride is one of the most used thermoelectric material due to its high Seebeck coefficient and with low thermal conductivity, which leads to be ideal material for building thermoelectric modules for commercial purposes[12]. Undoped Bi_2Te_3 generally has a Seebeck coefficient of roughly $-200 \mu\text{V}/\text{K}$ at ambient temperature, meaning that a temperature difference of 1 K across the material can result in a voltage differential of $-200 \mu\text{V}$ [13].

- Lead telluride (PbTe):

Lead telluride has pretty high Seebeck coefficient which gives advantages to use for high temperature generation applications [14].

- Silicon germanium (SiGe):

Silicon germanium is recently explored in the field of thermoelectric materials, it has promising high Seebeck coefficient and low thermal conductivity which makes it potentially more efficient than other materials. The Seebeck coefficient of undoped SiGe is approximately -180 V/K at ambient temperature. However, by doping the material with impurities like boron (B) or phosphorus (P), the Seebeck coefficient may be raised. Doping can improve a material's electronic band structure and carrier concentration, resulting in improved thermoelectric performance [15, 16].

- Skutterudite (CoSb₃):

CoSb₃ is a well-known thermoelectric material that belongs to the family of skutterudites. This material has good thermoelectric property but also has high thermal conductivity which is not good to choose for the manufacturing a thermoelectric module, while there are other factors used to tune these properties by doping [17].

2.2 Thermoelectric Effect

Thermoelectric effect is the phenomenon of thermoelectric materials which gives voltage difference across the terminals when temperature gradient is applied across the sides of it and also when applied voltage difference across the sides, when temperature gradient is obtained across the sides. This effect applied to many fields to generate electricity, generate temperature differences or cool or heat the objects and also as sensors in finding the temperatures or current. This effect can make thermoelectric modules use as temperature controllers. The Seebeck effect, Peltier effect, and Thomson effect are three independently recognized effects that are together referred to as the thermoelectric effect.

2.2.1 Seebeck Effect

In the year 1821, Thomas Seebeck, a German physicist first time witnessed that when two dissimilar metals in wire form were connected to form a complete electric circuit and applied temperature difference across the junction of joints, which led to produce voltage difference. Where the circuit is called as thermocouple. This phenomenon is due to conversion of thermal energy to electric energy [18].

The voltage produced by Seebeck effect is very less and would be in microvolt per kelvin of temperature difference at the junction. Numerous of such materials can be connected to increase

the voltage difference across the junctions of the dissimilar metals. Usually, the electrons flow from hot side to cold side, hence its one directional an dc is flown. Further, electrons on a warmer side give high average momentum as compared to the cold side which leads to negative charge in cold side compared to hot side.

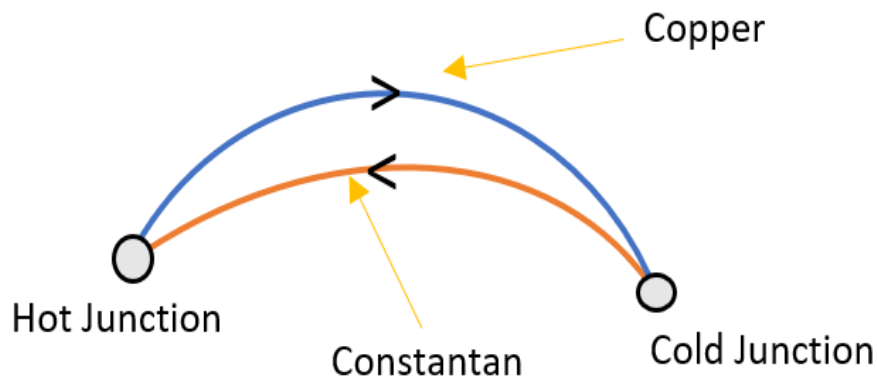


Image 1: Seebeck Effect

The above Image 1 explains one of the examples where two dissimilar metals copper (Cu) and Constantan are connected to form two junction to form a thermocouple, when these junctions are experienced temperature gradient, it is found there is flow of electrons from hot junction through copper to cold junction and from cold junction through constantan to hot junction. It is observed that the valence electrons at hot junction of the metals are responsible behind the thermal energy. The kinetic energy absorbed by the valence electrons at hot side are majorly subjected for migration from hot side to cold side than the electrons from cold side. As the electrons tries to attain saturation, moves from higher state fermi energy to lower state fermi energy, while the flow of electrons in the loop is observed. While it is observed that the potential difference was in microvolts and is directly proportional to the change in temperature in the junctions. [18, 19].

$$\Delta V = S(Th - Tc) \quad (2)$$

ΔV - Potential Difference (in microvolts),

Th - Temperature at hot end (in kelvin),

Tc - Temperature at cold end (in kelvin),

S - Seebeck Coefficient ($\mu V/K$).

2.2.2 Peltier Effect

In the year 1834, Jean Charles Athanase Peltier, a French watch maker discovered a thermoelectric effect. If a current is flown in the circuit composed of connecting dissimilar materials, the temperature difference is seen at the junctions. The flow of electrons at the junctions is same as Seebeck current, the heat is released at hot junction and absorbed at cold junction. It is observed that the total amount of heat(Q) absorbed or rejected to the surroundings from the junction is proportional to the current(I) passing through the junction.

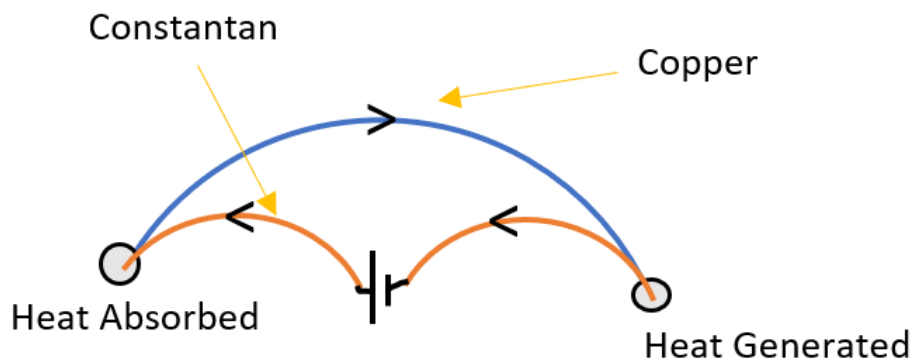


Image 2:Peltier Effect

The above Image 2 illustrates an example for Peltier effect observed with the same dissimilar material referred to Peltier effect. When a potential difference is introduced in one of the metals in a loop, one of the junctions would absorb heat and reduces the temperature compared to surroundings and the other junction would generate heat and increases the temperature to the surroundings. When the charge is applied to the thermocouple, the valence electrons would absorb energy at one side and attains higher fermi levels and flow to the other junction to release the energy and attain lower fermi energy levels, which cause the junctions to generate or absorb the energy. The direction of charge flow causes the particular junction to absorb or release the Energy. It is observed that the heat generated or absorbed by the thermocouple is directly proportional to the charge in the loop and also depends on the metals selected to form a thermocouple. Hence Peltier coefficient would change for different metals.

$$Q = \pi I \quad (2)$$

Q - Heat Liberated (in joules),

I - Current passed through the loop (in coulomb),

π -Peltier Coefficient

2.2.3 Thomson Effect

In the year 1854, British physicist, William Thomson (Lord Kelvin) discovered the thermoelectric effect where the link between Seebeck and Peltier effect is seen. He discovered that when a current is passed in the homogenous conductor and subjected to temperature gradient, then heat must be exchanged to the surroundings of the conductor. Where in the homogenous metal if there is a thermal gradient across the ends, there it is sure to also be having potential gradient in the metal. It can also be found that the junction potential gradient is altered if there exists any temperature gradient on homogenous metal. Thomson effect can be further understood by referring the below Image 3. The metal conductor has free charge carriers in the form of electrons, when the part of the metal is experiences external heat, then the electrons under that area would be more energetic compared to cold part of the metals, where the heat energy is absorbs and kinetic energy is increased for the electrons, which causes to move to the cold place to attain saturation. By this flow of electrons is observed through the metal this affects when the metal is used as a conductor for electricity, where witness to slightly change of potential difference across it.

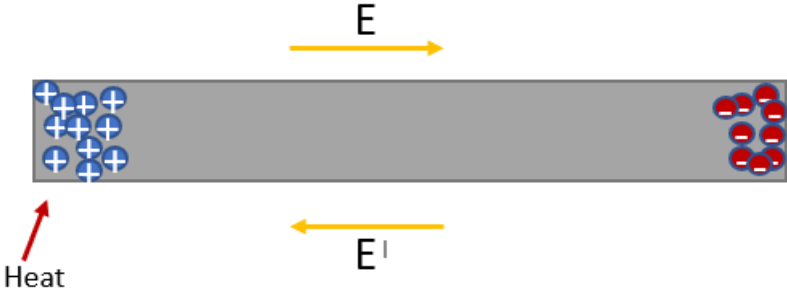


Image 3: Thomson Effect

$$E' = \sigma \frac{dT}{dx} \tag{3}$$

E' - Effective field is proportional to thermal gradient.

σ - Thomson Coefficient.

$\frac{dT}{dx}$ - Change in temperature.

It is observed that, Thompson effect would increase the potential difference than the applied, is when metals like copper and zinc are used. The hot end of this metals is at higher potential, which lets to flow of electrons responsible due to thermal gradient on the same direction of the external potential source. This effect is called as positive Thomson effect. If the conducting

metals has lower potential and simultaneously has heat source at the same side then the electrons responsible for flowing due to temperature gradient would be opposite to the external source potential, which causes negative Thomson effect.

2.3 Working Principle of Peltier Module.

Primary, Seebeck effect was discovered first in 1821 by Thomas Seebeck, where the flow of current is observed when two different metals are connected and both ends are applied with different temperatures. Then in 1834, Jean Charles Athanase Peltier exhibits the Peltier effect, where the flow of current across the different materials led to the change in temperature across the junctions, which led to the development of Thermoelectric modules. Metal wires of bismuth and copper joined to form an electric circuit would exhibit the change in temperature in leads. The temperature is increased across the junction when electricity is flown from bismuth to copper, while the temperature is reduced on other junctions when the electrons are flown from copper to bismuth. Usually, nowadays the semiconductors with thermoelectric properties are chosen and doped to form n-type and p-type semiconductors. When these doped thermoelectric materials are formed an electric circuit and then applied current, the heat is generated when the electrons are flown from n-type to p-type, i.e., where the electron from higher energy from n-type is jumped to the lower energy level of p-type. While the heat is absorbed when the electron is flown from p-type to n-type to fill the energy gap between higher level to lower level. This results in a change in temperatures on ceramic plates [18–20].

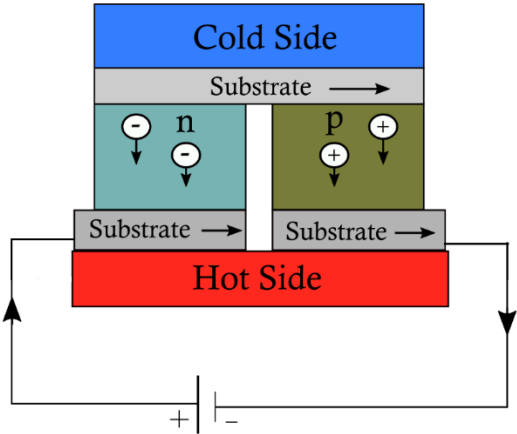


Image 4: Working of Peltier Module

The above Image 4 [21] shows the Peltier phenomenon of unit arrangement, as the current flows from the positive terminal to the negative terminal, which means the flow of charge is from the negative to the positive terminal. This led to moving the electron from a higher level to a lower level on the hot side while the electrons by releasing energy in the form of heat on the cold side

the electrons from lower-level jumps to a higher level where the energy is absorbed which absorbs heat. The theoretical parameters of thermoelectric module is determined by the following equations [22].

$$V = \alpha(Th - Tc) + IR \quad (4)$$

$$P = \alpha(Th - Tc) + I^2R \quad (5)$$

$$Ql = \alpha ITc - \frac{1}{2}I^2R - Kt(Th - Tc) \quad (6)$$

$$Qh = P + Ql = \alpha ITh - \frac{1}{2}I^2R - Kt(Th - Tc) \quad (7)$$

$$COP = Ql/P \quad (8)$$

$$Z = \frac{\alpha^2}{KtR} = \left(\frac{\alpha^2\sigma}{\lambda}\right)T \quad (9)$$

R -The total resistance offered by the module (ohm),

Kt -The total thermal conductivity of the module ($W^\circ C^{-1}$),

α -The Seebeck coefficient of the module ($V^\circ C^{-1}$),

Z -Figure of merit,

λ - Thermal conductivity,

α - Electrical Conductivity.

The performance and efficiency of the Peltier module can be assumed by the dimensionless parameter called as the figure of merit denoted as Z .

2.4 Overview of Peltier Module

The commercial Peltier modules consists of a thermoelectric material, which has less thermal conductivity but more Seebeck coefficient. These modules work on Peltier effect and the effect is directly proportional to the flow of the current. The selected material is doped to obtain p-type and n-type of it and made into small rectangular pellets, these pellets are connected in series to alternative p- type and n -type with conducting metal to form a junction. While the two ceramic plates are held by the junctions formed by the conducting metal with industrial glue usually to decrease thermal short circuit across the ceramic plates. The ends of the thermoelectric material are connected with wire for external supply. The below Image 5 [21] shows the arrangement of the thermoelectric metal pellets, where it is observed the material forming junction is in contact with the ceramic plate and the materials are connected in series but parallel

with the ceramic plate. The maximum current that can be supplied would be directly proportional to number of pellets used in the module and its size choose for application.

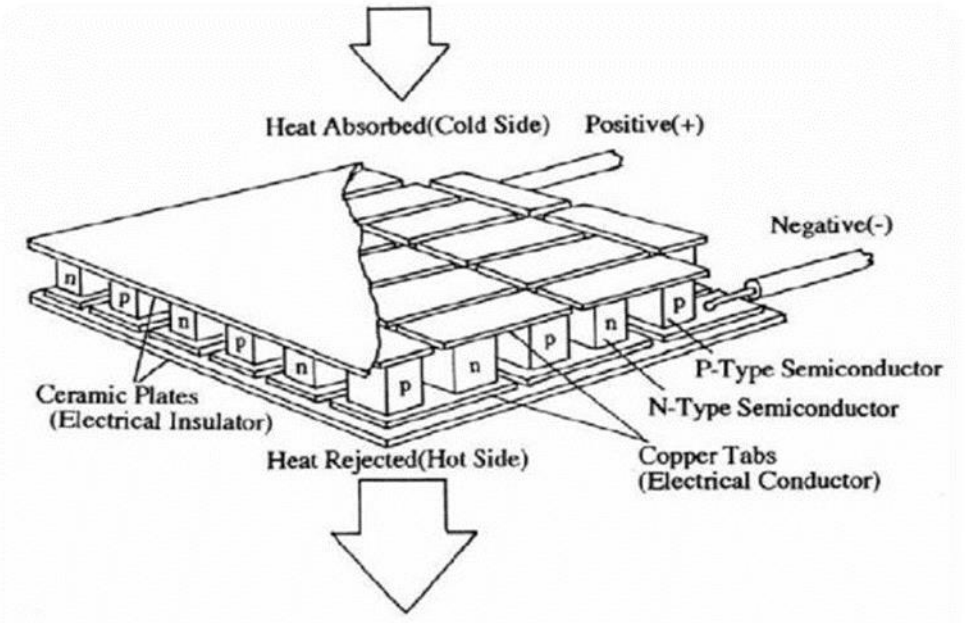


Image 5: Construction of Peltier Module

The Image 5 above represents the construction of the thermoelectric module, the series of n-type and p-type are sand-witched between electric insulators (i.e., ceramic plates) on both sides. While thermally they are parallel across two ceramic plates.

The below Image 6 [21], shows the dismantled Peltier module to expose the arrangement of the thermoelectric material in series electrically and parallel in thermal to the white ceramic plate.

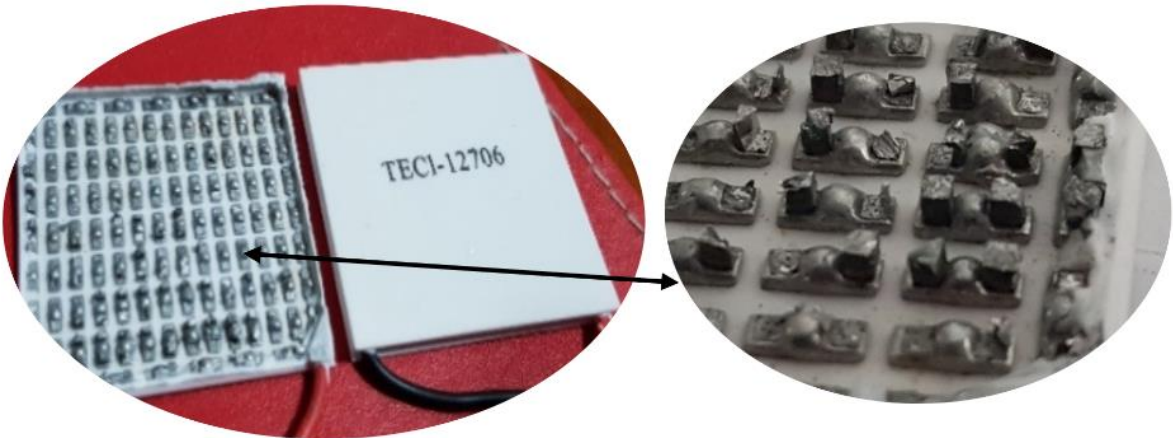


Image 6: Dissection of Peltier Module.

2.5 Choosing of Peltier Module

The datasheet from the manufacturer contains information on the Peltier module's performance specifications. Calculating the variables that should be reflected in the enclosing box is advised before choosing the Peltier module. Then decide on a Peltier module that meets the criteria. The performance curves would be provided by some manufacturers to quantify if the Peltier is suitable for the application. By choosing the working voltage of the circuit and change in temperature across the Peltier module, an operating current can be found by tracing the curve. While knowing the applied energy with the achievable change in temperature, an operating current can be obtained. Usually, some manufacturers give general data limiting the operating parameters of the Peltier module. The parameters usually on datasheet disclosed by the manufacturers are as follows.

- I_{max} (in Amps) - The maximum input amps to achieve maximum change in temperatures across the ceramic faces.
- Q_c (in watts) - The maximum amount of heat that can be absorbed on cold side of the Peltier module.
- Q_{max} (in watts) – The maximum amount of heat that can be generated on hot side of the Peltier module.
- T_h (in °C) – The temperature on the hot side of the Peltier module.
- T_c (in °C) – The temperature on the cold side of the Peltier module.
- ΔT (in °C) – The difference temperature between hot side and cold side of the Peltier module.

$$\Delta T = T_h - T_c \quad (10)$$

- ΔT_{max} (in °C)- The maximum temperature difference that can be achieved by the Peltier module.
- V_{max} (in Volts) - The maximum voltage the Peltier module can tolerate without damaging.
- Module Resistance (Ohms)- The electrical resistance of a thermoelectric module is quantified as module resistance.

Thermoelectric Controller Review

The research in maintaining the temperature under control for various applications for cooling or heating various materials has led to develop controllers. To maintain the specific temperature by applying heat or removing heat, there is no requirement of adding two different systems one responsible for heating and other responsible for cooling which cause the system huge and expensive. Meanwhile using Peltier modules can achieve temperature on both directions. In classic old applications, the system was equipped with rotating shafts to pump coolant for transferring heat from the interested volume. Which led to increase in maintenance and risk of mishandling the hazardous coolants. There are some manufacturers focusing on controlling the thermoelectric modules, the next section would be the review regarding commercially available controllers.

2.6 Classification of Controllers.

The study of the readily available thermoelectric controllers led to know the features offered and choosing the proper controllers as per the application. There are many factors to be considered before finalizing the controllers, some factors are temperature range, cooling, heating capacity, precision, control interface and safety features.

Some of the features available with the thermoelectric controllers which are readily available in different applications are as follows:

- Single-Channel Temperature Controllers

These types of controllers are capable of controlling single thermoelectric modules and would provide a temperature sensor for monitoring the temperature. These controllers are cost efficient for applications and can be integrated to the main products.

- Multi-Channel Temperature Controllers

This type of controllers is capable of controlling two different Peltier Modules which are connected in series or parallel according to the requirement. To reduce the thermal shorting across the faces of the Peltier modules, stacking multiple Peltier modules is recommended.

- Benchtop Temperature Controllers

These thermoelectric controllers are used specifically for research or laboratory environments as it is precise, versatile and compact device with stable control and data-logging. Some of the fields like Scientific Instrumentations for calibrating products, electronic cooling like laser

diodes and semiconductors, Optics, photonics and Sample preparations in microbiological and life science research.

- Digital Temperature Controllers

These controllers are advance devices which are equipped with digital displays for user interface and a control algorithm for precise control or tracking of temperature curve as per the application. PID algorithm is usually flashed for the inbuilt microcontroller, this device would be monitoring the temperature real-time. Some other features of this controllers are like Set-point configurator where user can set the required temperature to be generated, alarm configurator for setting the external fan or interrupts to stop over heating of the devices, Multiple channels are available for driving multiple Peltier modules to make suitable for multiple zone applications or experiments. Data logging and connectivity is also available for some high-end models for tracking the change in temperatures and record the data overtime, interaction with other devices is also available by using communication protocol and connecting using Universal Serial Bus (USB), RS232 or Ethernet. These controllers are also equipped with remote control and data acquisition using Internet of Things (IOT).

- Analog Temperature Controllers

These controllers are equipped with electromechanical switches like relays or bimetallic strips for controlling the flow of current through the Thermoelectric modules. These controllers are equipped with analog dials or gauges to display current, temperature and setpoint. These controllers have huge hysteresis and limited precision, basic control algorithm like on-off control or Proportional control are used, which may not provide the same level of accuracy and response like PID algorithm.

- USB or PC controlled Temperature controllers

These controllers are under the category of advanced temperature control devices, where the functioning of the controller is achieved by the interfacing the power circuit by a PC or laptop, which enables the user to leverage the computer power for set precise temperature parameters monitoring in real-time. It is normally used for scientific field for performing logging of temperature data and monitoring the performances of the material testing, environment chambers and Quality controlling. Graphical user Interface (GUI) is provided by the manufacturer where the virtual display of the parameters is visible and also the advance preset algorithms be loaded to control the devices accordingly.

- Programmable Temperature Controllers

These controllers are specialized devices designed for regulate temperature automatically by taking input from user according to the required custom temperature profiles or sequences. This controller can handle multiple setpoints to follow and some controller provide repeat and loop functions to validate the generated temperature continuously to the set temperature. In this controller onboard microcontrollers are used to communicate with temperature sensors and provide the driving parameters for the Thermoelectric modules by the specifically designed power circuit. These controllers can be used in many applications by interfacing the required power circuit and programming the tune control algorithm to the requirement.

- High-Power Temperature Controllers

These types of controllers are specialized devices to handle large current and power levels to make it suitable for achieving significant cooling or heating capacity. These types of controllers are more suitable for driving Peltier modules as it demands huge current and specifically designed for industrial applications, Aerospace and defence for simulating and testing components and materials in rapid heating or cooling environment and in semiconductor manufacturing field it is used for annealing silicon wafer for maintaining precise temperature which is crucial for achieving desire material properties. For efficient usage of these controllers a proper heat dissipation device should be coupled to reduce overheating and enhance thermal protection by proper thermal management.

2.7 Hardware Overview on available controllers.

The thermoelectric controllers are the devices that control the functioning as per the required applications. For regular sensing the generated change in temperature or to monitor the set temperature a temperature sensor or a bimetallic strip is used. The controller hardware is designed as per the following parameters.

- Thermoelectric Module specification

The thickness of the conductor traces width and size of the board depends on the working voltage ratings, maximum current, heating and operational temperatures of the board components for driving the thermoelectric module as per the requirements. The number of Peltier modules to be driven also matters to affect the design of the board.

- Temperature Range

Fixing the temperature range for operating the Peltier module would also help in choosing the significant electronic components for higher efficiency usages and reduce the heating of the board unnecessarily. This also helps to source the adequate components than choosing more capable components which is not recommended.

- Microcontroller/PLC Selection

The microcontroller would be the main component for taking decision in processing or reversing the signal for supplying or reversing the power supply to the Peltier module. The temperature sensors communication protocol or by direct change in voltage as per the temperature, while these signals should be processed by the microcontroller for comparing the setpoint. The user interaction should be also included in the form of LCD display for displaying the set and generated temperature and any devices like potentiometer or buttons to set the required setpoint, so the proper suitable microcontroller should be chosen for interfacing the inter acting components. Some of the microcontrollers are connected with the computers to monitor the real time changes in generated temperatures sensed by the controller where the communication is possible preferred physical cable as per the requirements, for example RS-232, UART, SPI or I2C.

- Power supply selection

The proper power supply should be chosen to power the components on the board for operating as per the datasheet. On board regulators should be used for driving the required voltage and current to the onboard components as per the data specifications. To reduce the unwanted heat or consumed by the regulators then proper sizing of regulators must be choose accordingly as per the requirement.

- Heat sink for board components

The on-board components which generate heat while driving current like power MOSFETS should be provided by proper heatsink for safe operation as per the datasheet. By this the heat would be properly transferred out of the board than affecting other components.

- Safety features

The electronic circuit may get voltage fluctuations and which affects the components, to avoid this a proper filtering should be included in the circuit design by adding capacitors. To reduce the power source fluctuation and current spikes, a suitable fuse should be used.

- Cost and power efficiency

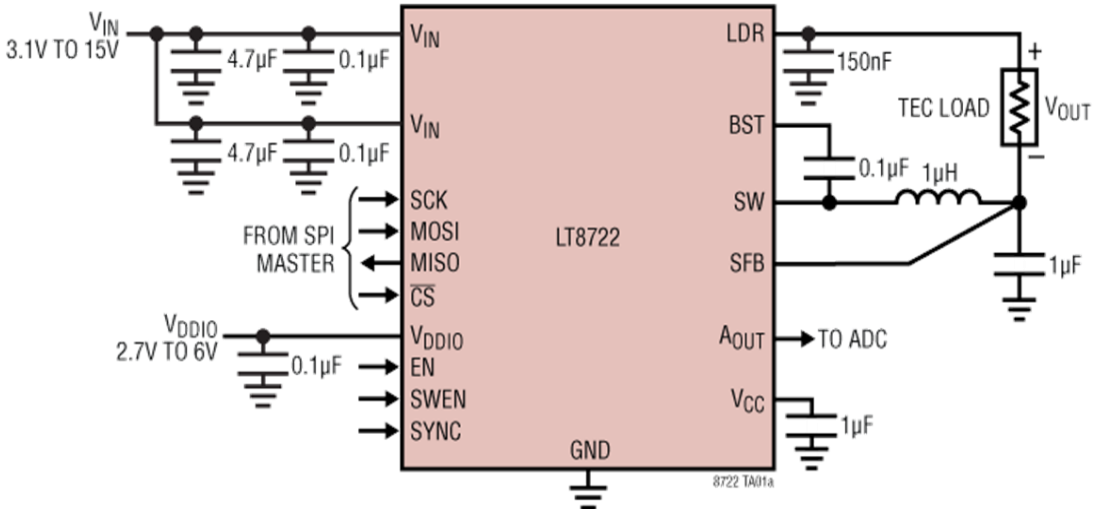
The overall design and manufacturing should be under budget as per to provide the desire to control the considered Thermoelectric module with performance and functionality at affordable prices. Considering the particular components meet requirements than over engineering is preferred. Following the cost-effective manufacturing without compromising the quality is recommended and the integration and compactness of the board design would improve the portability and reduces the manufacturing processes.

- Mechanical design

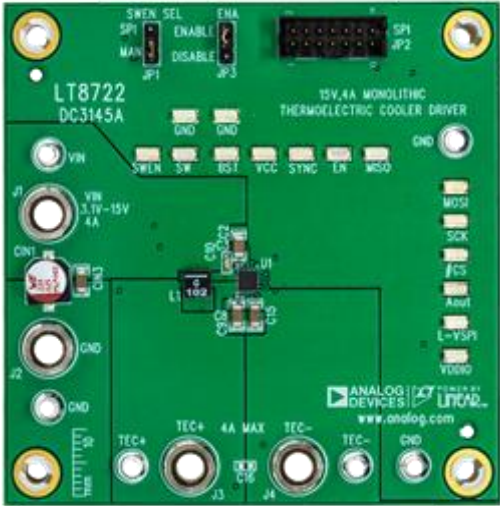
The board design should be optimised to support the other peripherals interface by providing onboard integration by proper mounting or by adding a connectivity port. The exposed environment should also be considered to deviate the heat flow to the environment. The proper enclosure design is recommended to protect the components to withstand force of 2G.

Following are the market available Thermoelectric controllers.

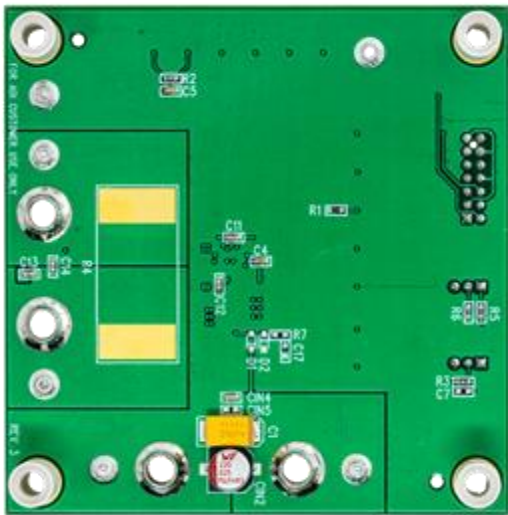
2.7.1 LT8722



(a)



(b)



(c)

Image 6:LT8722 (a) Schematics, (b) Demonstration Circuit Front Side, (c) Demonstration Circuit back side

LT8722 as shown in the Image 6 (a) is an ultracompact high performance with highly efficient full bridge DC/DC convertor, which is half driven by Pulse Width Modulation (PWM) from the external microcontroller and other half is driven from power stage. LT8722 can be used with single inductor and capable of producing ± 54 W power to the load. A 25-bit Digital to Analog convertor (DAC) is integrated to control the voltage output from LT8722, while two 9-bit DACs are additionally equipped to control the positive and negative output current limit.

A Serial Peripheral Interface (SPI) protocol can be used to tune and control LT8722 with external microcontroller or microprocessor, as per the company recommendation, EVAL-ADuCM362-ADuCM363 kit is used for controlling. The SPI would operate at 10 MHz for fast readback and controlling and to set the maximum regulation voltage and current limits with updating chip status. The chip is controlled externally by a computer were enabling or disabling is through SPI interface. This device is capable to supply maximum of 4 A and wide range of 3.1 to 15 V input with 92.6% efficiency. The SPI interface allows to get set output regulation voltage, set output current limit, device status and on/off the device. The dimension of LT8722 is 3×3×0.95 mm LQFN package with 18 leads. Maximum operating temperature of the package is 260 degrees centigrade. For operating this driver, logic high voltage should be supplied for SWEN pin.

Demonstration Circuit 3145A as shown in the Image 6 (b) is an evaluation kit used to interface the thermoelectric cooler driver like LT8722. The kit is capable of operating in 3.1 V to 15 V and the output voltage is decided by the SPI interface by the computer, the maximum current is 4 A with switching frequency of 500 kHz to 3 mHz. The kit can be controlled by using a Linduino board to monitor and can be controlled by using command line in Arduino IDE.

2.7.2 MAX1978/MAX1979

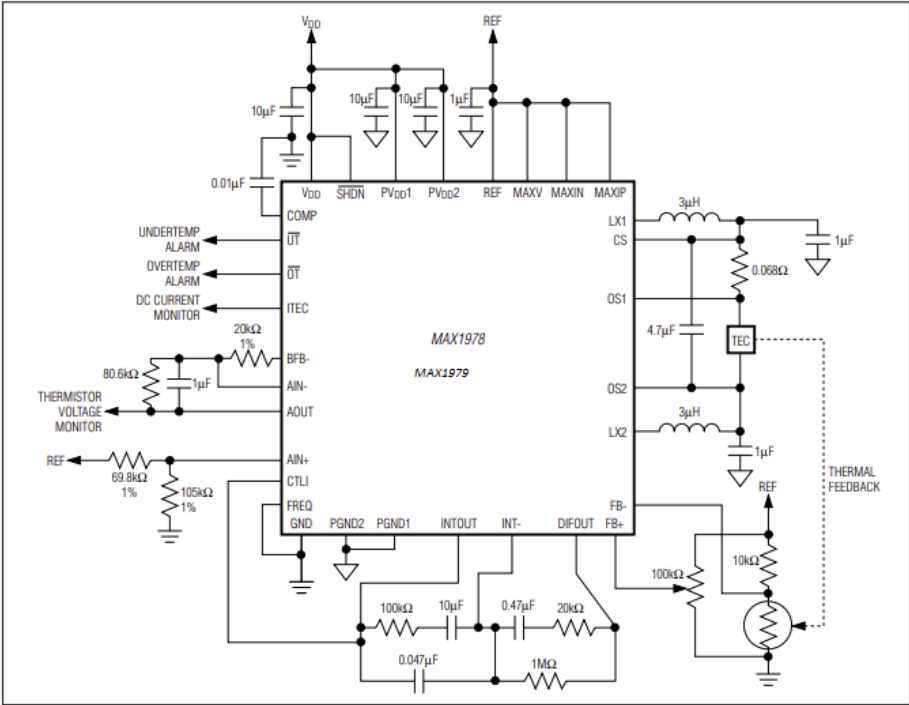
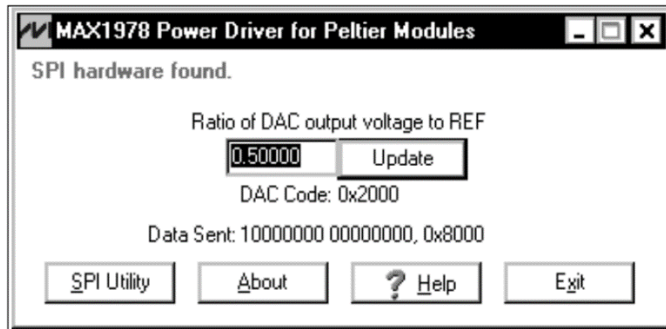
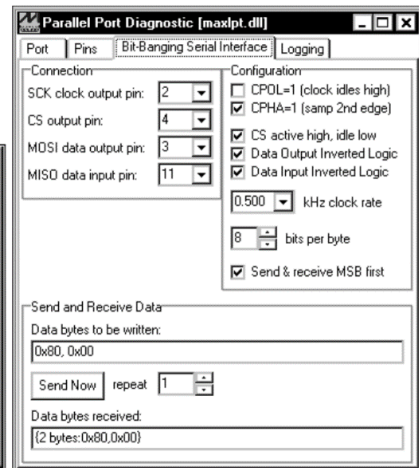


Image 7: MAX1978/MAX1979 schematics

MAX1978/MAX1979 are highly efficient thermoelectric module controllers, which are very small (circuit footprint $<0.93\text{in}^2$ and circuit height $<3\text{mm}$) and reliable for precision controlling the temperature manufactured by Maxim Integrated. These controllers are single-chip, has in-built power MOSFETS and thermal control-loop schematic to minimise the external components, which is highly portable in integrating these chips in a Printed Circuit Boards (PCBs). The thermal stability can be achieved to 0.001 C and temperature ranges of control is $-40\text{ }^\circ\text{C}$ to $+85\text{ }^\circ\text{C}$, both the are manufactured in 48 Thin QFN-EP (EP- Exposed pad). The MAX1978 is used for bipolar output current of $\pm 3\text{ A}$ by two synchronous buck regulators, which is suitable for setting for both heating and cooling but when the setpoint is set close to the natural generated temperature then no current is driven to the TEC. These chips are mostly suitable for heating than cooling as for cooling more current should be driven to the thermoelectric module. The inbuilt MOSFETS are designed to obtain noise and EMI (Electromagnetic Induction) reduction. The driving current is directly controlled than the voltage to reduce the current surge and separately heating and cooling current /voltage limits are provided for proper protection of the thermoelectric module. The pins referring Image 7 MAXIP, MAXIN and MAXV are used to set Maximum Positive TEC current, Maximum Negative TEC current and Maximum Bipolar Voltage respectively by providing the DAC input as shown in Image 8(a) from the user interface to the REF pin, and the maximum TEC voltage is 4 times the V_{MAXV} . The pin FREQ sets the internal oscillator to 500 kHz when it is connected to GND and 1 MHz when connected to VDD, it is observed that maximum efficiency is obtained when the FREQ is connected to ground. The MAX1979 is a unipolar operating from 0 to 6 A which is suitable for only heating or cooling application. The chip is integrated with a chopper- stabilized instrumentation amplifier and high precise integrator amplifier for achieving PI or PID controllers. For control feedback, an external Negative Temperature Coefficient or Positive Temperature Coefficient thermistor or any temperature sensor can be used. The operating temperature and current is are provided to monitor by the interface as Image 8(b). An over temperature and under temperature alarm pins are provided to monitor by the user interface.



(a)



(b)

Image 8: (a) DAC Input User Interface, (b) Monitoring Interface.

These integrating chips are usually used Fiber Optic Laser Modules, Wavelength Division Multiplexing (WDM), Dense Wavelength Division Multiplexing (DWDM) Laser-Diode Temperature Control, Fiber Optic network Equipment, Telecom Fiber Interface and Erbium-Doped Fiber Amplifier (EDFA) Optical Amplifier.

2.7.3 TC-48-20



(a)

(b)



(c)

Image 9: (a) MP-3193, (b) OEM Design, (c) Hardware.

TC-48-20 as shown in Image 9(b) is a simple to use and also powerful controller for only cooling or heating, this controller can be set parameters by RS232 cable from the computer and a MP-3193 sensor as shown in Image 9(a) would be utilized for feedback of temperature. This controller can be used for high power and variable -output linear supplies. For advance setup for this controller a company software based on LabVIEW source code is given to the customer for graphical interface and controlling, while most of the configuration can be done using keypad. This device would use PWM for controlling the temperature generated or absorbed by the Thermoelectric module from loads required 0 to 50 VDC with onboard power transistor, and

two power channels are available for controlling the fan, pump and resistive heaters for supporting the compatibilities of system peripherals for keeping system in control. This device can also be used for controlling a linear powered output load with directly switching the driving current, which allows to limit the heat wastage. By this the linear controlling the loads, elimination of the fluctuations to electronic glitches are observed.

TC-48-20 can be setup in single or dual power parameters for controlling the system that is ≥ 9 VDC, < 50 VDC input supply for control circuitry and Thermoelectric system for single power supply configuration, and for dual power supply configuration it is recommended ≥ 9 VDC, < 50 VDC at 150 mA for controller circuitry and ≥ 0 VDC, < 50 VDC and maximum limiting to 20 A for both Thermoelectric system and two external peripherals (1 A each). The controlling temperature is ranged from -20 °C to $+100$ °C using MP-3193 thermistor or -20 °C to $+85$ °C using 10k ohm equivalent thermistor.

2.7.4 TEC-8A-24V-PID-HC-RS232

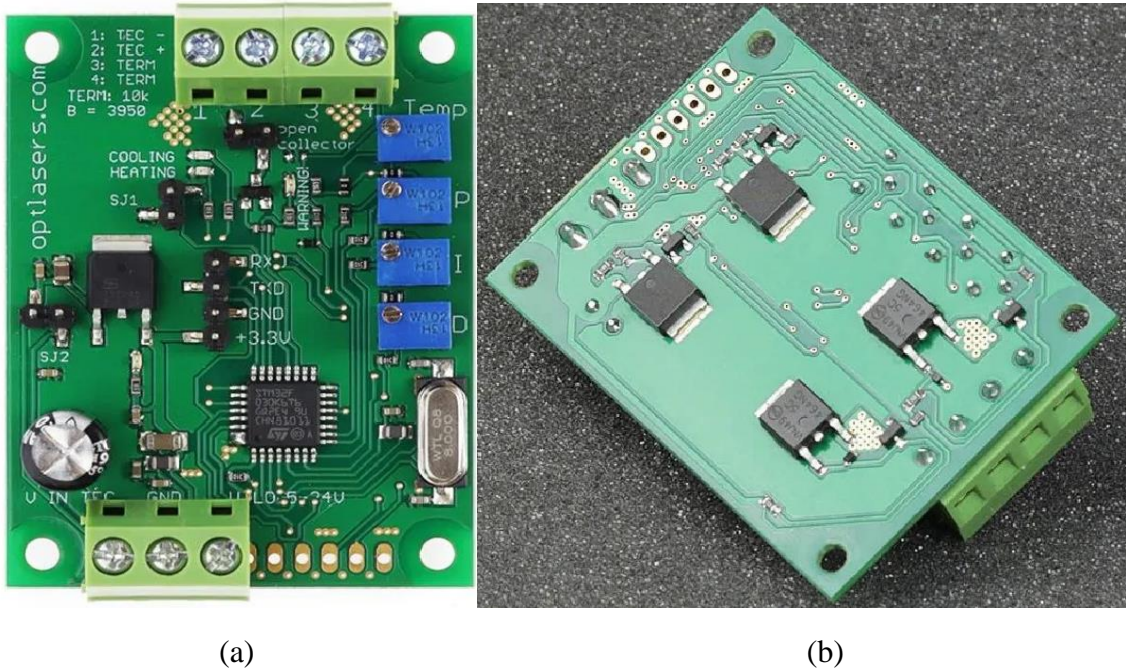


Image 10: TEC -8A-24V-PID-HC-RS232 (a) Front Side, (b)Back Side.

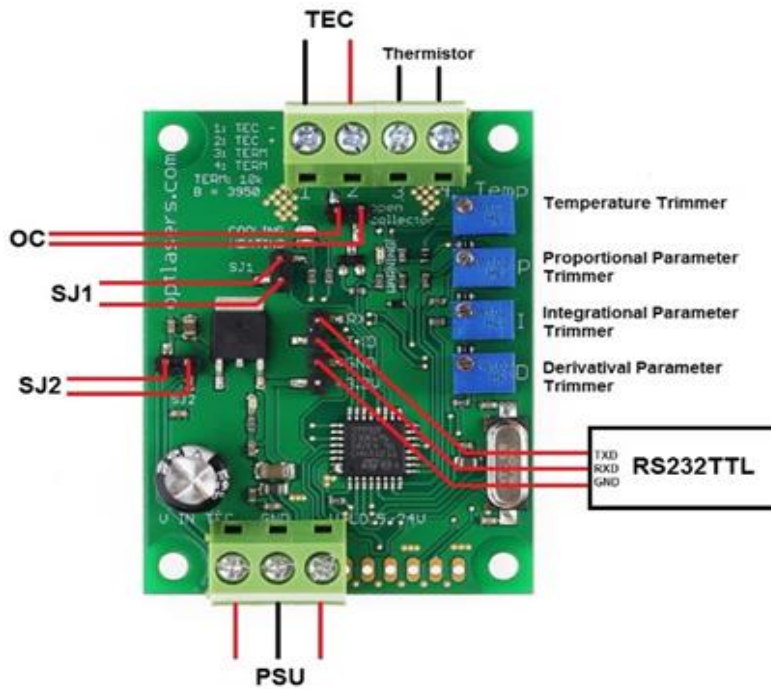


Image 11: TEC -8A-24V-PID-HC-RS232 Connections.

The thermoelectric controller is been manufactured by the company Opt Lasers for controlling the temperature of laser diode coupled with thermoelectric module; the hardware is programmable controller used to drive single Peltier module. By observing the Image 10(a) of this controller, it is noticed there are 5 power MOSFETS, 4 regulators (for setting P, I, D and temperature trimming variable) and a basic onboard STM32f030 microcontroller. Behind the board as shown in Image 10(b) it is seen the 4 MOSFETS are used to obtain H-bridge configuration circuit to drive power to the Peltier module. For booting the microcontroller, there is on board 8kHz quartz crystal. Proper wire connectors are provided for input power source (3 pin) and output driven power is sent to dedicated connectors to the Peltier module as shown in the Image 10(a). The board dimensions are $59 \times 44 \times 18 \text{ mm}^3$.

This thermoelectric controller is capable of driving Peltier cells up to 8 A and 24 V rating but specially considered to control Peltier cell with rating 12 V. The PID parameters can be manually set or set by a computer using SJ1 jumper closed, on board temperature setpoint trimmer, Proportional, Integrator and Derivative parameter adjustments are available to set parameters manually when SJ1 jumper is open. The communication and control with computer are done using UART protocol with RS232TTL cable which is available in the company website. This controller can only control single Peltier module with PWM for both heating and cooling by changing the direction using 4 power MOSFETS in H-Bridge configuration as seen in the second Image 10(b). This device can set the temperatures from the range $-100 \text{ }^\circ\text{C}$ to $+100 \text{ }^\circ\text{C}$ and temperature accuracy of $\pm 1 \text{ }^\circ\text{C}$ within $-50 \text{ }^\circ\text{C}$ to $+50 \text{ }^\circ\text{C}$ and stability of $\pm 0.05 \text{ }^\circ\text{C}$. The factory

temperature setpoint to this controller is 20 °C and it is recommended to use the sensor P0K1.202.3FW.B.007. This controller has no reverse polarity protection, the circuit would be damaged if there is reverse polarity. Over heating and cooling cut off is seen when the generated temperature is reached higher than 120 °C or below 120 °C. This circuit can be powered by single power source or two power sources, when the single power source is connected then SJ2 jumper should be connected to supply power for both power circuit and microcontroller with sensor as shown in the Image 11. If two power sources are used then the jumper should be disconnected, this feature helps in driving power for microcontroller and power circuit separately when required to obtain more heating or cooling application.

2.8 General Assumptions on Thermoelectric controllers

The industry available controller is designed on the usage considering particular applications, which varies on the power parameters. Controllers have a feedback loop for monitoring the temperature and to update the parameters on driving current. The signals from the controller are planned on the construction and working of power circuit to modify the direction and power to the Peltier module, i.e., controller can update the driving power to Peltier module either by modifying PWM signals or by changing the switching sequence among the signalling pins by considering the feedback from the sensor. The power circuit designing parameters are considered by deciding number of Peltier modules to be driven, maximum current and voltage the circuit should be capable of withstanding, integration of electronic components for achieving compact size and form factor. The efficiency and power management for driving supporting peripherals should be considered during design i.e., to distribute the required operating current and voltage for microcontroller, and user interface like lcd display etc. The controllers are capable of providing multiple control interfaces such as analog, digital or computer-based communication. Common interfaces can be seen are USB (Universal Serial Bus), RS-232, RS-485, and Ethernet. To protect the thermoelectric device and the system controller, an overcurrent protection, over temperature protection and a short-circuit protection would be seen in the schematics of the power circuit.

3 Ideology and approaches for developing and fabricating the Thermoelectric Controller.

The following system schematics are planned to execute for achieving to control the thermoelectric devices.

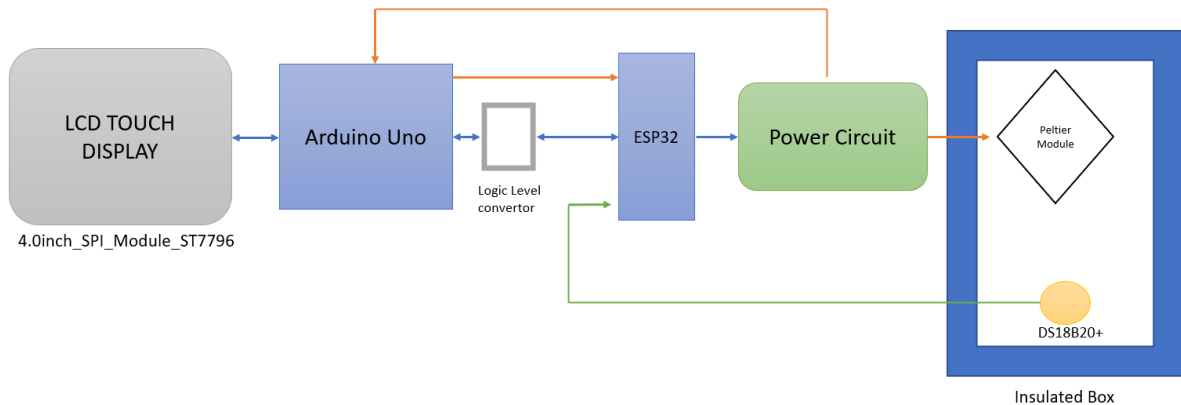


Image 12: System Schematics

The whole system provides the interface to the user with data logging and also control the power driving for the Peltier module. The given schematic as shown in Image 11 would explain the interface of the components to achieve the required characteristics as a system. Mainly The user interface is obtained by a 4-inch LCD touch display to display the data of the insulated box and also the user can set the required temperature from the working range. The touch interface is controlled by Arduino Uno which is powered by the power circuit, the setpoint variable is transferred to the ESP32 by Universal Asynchronous Receiver/Transmitter (UART) with Half-duplex mode.

A logic level convertor is used for connecting between Arduino uno and Esp32 as they operate at different voltage levels, which protects from damaging of the Esp32 board, tolerance of noise is increased with seamless communication. The Esp32 further signals the power circuit to control the driving power for the Peltier module as per the user setpoint and the feedback from the DS18B20+ sensor is sent to the Esp32 for monitoring the generated temperature by the Peltier module which helps to modify the signals sent to the power circuit for changing the power parameters instantly for maintaining the generated temperature equal to the setpoint. The whole system can be powered by an external power source or a Lithium polymer battery.

ESP32 development board is selected which has dual core Xtensa dual-core 32-bit LX6 micro-processor, operating at 160 or 240 MHz obtain performance up to 600 DMIPS to control the

power circuit as it is capable of handling control algorithm, interfacing with the temperature sensor and communication with PC or another microcontroller. The power from the main source is branched to power Arduino Uno and Esp32 from Arduino Uno as it has internal regulator to provide 3.3 V DC, the touch display is also powered by the Arduino Uno, by this the whole electronics can be fascinated to the insulation box and can be used for multiple application for heating and cooling. To external heatsink fans attached to the Peltier module is powered by the Power circuit with the signals of the Esp32.

A typical digital temperature sensor is the DS18B20+. It makes use of the 1-Wire protocol, which enables several devices to talk to one another over a single data line. The DS18B20+ offers an accuracy of up to 0.0625°C and a wide working temperature range of -55 °C to +125 °C. Either an external power source or parasitic power, which comes from the data bus, can power it. The DS18B20+ is appropriate for a variety of applications since it comes in a variety of form factors, including through-hole, surface-mount, and waterproof variants.

The following sections would illustrate the procedures followed to build the Thermoelectric controller in details.

3.1 Design of power circuit.

The heat rejected or heat applied for cooling or heating the insulated box was calculated to be 65 W especially for the storage of insulin. By considering the Peltier module ATS-TEC40-35-018 an industrial grade, and capable of removing heat of 111.5 W, it is considered to design the power circuit to withstand current of 8 A but working at 3 A max as the Peltier module is more powerful than the requirement, where the same insulation box can be used for further cooling or heating for different application.

3.2 Schematics of the power circuit.

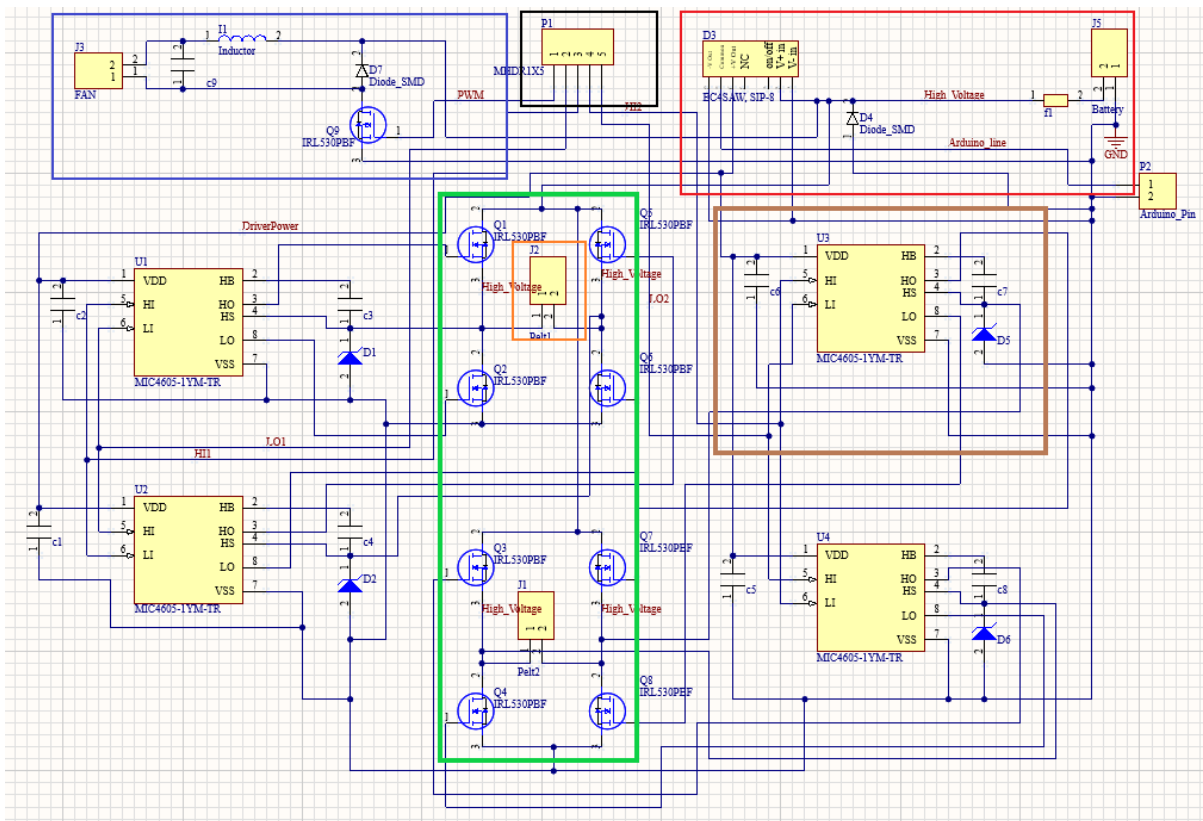
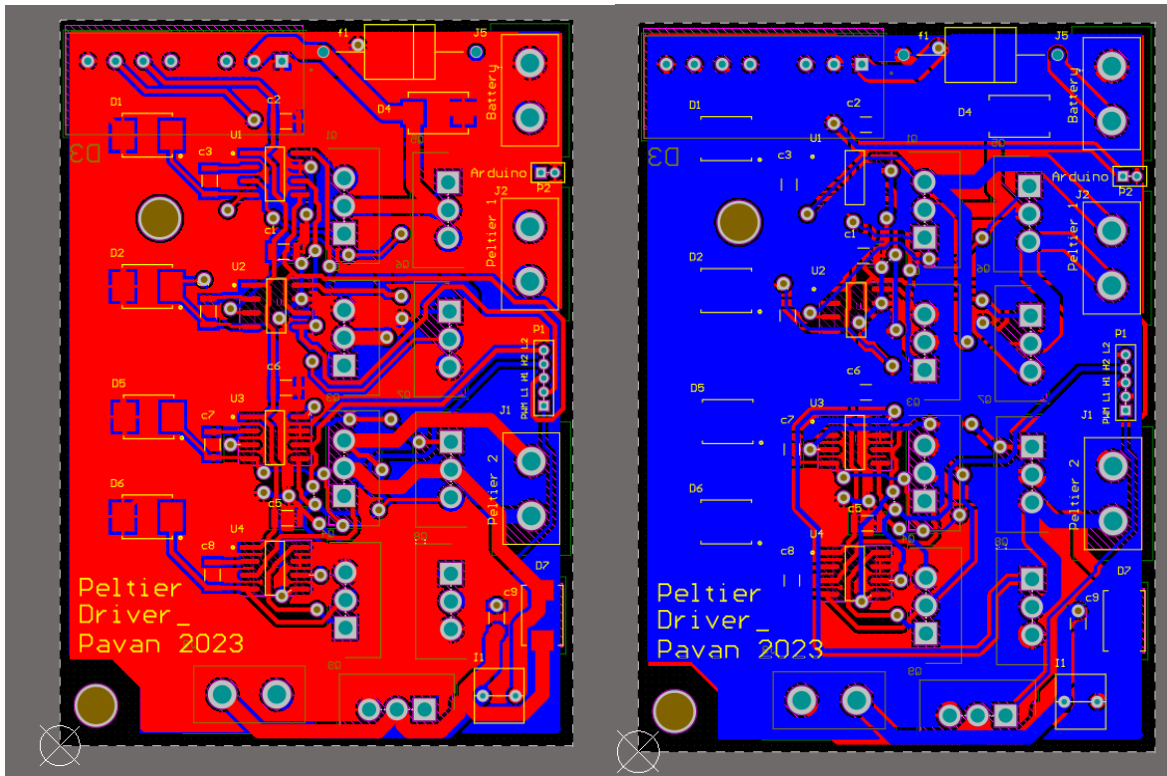


Image 13: Schematic layout of Power circuit.

The schematic layout as shown in Image 12 consists of two H-bridge circuits (inside green box) which are responsible for changing the current direction in the connectors Pelt1 and Pelt2 and these H-bridge circuits are connected in parallel to each other, a power circuit (inside the red box) to divide the voltage to a microcontroller and the H-bridge is obtained by using suitable DC/DC converter. The simple chopper circuit is also connected to the power circuit for powering the fan necessary for dissipating the generated heat from Peltier to the surrounding air. Molex connectors (inside the black box) are used to drive with the MOSFET drivers where signals are transferred from the microcontroller. The connection for the MOSFET driver is shown inside the brown box in Image 12.

The power circuit consists of an isolated DC/DC convertor EC4SAW-24D05N, through hole with input range 9-36 VDC input, ± 5 VDC output with ± 600 mA current. The main reason to use this convertor is to divide the voltage of $+5$ V to microcontroller single channel and $+10$ V to the MOSFET driver combining both the channels, while the total supplied voltage would be passed to the Peltier module. The H-bridge circuit consists of N-MOSFET IRLUO24NPBF from Infineon technologies which are capable of working at maximum 55 V continuously (VDSS) and 17 A (ID) current, resistance offered is 0.056Ω (RDS (on)), these MOSFETS can

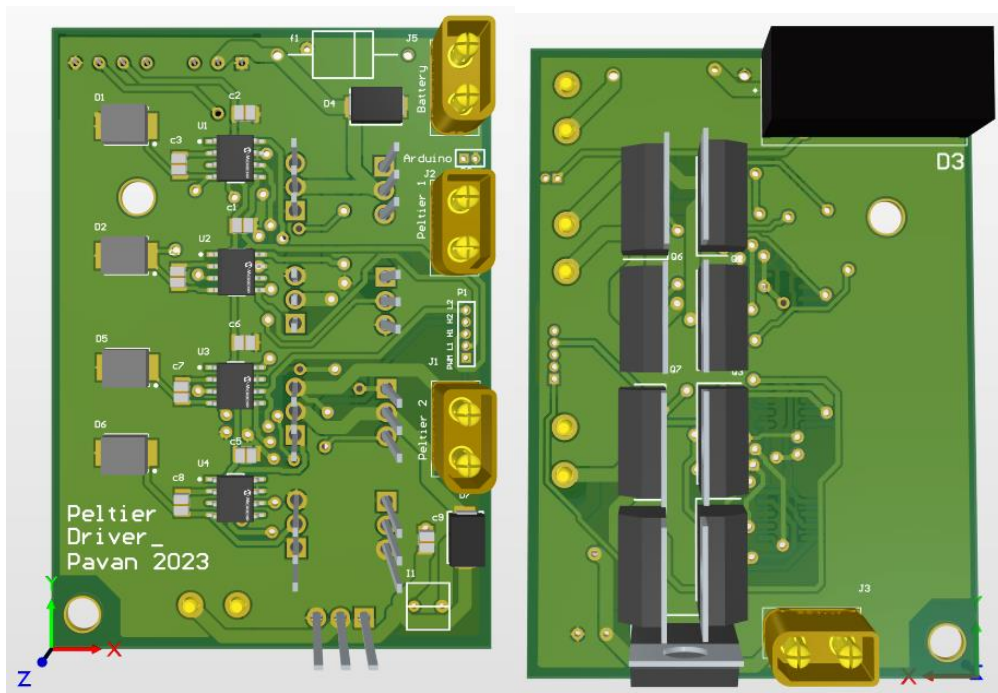
be operational till 200° C where the heat sink for cooling MOSFETS is not required for this application. To drive these MOSFETS, a half bridge MOSFET driver MIC4605-1YM of 8-pin SOIC package is used. These drivers are capable of supplying 5.5 V to 16 V to the gate with operating junction temperature of -40 °C to +125 °C which is suitable for many motor driving applications. To activate the proper gates of MOSFET it is recommended to check timing diagram of the MOSFET driver from manufacturer's datasheet to signal LI and HI pins from microcontroller for changing the direction to the Peltier module through HO and LO pins. There is an inbuilt undervoltage lockout (UVLO) with the driver which led to use bootstrap capacitor, in the circuit the capacitors of 0.2 μ F is used for both bootstrap and to reduce fluctuations from the supply. The power circuit consists of fuse holder connected to the battery supply and DC/DC convertor, and a diode is connected in blocking direction to the ground to the battery supply for reduce power surge for the circuit. The chopper circuit consists of a MOSFET to open the voltage for the external fans placed on both sides to the Peltier module by PWM from microcontroller, to keep the current continuous a capacitor is connected parallelly and an inductor in series to reduce the fluctuation of the voltage across the fan, this circuit is used basically to control the fan speed as desired and also to overcome the different voltage level of the fan compared to the voltage supply from the convertor. After validating the schematic layout, the Printed Circuit Board is designed. Image 13 illustrates the arrangement of the components to keep the PCB compact and the dimensions are 4.7 by 6.6 cm.



(a)

(b)

Image 14: PCB (a) Top Layer, (b) Bottom Layer.



(a)

(b)

Image 15: PCB board (a)Front side (b)Back side

3.3 Interfacing components

3.3.1 Interfacing ESP32 with DS18B20+

The temperature measurement from the digital sensor DS18B20+ can be done with ESP32 using two libraries OneWire and DallasTemperature. The sensor can be powered in two different modes like normal and parasite mode. For the controller application, we use normal mode where, power is supplied by external source. To keep the data transfer stable, a 4.7k ohm resistor is connected between data pin and power pin of the sensor, and GND of the sensor is connected to the GND of the ESP32 as shown in the Image 15 (b) [23, 24].

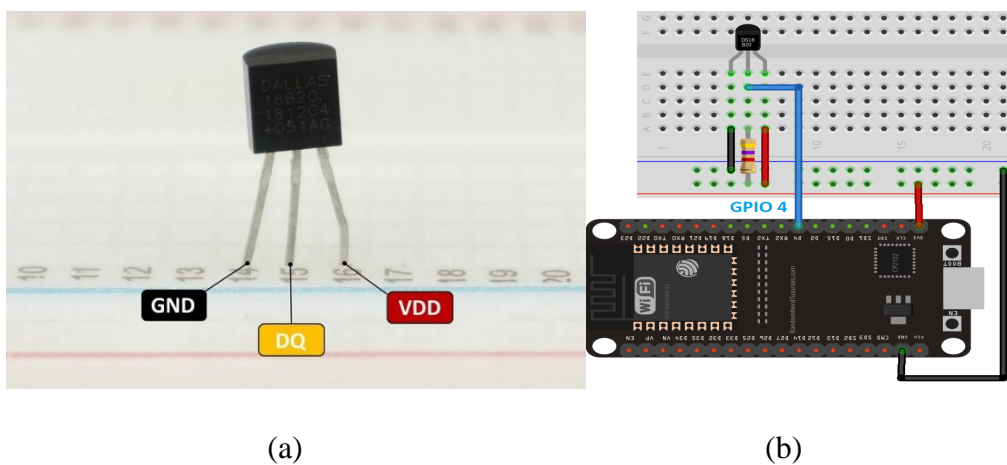


Image 16: DS18B20+ (a)Pinout, (b)Normal Mode Connection.

The measured data from the temperature sensor can be obtained two methods either By Index or By Address.

- By Index:

In this approach, the temperature sensor is been identified by the index numbers and by the library the measured data can be obtained by calling the respective index number in the same data line. For the temperature controller, we use this method as it is easy to use by finding the default index.

- By Address:

In this approach, the temperature sensor is defined by the hexadecimal address and every time the measured data from each sensor can be called by the defined address and assigned to variable in the code.

Code explanation for reading data from DS18B20+

```
#include <OneWire.h>
#include <DallasTemperature.h>
//The required libraries are included.

const int oneWireBus = 4; //Defining the GPIO pin connected for sensor.

OneWire oneWire(oneWireBus); //Passing OneWire instance to communicate with
existing OneWire devices.

DallasTemperature sensors(&oneWire); //Passing OneWire reference to Dallas
Temperature Sensor

float temperatureC, temperatureF; //Declaring variables

void getTemperature(){

    sensors.requestTemperatures(); //To read the updated measured temperature this
method is called.

    temperatureC = sensors.getTempCByIndex(0); //To read the measured data by
Celsius getTempCBy Index(0) is
used.
    temperatureF = sensors.getTempFByIndex(0); //To read the measured data by
Fahrenheit getTempFBy Index(0) is
used.

    Serial.print(temperatureC); //Measured data is printed on serial monitor.
    Serial.println("°C");
    Serial.print(temperatureF);
    Serial.println("°F");
}
```

Source Code 1: DS18b20+ interface with ESP32.

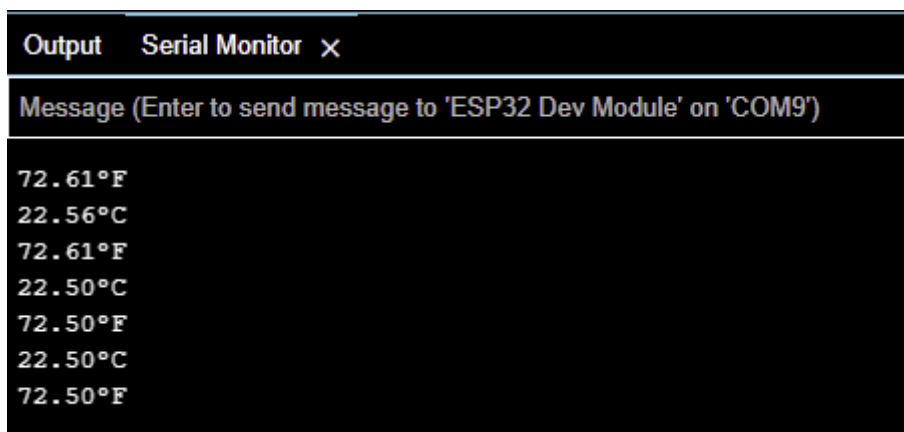


Image 17: Output readings from Temperature sensor.

3.3.2 Interfacing Arduino Uno and ESP32

The user interface would be controlled by Arduino Uno which is based on Atmega328P which is 8-bit Advance Virtual RISC(AVR) microcontroller equipped with 32 kB of flash memory, 2 kB of Static Random-Access Memory (SRAM) and 1 kB of Electrically Erasable Programable Read-Only Memory (EEPROM) for data storage, it consists of 16 MHz quartz crystal, 6 analog input pins, 14 digital input/output pins, a USB port for data communication between itself and computer and a power socket, it has dedicated Universal single Asynchronous Receiver/Transmitter(UART) port, Inter-Integrated Circuit(I2C) and Serial Peripheral Interface(SPI) for communication with other devices like sensors, microcontrollers, SD card and display modules. While in this project, the required setpoint from the user would be inserted using touch display which is connected using SPI interface with Arduino and stored in the variable in string format. On the other side, ESP32 which has dual core low power Xtensa 32-bit LX6 microprocessors which is integrated with 440 kB of ROM for booting and core function, 520 kB of On-chip SRAM for data and instructions, 8 kB of SRAM RTC FAST for Memory and used for data storage and another 8 kB of SRAM in RTC Slow Memory and can be used by co-processor during the Deep-sleep Mode.

ESP32 is connected to Arduino Uno by a logic level shifter for obtaining a UART communication protocol for data transfer as it works on 3.3 V but Arduino uno works on 5 V. UART communication or Serial communication is one of the simple protocols that can be used between two microcontrollers. It is usually performed by just connecting two wires of same potentials and a common ground between the communicating devices. By this protocol, simplex mode, half-duplex mode and full- duplex mode can be achieved according to the requirement. In simplex mode the data would be transferred in single direction where only single data wire would be required, in half-duplex mode the transfer of data would happen in bi-directional where the devices would take chances to transfer data one after other. In full-duplex mode, both devices would simultaneously i.e., transmitting and receiving the data in same time among the devices. Comparing to other communicating protocol like SPI and I2C, UART protocol communication requires only 2 wires while others need a hardware chip for communication. In this project, the decimal value of setpoint from the user should be sent to ESP32 for comparing it to generated temperature sensed from the DS18B20+ sensor, and the decimal value of generated temperature should be sent to the Arduino Uno and display on the display screen whenever the variable is changed. In this communication protocol, the serial form data would be converted to a parallel form by the CPU of one device from TX Pin and to the RX Pin of the other device where the parallel data would be converted to serial form data by the receiving device CPU and displayed.

To understand the starting and ending of the data, a transmitting and receiving bits are used to guide the CPU to start and stop reading the transferred data. The reading and transferring frequencies of the receiver device and transmitter device are set to a particular baud rate for specifying the flow of bits. The length of the communication wires, there can be a lag in data transfer among the devices, it is recommended to keep short wires until change in baud rates is less than 10 %.

The ESP32 development board has three UART ports like UART0, UART1 and UART2, which works on 3.3 V but Arduino Uno has one UART port works on 5 V and responsible for programming and also for serial communication.

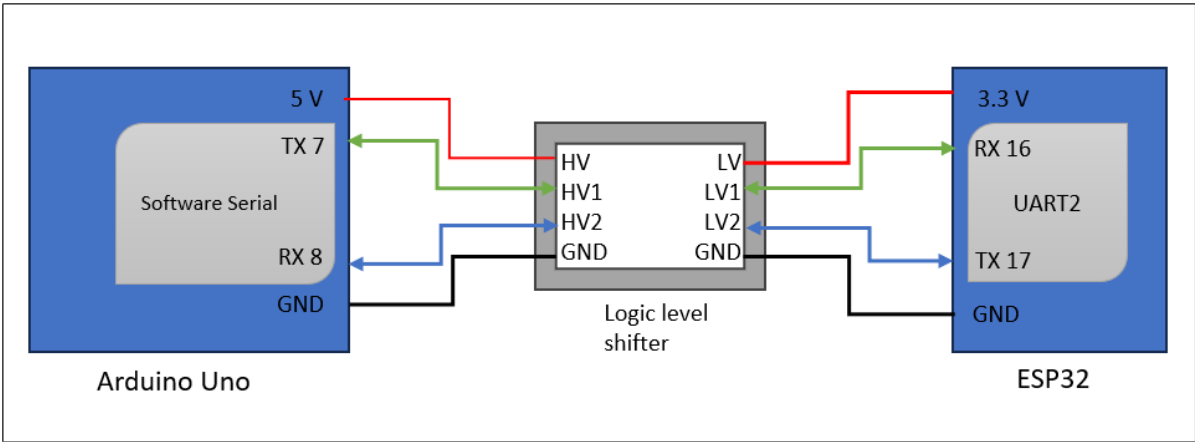


Image 18: UART wire connection schematics.

The Image 18 schematic explains the wire connections between Arduino uno and ESP32. The dedicated UART2 pins its TX2 (Pin 17) and Rx2 (Pin 16) are used for the communication in ESP32 board while by using software serial library with Arduino uno, digital pin 7 (TX) and 8 (RX) are used for UART communication while the dedicated UART port is not used for communication. The logic level shifter is placed between the Arduino Uno and ESP 32 to change the communication logic voltage suitable for the respective boards. The transmitter line from Arduino uno is finally connected to the receiver of ESP32 and transmitter line of ESP32 is connected to Receiver of the Arduino Uno. The voltages of the both boards are connected to the Logic level shifter circuit and common ground (GND) is connected.

```

#include <SoftwareSerial.h>
SoftwareSerial SerialPort(7, 8); // Initialize SoftwareSerial with RX on pin 7,
TX
                                on pin 8
float tset = -26.23; // Desired temperature value
float tgen; // Received temperature value
unsigned long lastTime = 0; // Stores the last time data was sent
const unsigned long interval = 100; // Interval between sending data (in
millisecons)

void setup() {
  Serial.begin(9600); // Initialize hardware serial communication at 9600
baud
  SerialPort.begin(9600); // Initialize software serial communication at 9600
baud
}
void loop() {
  // Listen for incoming data on the SoftwareSerial port
  if (SerialPort.available()) {
    tgen = SerialPort.parseFloat(); // Read and parse incoming float data
  }
  // Send data back on the same SoftwareSerial port
  unsigned long currentTime = millis(); // Get the current time
  if (currentTime - lastTime >= interval) {
    String tset_string = String(tset); // Convert the desired temperature to a
string
    SerialPort.write(tset_string.c_str()); // Send the desired temperature as a
string
    lastTime = currentTime; // Update the last sent time
  }
  Serial.print("tgen: ");
  Serial.println(tgen); // Print the received temperature value
}

```

Source Code 2: UART code for Arduino.

```

#include <HardwareSerial.h> // Include the library for hardware serial
communication
#include <OneWire.h> // Include the OneWire library for
communicating
                                with One Wire devices
#include <DallasTemperature.h> // Include the DallasTemperature library for
temperature sensors

#define ONE_WIRE_BUS 15 // Define the pin for the OneWire
communication
HardwareSerial SerialPort(2); // Initialize HardwareSerial on serial port 2
(RX, TX)
OneWire oneWire(ONE_WIRE_BUS); // Create a OneWire instance with the
defined
                                pin
DallasTemperature sensors(&oneWire); // Create a DallasTemperature instance
using
                                the OneWire instance
float tgen; // Variable to store the generated temperature
float tset; // Variable to store the set temperature

unsigned long lastTime = 0; // Stores the last time data was sent
const unsigned long interval = 1000; // Interval between sending data (in
millisecons)

```

```

// Function to update the generated temperature by reading sensors
void update_gentemp() {
  sensors.requestTemperatures();      // Request temperature readings from the
                                     sensors
  tgen = sensors.getTempCByIndex(0);  // Get temperature from sensor 1
}

void setup() {
  Serial.begin(9600);                 // Initialize hardware serial communication at
9600
                                     baud
  SerialPort.begin(9600);             // Initialize hardware serial communication on
serial port 2 at 9600 baud
  sensors.begin();                    // Initialize the DallasTemperature library
}

void loop() {
  update_gentemp();                   // Call the function to update the generated
temperature

  // Listen for incoming data on the HardwareSerial port
  if (SerialPort.available()) {
    tset = SerialPort.parseFloat(); // Read and parse incoming float data as the
set
                                     temperature
  }
  // Send generated temperature data back on the same HardwareSerial port
  unsigned long currentTime = millis();
  if (currentTime - lastTime >= interval) {
    String tgen_string = String(tgen); // Convert the generated temperature to
a
                                     string
    SerialPort.write(tgen_string.c_str()); // Send the generated temperature as
a
                                     string
    lastTime = currentTime; // Update the last sent time
  }

  // Print the values of tgen and tset to the Serial Monitor
  Serial.print("tset:");
  Serial.println(tset); // Print the set temperature
}

```

Source Code 3: UART code for ESP32.

Output from the above displayed codes for UART communication, ESP32 is connected to DS18B12+ sensor where the room temperature (23.13 °C) would be sensed and sent to Arduino Uno while in Arduino uno, a Tset variable is randomly set to -26.23 °C. From the serial monitor using Arduino IDE, the sensed temperature is displayed on Arduino Uno serial monitor comport while the set temperature is displayed on the serial monitor comport. The results can be seen as shown in the Image 19.

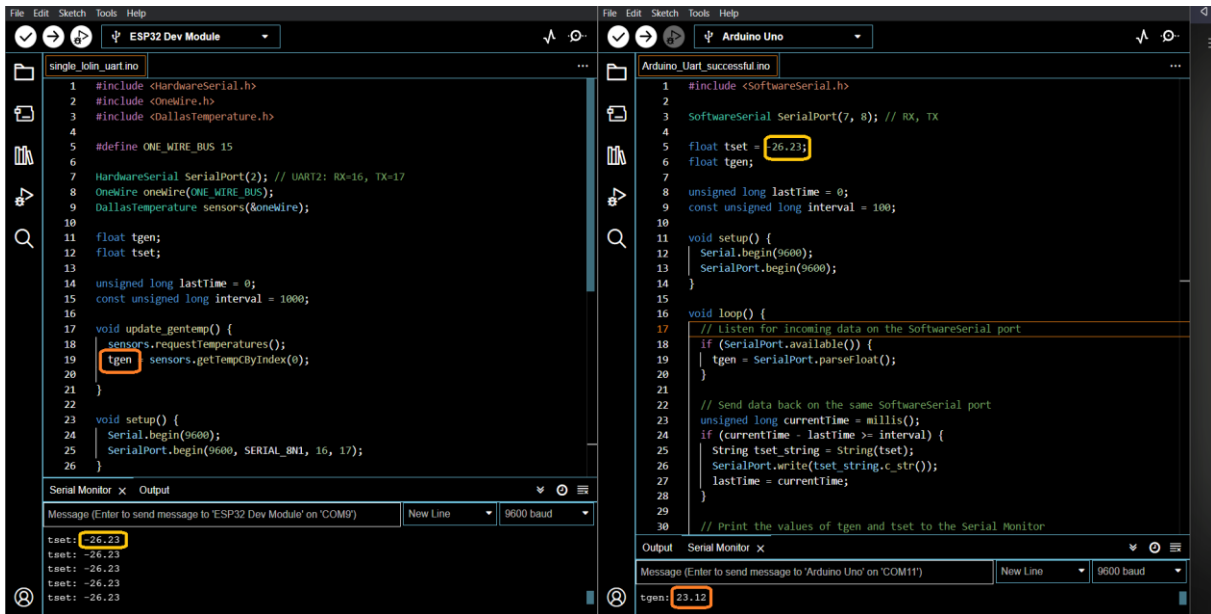


Image 19: Output results of UART communication

3.3.3 User Interface.

The user interface is simple and minimalistic design as shown in the Image 20, where the required data is displayed and the touch options are enabled to input the data from the user and to lock or unlock the screen. The design approach is observed as shown below. The SET virtual button is used to assign the value typed by using display and assign to the Tset variable. While Lock and Run virtual button is used to change the state of the display to accept input from the user or not respectively. The state of the display is indicated by the circle changing to green if accepting input from user or red if not accepting. To change the state of display accepting input to not accepting input, LOCK virtual button should be touched and for not accepting input stage to accepting then RUN virtual button should be touched.

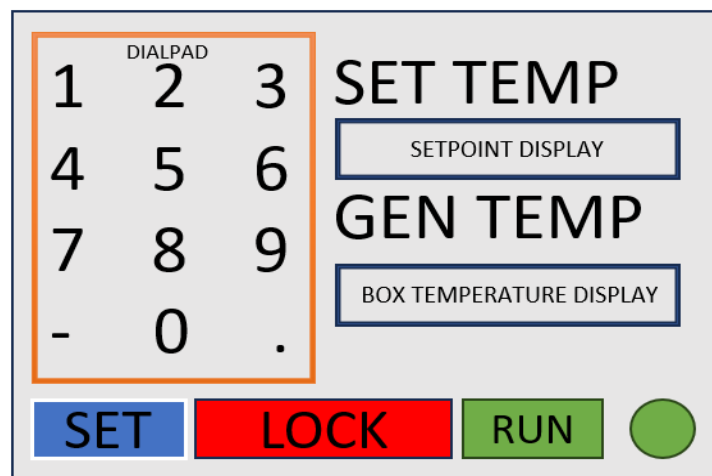


Image 20: User interface design.

The chosen display is 4.0-inch SPI Module ST7796, which is SPI serial bus enabled and require less IO Pins for usage. The display is of 480×320 resolution with optional touch function and SD card slot for displaying images. The library is user friendly to use and has a rich documentation with wide examples. The active area of display is 55.68 × 83.52 mm and with resistive touch type, while the display works with 3.3 V ~ 5 V. The wire connections are clear in data sheet and followed accordingly.

```
#include <LCDWIKI_GUI.h> //Core graphics library
#include <LCDWIKI_SPI.h> //Hardware-specific library
#include <LCDWIKI_TOUCH.h> //touch screen library
#include <SoftwareSerial.h>

//parameters define
#define MODEL ST7796S
#define CS A5
#define CD A3
#define RST A4
#define LED A0 //if you don't need to control the LED pin,you should set it
to -1 and set it to 3.3V

//touch screen parameters define
#define TCS 2
#define TCLK 3
#define TDOUT 4
#define TDIN 5
#define TIRQ 6

// Initialize LCD with specified parameters
LCDWIKI_SPI mylcd(MODEL,CS,CD,RST,LED); //model,cs,dc,reset,led

// Initialize touch screen with specified parameters
LCDWIKI_TOUCH mytouch(TCS,TCLK,TDOUT,TDIN,TIRQ); //tcs,tclk,tdout,tdin,tirq

// Initialize a SoftwareSerial port for communication
SoftwareSerial SerialPort(7, 8); // RX, TX

// ... (color definitions, spacing, variables, etc.)
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

#define SPACE_X 40
#define SPACE_Y 40
#define BUTTON_SPACING_X 20
#define BUTTON_SPACING_Y 20
#define BUTTON_TEXTSIZE 2

uint16_t px,py;
String input;
```

```

float tset, tgen, t1, t2;
int inputlength=0;
bool input_state = false;
bool machine_state = true;
unsigned long lastTime = 0;
const unsigned long interval = 1000;

void string_update(){
    // Function to update the string based on touch inputs
    // ... (if conditions based on touch coordinates)

    int inputlength = strlen(input.c_str());
    if((px>25 && px<40)&&(py>25 && py<45 && input_state && machine_state)){
        input.concat('1');
        input_state = false;
    }
    if((px>85 && px<105)&&(py>25 && py<45 && input_state && machine_state)){
        input.concat('2');
        input_state = false;
    }
    if((px>135 && px<165)&&(py>25 && py<45 && input_state && machine_state)){
        input.concat('3');
        input_state = false;
    }
    if((px>25 && px<40)&&(py>75 && py<115 && input_state && machine_state)){
        input.concat('4');
        input_state = false;
    }
    if((px>85 && px<105)&&(py>75 && py<115 && input_state && machine_state)){
        input.concat('5');
        input_state = false;
    }
    if((px>135 && px<165)&&(py>75 && py<115 && input_state && machine_state)){
        input.concat('6');
        input_state = false;
    }
    if((px>25 && px<40)&&(py>135 && py<180 && input_state && machine_state)){
        input.concat('7');
        input_state = false;
    }
    if((px>85 && px<105)&&(py>135 && py<180 && input_state && machine_state)){
        input.concat('8');
        input_state = false;
    }
    if((px>135 && px<165)&&(py>135 && py<180 && input_state && machine_state)){
        input.concat('9');
        input_state = false;
    }
    if((px>25 && px<40)&&(py>190 && py<240 && input_state && machine_state)){
        if(inputlength == 0){
            input.concat('-');
            input_state = false;
        }
    }
    if((px>85 && px<105)&&(py>190 && py<240 && input_state && machine_state)){
        input.concat('0');
        input_state = false;
    }
    if((px>135 && px<165)&&(py>190 && py<240 && input_state)){
        char ch = '.';
        const char* str = input.c_str();
    }
}

```

```

    // check if the character exists in the string
    if((inputlength !=0)&&(strchr(str, ch)== NULL)){
        input += ch;
    }
}
if((px>10 && px<105)&&(py>235 && py<300 && input_state && machine_state)){
    const char* charArray = input.c_str();
    if(input !=0){
        tset = atof(charArray);
        if(tset>65){
            tset = 65;
        }
        if(tset<-5){
            tset = -5;
        }
    }
    input='\0';
    input_state = false;
    mylcd.Fill_Rect(10,300,480,320,WHITE);
    mylcd.Set_Text_colour(RED);
    mylcd.Set_Text_Back_colour(128,128,128);
    mylcd.Set_Text_Size(5);
    mylcd.Fill_Rect(180,70,300,50,WHITE);
    mylcd.Print_Number_Float(tset, 2, 190, 70, '.', 0, ' ');
}
if((px>115 && px<270)&&(py>255 && py<300 && input_state)){
    machine_state = false;
    input_state = false;
}
if((px>280 && px<375)&&(py>255 && py<300)&& input_state){
    machine_state = true;
}
mylcd.Set_Text_colour(BLACK);
mylcd.Set_Text_Back_colour(128,128,128);
mylcd.Set_Text_Size(2);
mylcd.Print_String(input,10,300);
delay(100);
}

void update_tgen(){
    // Function to update the generated temperature display
    // ... (update the generated temperature display)

    mylcd.Set_Text_colour(RED);
    mylcd.Set_Text_Back_colour(128,128,128);
    mylcd.Set_Text_Size(5);
    float t1 = tgen;
    if(t1 != t2){
        mylcd.Fill_Rect(180,180,300,50,WHITE);
        mylcd.Print_Number_Float(tgen, 2, 190, 180, '.', 0, ' ');
    }
    t2=t1;
    delay(100);
}

void display_screen(){
    // Function to display the initial screen with buttons
    // ... (draw buttons and labels)

    int x=40;

```

```

int y=40;
float z=1.0;
mylcd.Set_Text_colour(BLACK);
mylcd.Set_Text_Back_colour(128,128,128);
mylcd.Set_Text_Mode(1);
mylcd.Set_Text_colour(BLACK);
mylcd.Set_Text_Size(5);
mylcd.Print_String("1", 20, 20);
mylcd.Print_String("2", 80, 20);
mylcd.Print_String("3", 140, 20);
mylcd.Print_String("4", 20, 80);
mylcd.Print_String("5", 80, 80);
mylcd.Print_String("6", 140, 80);
mylcd.Print_String("7", 20, 140);
mylcd.Print_String("8", 80, 140);
mylcd.Print_String("9", 140, 140);
mylcd.Print_String("-", 20, 200);
mylcd.Print_String("0", 80, 200);
mylcd.Print_String(".", 140, 200);
mylcd.Set_Text_Size(5);
//mylcd.Draw_Rectangle(10,250,110,300);
mylcd.Fill_Rect(10, 250, 100, 50,BLUE);
mylcd.Print_String("SET", 20, 260);
mylcd.Draw_Rectangle(10,10,175,245);
mylcd.Draw_Rectangle(11,11,174,244);
mylcd.Draw_Rectangle(9,9,176,246);
mylcd.Set_Text_Size(5);
mylcd.Print_String("SET TEMP", 190, 20);
mylcd.Print_String("GEN TEMP", 190, 120);
//mylcd.Draw_Rectangle(115,250,270,300);
mylcd.Fill_Rect(115, 250,160,50,RED);
mylcd.Print_String("LOCK", 140, 260);
//mylcd.Draw_Rectangle(10,250,110,300);
mylcd.Fill_Rect(280, 250, 100, 50,GREEN);
mylcd.Print_String("RUN",285, 260);

}
void setup() {
  // put your setup code here, to run once:
  // Setup function runs once at the beginning
  // Initialize LCD, touch screen, and pins
  // ... (initializations)
  mylcd.Init_LCD();
  mylcd.Set_Rotation(1);
  mytouch.TP_Set_Rotation(3);
mytouch.TP_Init(mylcd.Get_Rotation(),mylcd.Get_Display_Width(),mylcd.Get_Display
_Height());
  mylcd.Fill_Screen(WHITE);
  display_screen();
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, INPUT);
  Serial.begin(9600); // Initialize hardware UART at 9600 bps
  SerialPort.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  // If the machine is in "RUN" state
  // ... (read touch inputs, update tgen, communicate with SerialPort)

```

```

if(machine_state){
  px=0;
  py=0;
  mytouch.TP_Scan(0);
  if (mytouch.TP_Get_State()&TP_PRES_DOWN)
  {
    px = mytouch.x;
    py = mytouch.y;
    input_state = true;
    string_update();
  }
  mylcd.Set_Draw_color(GREEN);
  mylcd.Fill_Circle(425,275,25);

  if (SerialPort.available()>8) {
    String a = "";
    char c;
    c = SerialPort.read();
    while(c!='S'){
      c = SerialPort.read();
    }
    tgen = SerialPort.parseFloat();
    SerialPort.flush();

    Serial.print("tgen: ");
    Serial.println(tgen);
  }
  //SerialPort.flush();

  // Send data back on the same SoftwareSerial port
  unsigned long currentTime = millis();
  if (currentTime - lastTime >= interval) {
    String tset_string = String(tset);
    SerialPort.write(tset_string.c_str());
    lastTime = currentTime;
  }
  update_tgen();
}

if(!machine_state){
  // If the machine is in "LOCK" state
  // ... (read touch inputs, update tgen)
  px=0;
  py=0;
  mytouch.TP_Scan(0);
  if (mytouch.TP_Get_State()&TP_PRES_DOWN)
  {
    px = mytouch.x;
    py = mytouch.y;
    input_state = true;
    string_update();
  }
  mylcd.Set_Draw_color(RED);
  mylcd.Fill_Circle(425,275,25);
  update_tgen();
}
}
}

```

Source Code 4: Code for Display UI with UART communication.

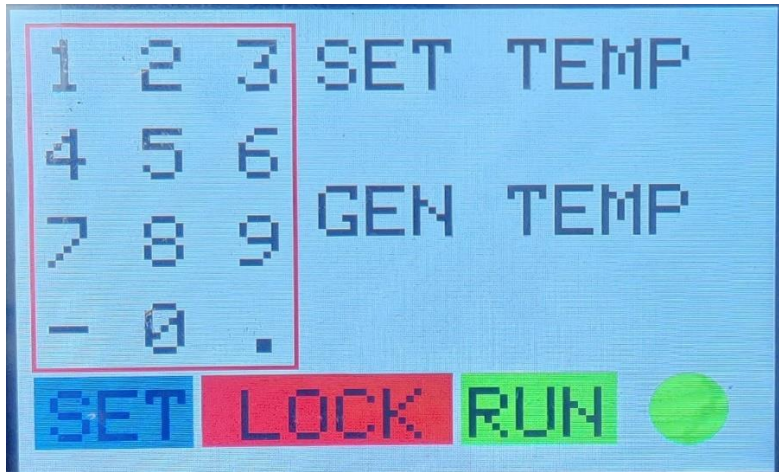


Image 21: Output LCD display.

4 Control Algorithm

The control algorithm should trigger the respective direction of the power delivery to the pelt ports for heating or cooling respectively as per the DS18B20+ sensor input to the ESP32. The study shows there are many algorithms for controlling the systems, while choosing a particular algorithm, it is recommended to understand the system or device fully and its related parameters affecting the whole system like stability, performance and robustness.

The power for generating heat or cold is decided by the direction of the power supply to the Peltier module. By looking the connections on the power circuit, it is required to control the flow of current by singling the respective power MOSFETS by the half-bridge MOSFET drivers. To keep the MOSFET driver working, it is needed to charge bootstrap capacitor to flow of current from top MOSFET to bottom as N-type MOSFETS are used. Hence the short spike of voltage should be supplied in opposite direction to refuse driver entering Under Voltage Lock-out (UVLO). By which a proper tuning of signals is expected to control the power circuit to reduce the short circuit. The driver doesn't open the MOSFET fully of both the high-side and low-side input is high, and required to generate the signals as per the timing diagram given in the datasheet by the manufacturer.

For designing of the control algorithm, it is highly recommended to understand the connections for signalling respective pins of MOSFET driver for opening respective MOSFETS to get the desired directions to drive the power to the Peltier module. The connections are shown in the Image 22

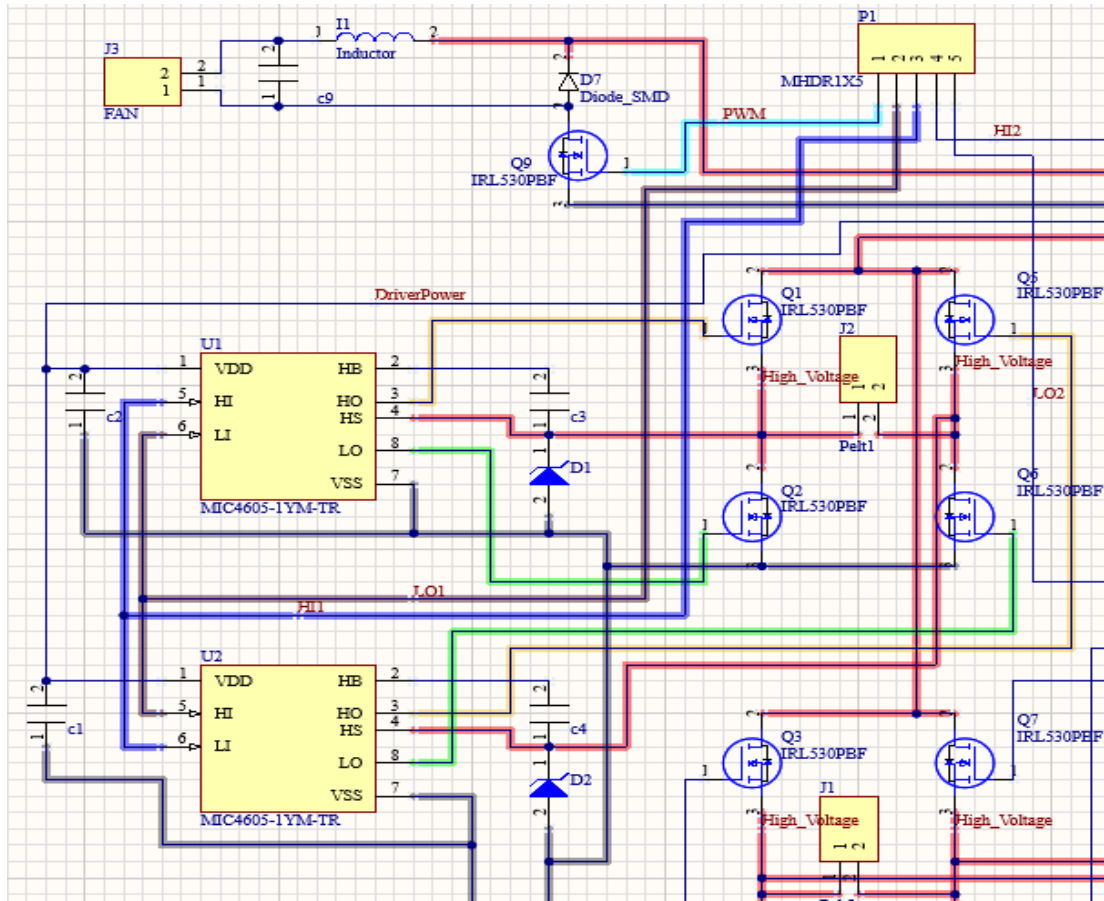


Image 22: Control connections.

The H- Bridge circuit is powered directly by the power source which is set for supplying 14.5 V and 3 A continuous. The driver is powered by 10 V with DC/DC convertor, and the signals from the ESP32 should be connected to HI and LI pins, which are connected to the Pin2 and Pin3 of Molex connector P1. Considering the MOSFETS Q1-Q2- Q6-Q5 the first H-Bridge circuit, which power the Pelt1 port, the driver U1 is controlling gates of Q1 and Q2 in half-bridge configuration and the driver U2 is controlling gates of Q5 and Q6. To drive the power from MOSFET Q1 to Q6, it is required to set HO of U1 and LO of U2 high, i.e., HI of U1 and LI of U2 should be signalled together, hence those both pins are connected to pin3 (Blue line) of Molex connector P1. While the power should be supplied in the direction from Q5 and Q6, the LO of U1 and Ho of U2 should be high, i.e., LI of U1 and HI of U2 should be signalled together, which is connected together to the Pin 2 (Brown line) of the Molex connector P1. The same way the second H-Bridge is connected to power Pelt 2 port and signals from ESP32 is given to the Pin4 and Pin 5 of the Molex P1 connector. The table below would describe the signalling of respective pin from Molex P1 with directions of power supply from the respective H-Bridge circuit.

Table 1: Signalling pin layout

			Input Signal Pin		Output Signal Pin	
H-Bridge1 Q1-Q2-Q6-Q5 (Pelt1)	Direction of power	Molex Pin signals from ESP32	U1	U2	U1	U2
	Q1-Q6	Pin3	HI	LI	HO	LO
	Q5-Q2	Pin2	LI	HI	LO	HO
H-Bridge2 Q3-Q4-Q8-Q7 (Pelt2)			U3	U4	U3	U4
	Q3-Q8	Pin5	LI	HI	LO	HO
	Q7-Q4	Pin4	HI	LI	HO	LO

4.1 PWM signal design.

Pulse Width Modulation (PWM) is one of the popular methods used in electronics for controlling the power delivered to the devices to be controlled. The power can be modified by varying the width of the pulses in a square wave signal. This way is more preferred to the applications where precise control of voltage or current is required. By using PWM for controlling a device it is easy to increase efficiency, simplicity and cost effective.

The PWM signal would consists of recurring fixed period or cycle time, where each cycle the defined width is varied. When the width is small and applied to the device, the average power delivered is low because the time spent on being low state by the device is more. In maximum power the device would be more when the duty cycle is high.

For the present controlling power to the Peltier module, the PWM signals should be varied as per the control algorithm. The Peltier module is a non-linear device where there is a good possibility of controlling it using feedback loop structure. It is observed the generated heat or removed heat by the Peltier module is fast compared to other slow reacting devices, hence the Proportional – Integral (PI) algorithm is preferred which is easy and it is more than sufficient for this application [25–28]. Moreover, in this application the signal is maximum if the absolute difference between the set point and generated point is more than 4 degrees and only the PI algorithm triggers when the generated point is in between range of ± 4 degrees to the setpoint.

Considering the timing diagram of signalling the Power MOSFET driver from the datasheet, and to charge the bootstrap capacitor it is recommended to induce the voltage spikes in reverse direction than the considered main direction for heating or cooling for very short time in a cycle of the PWM so that the driver doesn't go to Under Voltage Lockout (UVLO). For more

explanation, if the insulated box requires to heat then considering Half-Bridge1 and Peltier module is generating temperature in the direction Q1-Q6 i.e., setting Molex pin3 high and Molex pin2 low for maximum time but set Molex pin2 high periodically only to charge the bootstrap capacitor but at that period the Molex pin3 should be set low. The main driving PWM signal is decided by the error ($|\text{Setpoint} - \text{generated point}|$) from the ESP32 while the width of PWM signals is decided by PI control algorithm only when the error is less than 4 degrees else maximum width of PWM signals are driven to the required pin of the Molex connector, while the bootstrap charging PWM is constant and would change the direction opposite to the main PWM signal. To avoid the short circuiting, it is recommended to avoid overlapping of signals in the main power driving signal and the bootstrap charging signal. As the present using Peltier module is more powerful than the required to the insulation box, we are using single H-bridge circuit for validating the control algorithm. More detailed PWM signalling is shown in the Image 23 below.

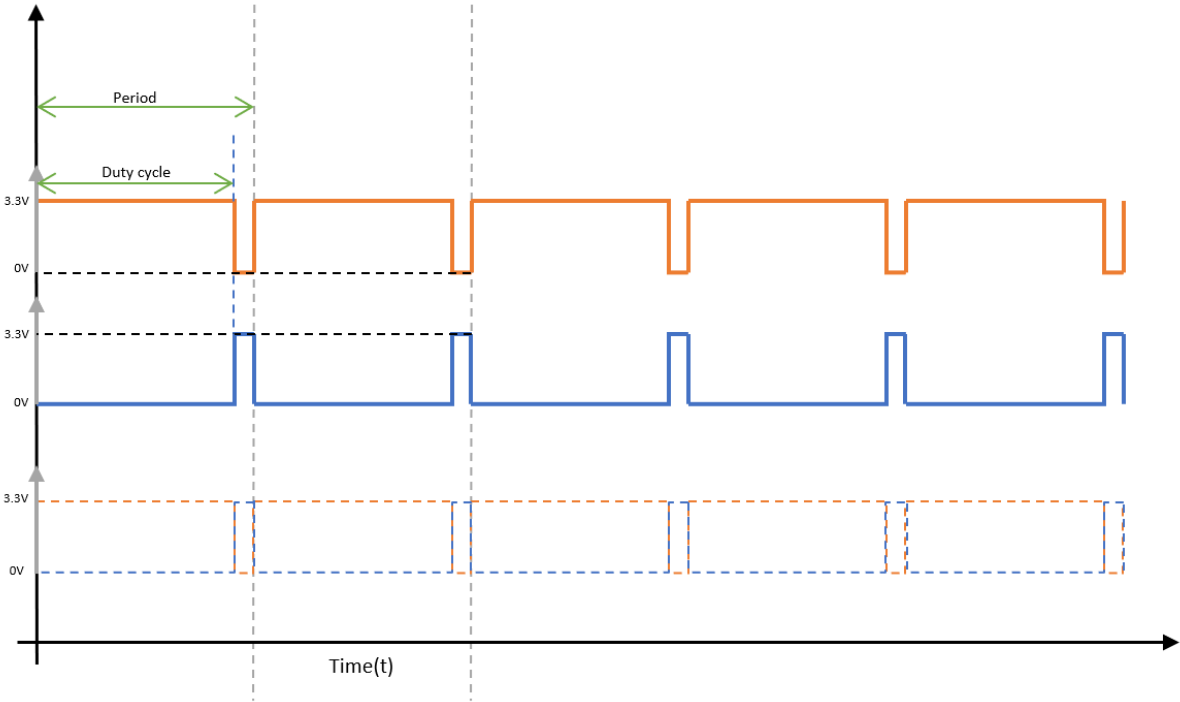


Image 23: PWM Signals

The main expected driving PWM signals from ESP32 which is shown in orange colour would be supplied in the required direction according to heating or cooling expectations from the user. The duty cycle of this main signal is decided by the PI algorithm and the width is modified accordingly. While the constant width blue signals are supplied to the opposite side of the H-bridge only to charge the bootstrap capacitor for normally working of the MOSFET driver. The main PWM signal is not overlapped with the secondary signal. For this application for

controlling the Peltier module, Motor Control Pulse Width Modulator (MCPWM) peripheral from ESP32 is used which is versatile PWM generator. By briefly checking the main and capacitor charging signals from ESP32, it is observed both signals should be properly fit in the unit period of the frequency without overlapping each other. In other view, the charging signal should be sent to the Molex pin which is opposite to the main power supply triggering Molex pin. By trial and analysis, it was decided to keep the frequency of generated signals from the ESP32 to be 1000 Hz and initial duty cycle of both signals are set to 0. The main signals are generated using timer0 while the charging signal is generated by timer1 and the charging signal is shifted by a required phase and synchronized to the main driving signal. The code for obtaining the signals is given. While the width of the main signal is modified using PI algorithm. The GPIO pin 2 and 4 are used to generate the signals. The testing box was equipped with a light sensor connected to the GPIO pin 12 which is used to detect if the box was closed or opened for starting the process of generating the required temperature.

4.2 Algorithm behaviour

The Peltier module can heat or cool according to the direction of current supplied in the H-bridge, the algorithm would trigger the responsible GPIO pins with PWM signals for heating or cooling the insulated box. The GPIO pins are triggered to maintain the difference between the generated temperature and the set temperature by the user to be zero, where the direction of signals is sent to the responsible GPIO pins to heat or cool for reaching the desired temperature or to maintain the difference zero if there is an error. To make the box robust, the maximum duty cycle signals are signalled in particular GPIO pins to reach the respective set temperature until the absolute error is more than 4 degrees which is selected by trial and, else the PWM signals are generated as per the PI algorithm. It is observed that by keeping range of ± 4 degree as control ranges from the set point, the control algorithm could engage and modify the signalling PWM signals to reach and maintain setpoint, else due to the full Peltier operation, the temperature was over shooting and pushing out of control range hence ± 4 degrees from setpoint was selected for triggering the PI control algorithm. The control algorithm works and controls the Peltier module only if the box is closed, to detect the closed box a light sensor is placed inside edge of the box and connected to the digital GPIO pin 12. The following flowchart from Image 24 would describe the flow of control in the ESP32.

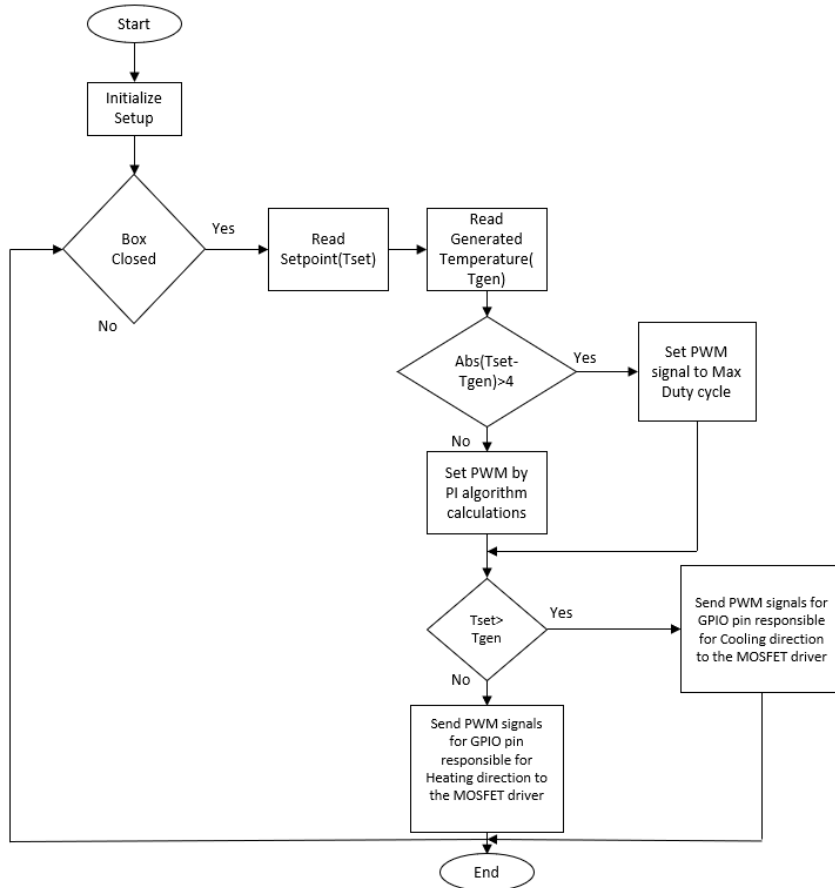


Image 24: Control Flowchart.

```

#include "driver/mcpwm.h"
#include "soc/mcpwm_periph.h"

#define GPIO_PWM0A_OUT 2 // Set GPIO 15 as PWM0A
#define GPIO_PWM1B_OUT 4 // Set GPIO 16 as PWM0B
#define GPIO_FAN_OUT 13
#define ONE_WIRE_BUS 15

float n; // Control signal value
float tgen, tgen2; // Generated temperatures
float tset, lastTset; // Set temperature values
float error, P, I; // Error, Proportional, and Integral terms
float Pconst = 0.45; // Proportional constant
float Iconst = 0.0045; // Integral constant

// Function to initialize MCPWM GPIO pins
static void mcpwm_example_gpio_initialize(void) {
    // GPIO initialization
    mcpwm_gpio_init(MCPWM_UNIT_0, MCPWM0A, GPIO_PWM0A_OUT);
    mcpwm_gpio_init(MCPWM_UNIT_0, MCPWM1B, GPIO_PWM1B_OUT);
}

// Function to set Peltier module duty cycle based on control signal
static void peltier_set(float duty_cycle) {
    // Setting signals low initially
    mcpwm_set_signal_low(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_B);
    mcpwm_set_signal_low(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_A);
}
  
```

```

if (duty_cycle > 0) {
    // Enabling synchronization for positive duty cycle
    mcpwm_sync_enable(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_SELECT_TIMER0_SYNC,
10);

    // Setting duty cycles for PWM signals
    mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A, duty_cycle);
    mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A,
MCPWM_DUTY_MODE_0);

    mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_B, 0.1);
    mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_B,
MCPWM_DUTY_MODE_0);
} else {
    // Enabling synchronization for negative duty cycle
    mcpwm_sync_enable(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_SELECT_TIMER0_SYNC,
990);

    // Setting duty cycles for PWM signals
    mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A, 0.1);
    mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A,
MCPWM_DUTY_MODE_0);

    mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_B, -(duty_cycle));
    mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_B,
MCPWM_DUTY_MODE_0);
}
}

// Function to map error to control signal range
void maprange() {
    // Input and output ranges for mapping
    float in_max = 4;
    float in_min = -in_max;
    float out_max = 98;
    float out_min = -out_max;

    // Calculating error
    float error = (tgen - tset);

    if (error <= in_max && error >= in_min) {
        // Calculating P and I terms
        float x;
        P = float(tgen - tset) * Pconst;
        I += float(tgen - tset) * Iconst;
        x = P + I;

        // Mapping control signal to output range
        n = (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
    }
    // Handling error outside input range
    if (error > in_max) {
        n = out_max;
        I = 0; // Clearing integration part of regulator to eliminate wind-up
effect
    }
    if (error < in_min) {
        n = out_min;
        I = 0; // Clearing integration part of regulator to eliminate wind-up
effect
    }
}

```

```

}
// Handling zero error
if (error == 0) {
    n = 0;
}
}

// Function to update generated temperature values
void update_gentemp() {
    // Updating temperatures from sensors
    // Note: 'sensors' is assumed to be a global object of some temperature sensor
    library
    // This section of code is not provided in the snippet
    float t1 = sensors.getTempCByIndex(0);
    float t2 = sensors.getTempCByIndex(1);

    // Printing debug information
    Serial.print("n:");
    Serial.print(n);
    Serial.print(" t1:");
    Serial.print(t1);
    Serial.print(" t2:");
    Serial.print(t2);
    Serial.print(" tset:");
    Serial.print(tset, 2);
    Serial.print(" tgen:");
    Serial.print(tgen, 2);
    Serial.print(" Preg:");
    Serial.print(P, 4);
    Serial.print(" Ireg:");
    Serial.println(I, 5);
}

// Function to initialize PWM configuration
void PWM_init() {
    // Initializing GPIO pins for MCPWM
    mcpwm_example_gpio_initialize();

    // Configuring initial parameters for MCPWM
    mcpwm_config_t pwm_config;
    pwm_config.frequency = 1000; // Frequency = 1000Hz
    pwm_config.cmpr_a = 0; // Duty cycle of PWMxA = 0
    pwm_config.cmpr_b = 0; // Duty cycle of PWMxB = 0
    pwm_config.counter_mode = MCPWM_UP_COUNTER;
    pwm_config.duty_mode = MCPWM_DUTY_MODE_0;

    // Initializing MCPWM with the above settings
    mcpwm_init(MCPWM_UNIT_0, MCPWM_TIMER_0, &pwm_config);
    mcpwm_init(MCPWM_UNIT_0, MCPWM_TIMER_1, &pwm_config);
    mcpwm_sync_enable(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_SELECT_TIMER0_SYNC, 0);
    mcpwm_sync_enable(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_SELECT_TIMER0_SYNC, 30);
    mcpwm_set_timer_sync_output(MCPWM_UNIT_0, MCPWM_TIMER_0,
MCPWM_SWSYNC_SOURCE_TZ);
    float dutyC = 1.0;
    mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A, dutyC);
    mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_B, dutyC);
}

// Arduino setup function
void setup() {
    Serial.begin(9600); // Initializing serial communication

```

```

PWM_init();          // Initializing PWM configuration
delay(500);          // Delay for 500ms
pinMode(12, INPUT_PULLUP); // connected to light sensor for detecting if the
box
                        is closed or opened.
}

// Arduino loop function
void loop() {
  if(digitalRead(12)==HIGH){
    update_gentemp(); // Updating generated temperatures
    maprange();       // Mapping error to control signal
    peltier_set(n);   // Setting Peltier module duty cycle
  }
}

```

Source Code 5: Control Algorithm code.

As ESP32 has dual core processing capabilities, the task of transferring and receiving the Tset and Tgen variable without interfering the control algorithm executed in core 1, a secondary task with normal priority is created to execute the UART communication in core 0.

```

HardwareSerial SerialPort(2);

TaskHandle_t Task1;

float tgen, tset;
unsigned long lastTime = 0;
const unsigned long interval = 1000;

void Task1code(void* parameter) {
  for (;;) {
    if (SerialPort.available()) {
      tset = SerialPort.parseFloat();
    }

    // Send data back on the same UART port
    unsigned long currentTime = millis();
    if (currentTime - lastTime >= interval) {
      String tgen_string = String(tgen);
      SerialPort.write("S");
      SerialPort.write(tgen_string.c_str());
      lastTime = currentTime;
    }
  }
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  SerialPort.begin(9600, SERIAL_8N1, 16, 17);
  xTaskCreatePinnedToCore(
    Task1code, /* Function to implement the task */
    "Task1", /* Name of the task */
    10000, /* Stack size in words */
    NULL, /* Task input parameter */
    0, /* Priority */
    1); /* Core to run on (0 or 1) */
}

```

```

    0,      /* Priority of the task */
    &Task1, /* Task handle. */
    0);

}

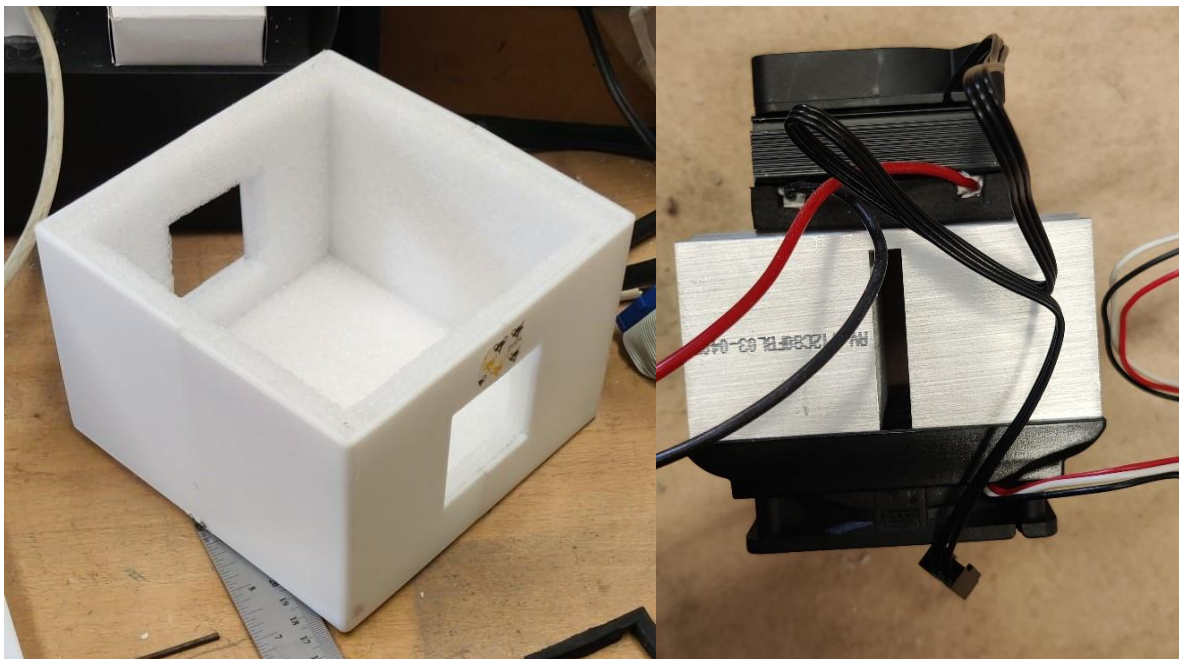
void loop() {
    // put your main code here, to run repeatedly:
    delay(100);
}

```

Source Code 6: Multicore processing code.

4.3 Test Setup

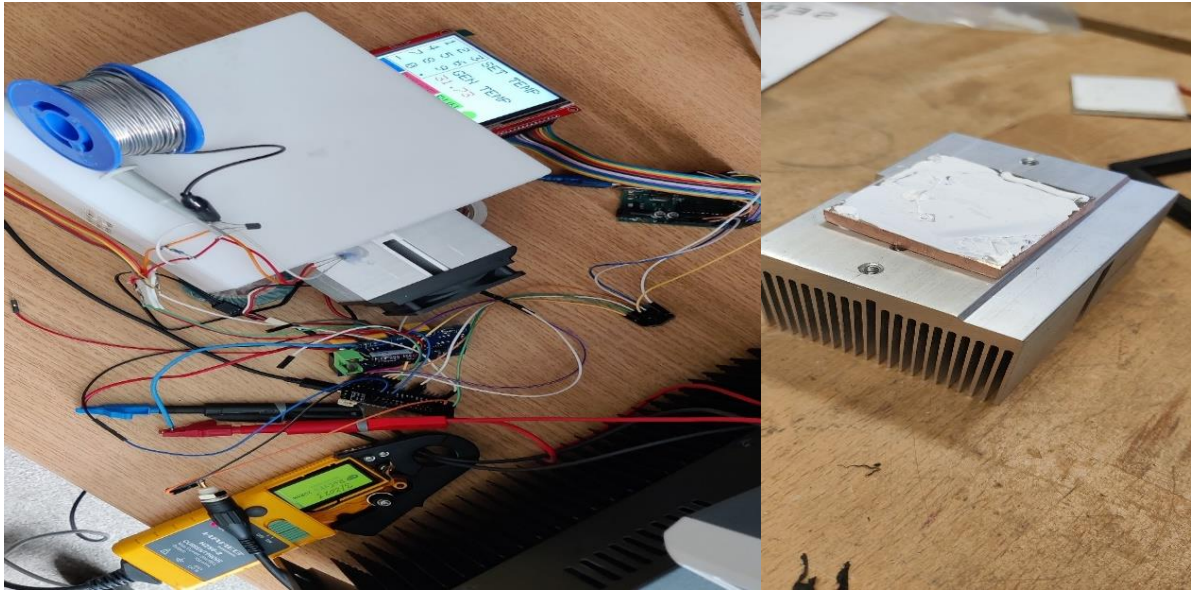
The prototype box is designed to hold 2 Peltier modules but for experimenting the controller, only one Peltier module is used. The designed box is 3-d printed and inside is lined with an insulation foam to trap the temperature as shown in the Image 25(a). The Peltier module is then fitted with fans on both the sides and tightened with nylon bolts to reduce the thermal short circuiting, it is shown in Image 25(b). The electronics are then connected to the box and powered for testing as shown in Image 26(a) initially and further a hardware integrating box was designed and printed as shown in the Image 27(b). For testing the control algorithm only single Peltier module was used and for constant reading from the insulation box a glass of water is placed. The power source of 120W is supplied to the power circuit in experimenting the controller.



(a)

(b)

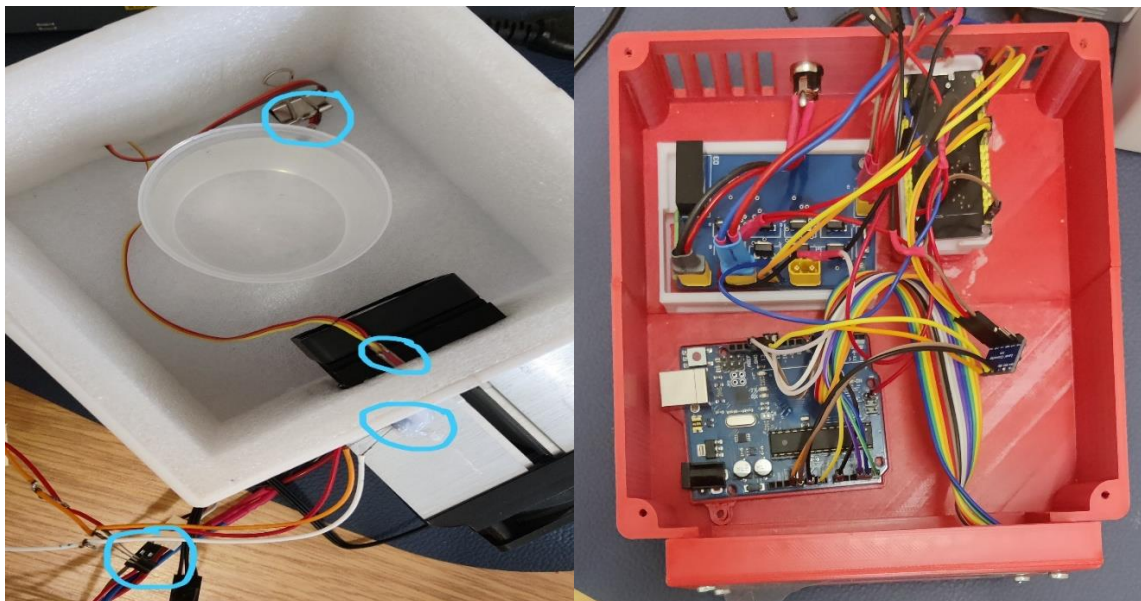
Image 25: (a)3D Printed Box, (b)Integrated Peltier module and Heatsink



(a)

(b)

Image 26: (a)Test Setup, (b)Heat Sink.



(a)

(b)

Image 27: (a)Sensor integration for testing, (b)Hardware Integration.

4.4 PI Tuning

The Proportional-Integral-Derivative (PID) algorithm [26, 27] is most popularly used control algorithm in various application which has feedback control mechanism. The aim of this

algorithm is to control the output as per adjusting the input setpoint based on the error between the desired output and actual process variable. This algorithm consists of three terms responsible for controlling the input to obtain a balance between responsiveness, stability and steady-state accuracy. The features of each term used is explained.

- Proportional Term(P)

This term is directly linked to the current error of the system which is difference between the expected output to the existing output. The error affecting by the system would directly proportional to this term. This term is responsible to throttle the system input to obtain the output near to the desired output. The higher set value for this term would increase overshoot and oscillations in the system. The system stability is determined by the oscillations of output are generated by assigning value to this term. The robustness of the system is obtained by considering this term under stability while tuning.

- Integral Term(I)

This term is used to overcome the dynamic losses like friction, disturbances or noise of the system, where the past errors are cumulative added over time and eliminated the remaining offset between the setpoint and the actual process variable. This term reduces the response of the system and decreases the oscillations.

- Derivative Term(D)

This term is used to reduce the change in error by addressing the change of errors in every sampling point and would push the output of the system which dampens the overshoot. But this term may amplify noise and can cause the system unstable, hence it is recommended to use for the system which are reacts very slow by nature.

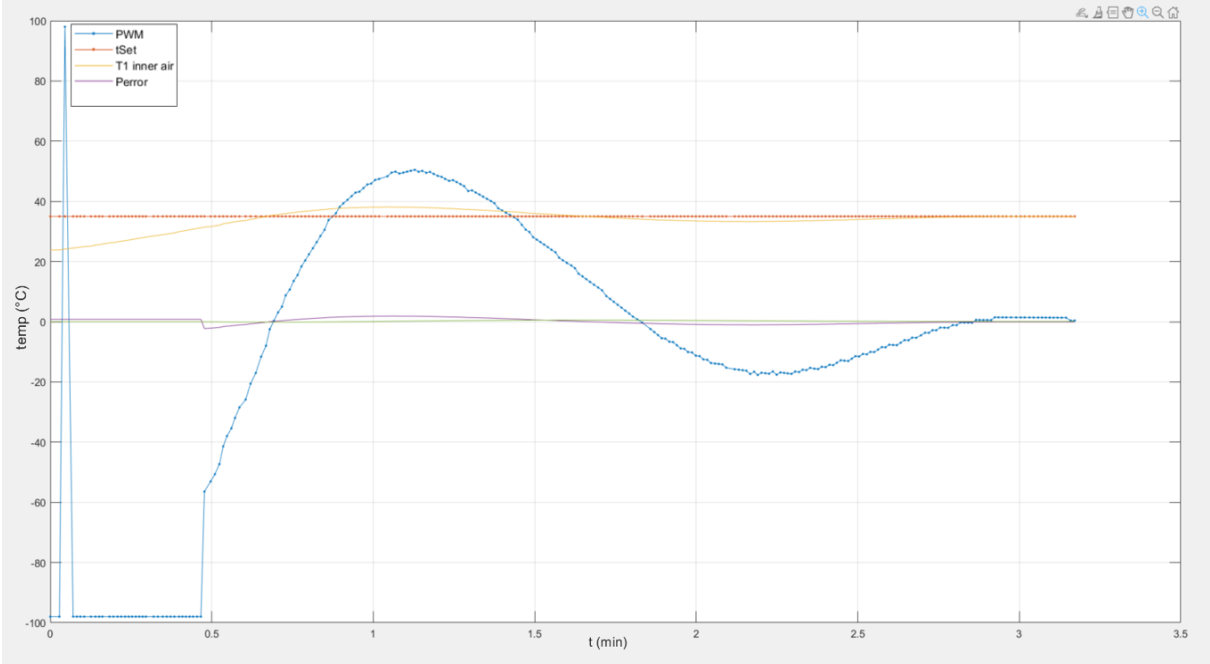
The overall control PID equation is as below.

$$u(t) = Kp e(t) + Ki \int_0^t e(t)dt + Kd \frac{de(t)}{dt} \quad (41)$$

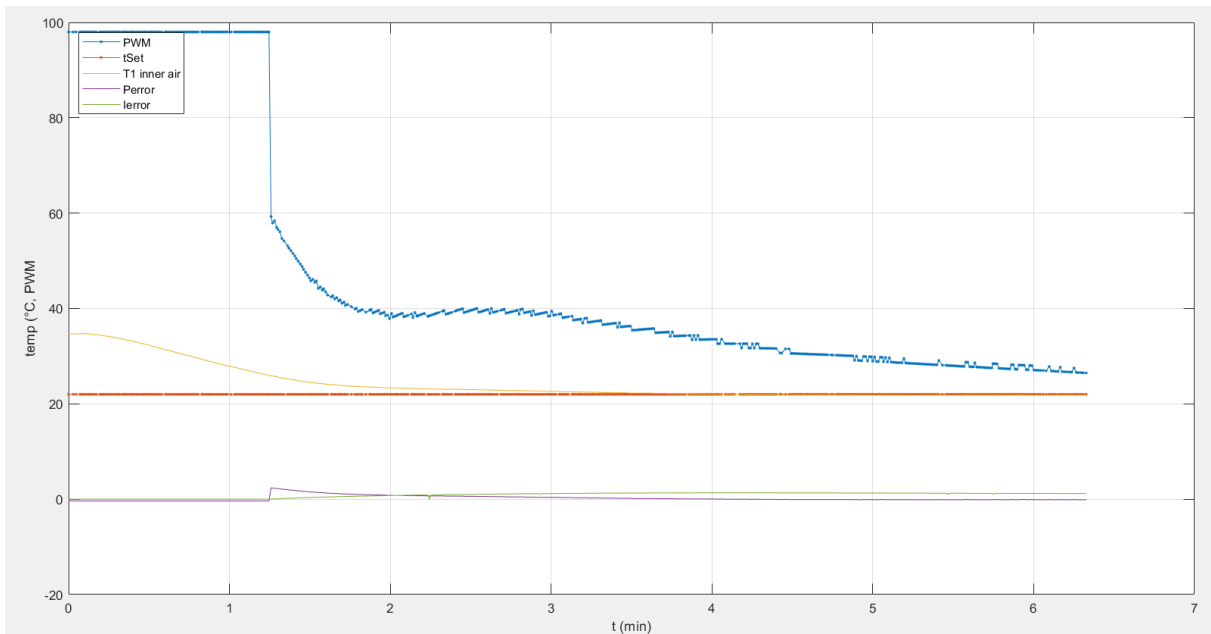
Like explained in the control flowchart, the controller is combined of only PI algorithm based PWM when the range of error is in between +-4 °C degrees else full PWM [29] is supplied to the Peltier module considering the direction to supply. By performing trial and error method, a

stable value for $P = 0.6$ and $I = 0.0040$ coefficients are found with minimal overshoot and reducing steady state error.

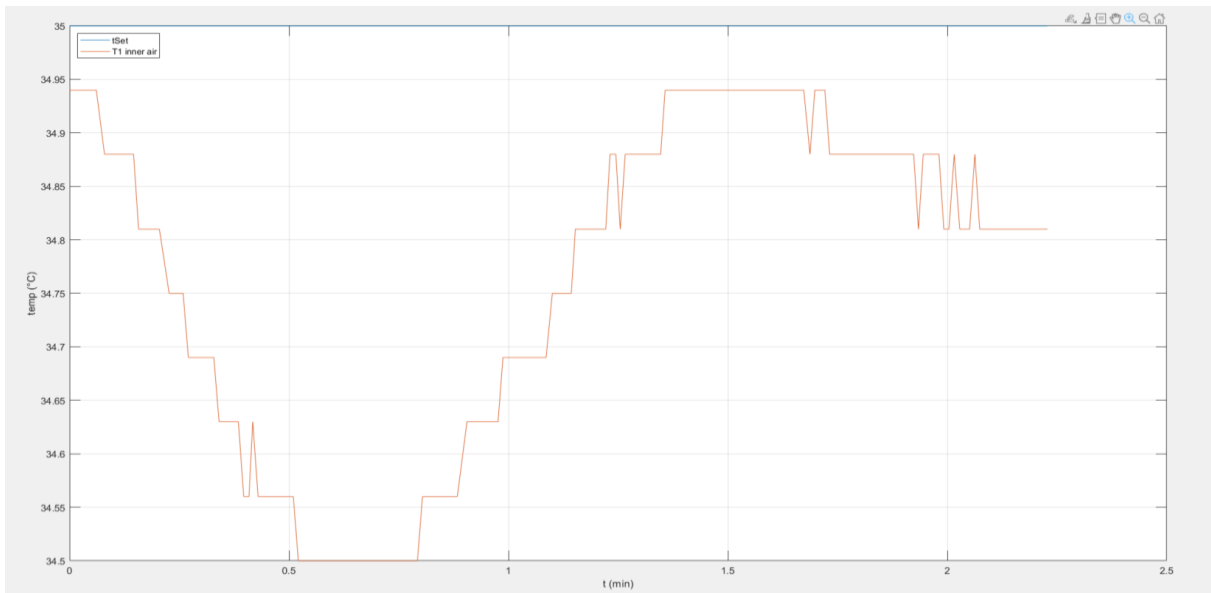
The behaviour of control algorithm is plotted in the Graph 1, where from initial box temperature of $23\text{ }^{\circ}\text{C}$ to $35\text{ }^{\circ}\text{C}$ was generated with over shoot of $2.7\text{ }^{\circ}\text{C}$ was observed and parameters like PWM, Proportional error and Integrating error is plotted against time in seconds. It is clearly observed that the box can reach to the stable setpoint within 180 seconds and bit fluctuations from sensor reading of $-0.5\text{ }^{\circ}\text{C}$ is seen from Graph 3, where the cause may be of floating-point calculations. In the Graph 2, the initial box temperature from $35\text{ }^{\circ}\text{C}$ is brought to $22\text{ }^{\circ}\text{C}$, where no overshoot is observed. The change in duty cycle in the powering PWM signal is observed when the threshold range of error $\pm 4\text{ }^{\circ}\text{C}$ from the setpoint, outside the threshold error, it is observed the change in the sign of maximum duty cycle is seen where the power signals are reversed in GPIO pin.



Graph 1: Setpoint temperature(tset), PWM, Generated Temperature(T1) VS time.



Graph 2: Setpoint temperature(tset), PWM, Generated Temperature(T1) VS time.

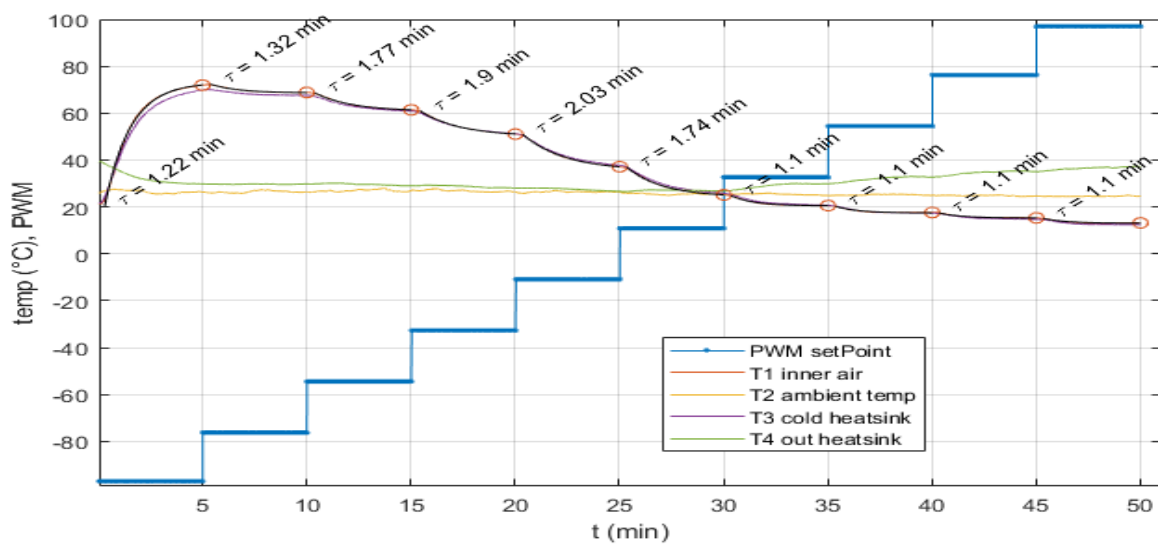


Graph 3: Error in regulation.

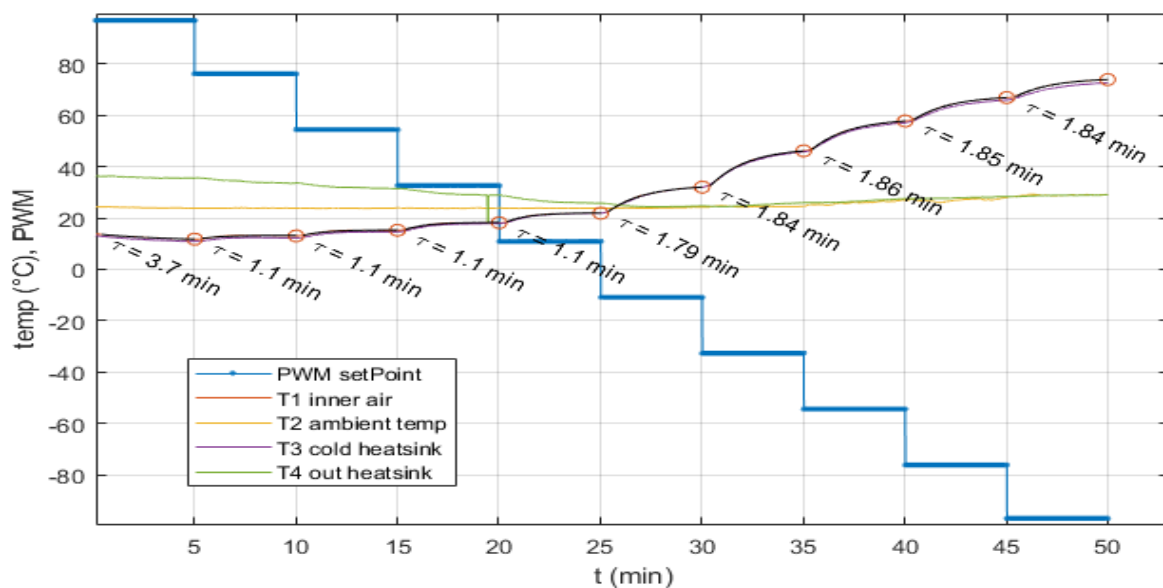
The accuracy of the DS18B20+ sensor is $\pm 0.5\text{ }^{\circ}\text{C}$ and the PI calculations are in float variables with 4 decimal points which may lead to have hardware errors in generating precise power delivery to the Peltier module. The fluctuations in reading the generated temperature are shown in the Graph 3.

5 Results and conclusion.

The initial test on the box was done to check if the box was insulated properly to hold the generated temperature and powered by 120 W (14.5 V) power supply unit. The box was subjected to change in the duty cycle of the signals generated from the range of -98(-sign indicates change in direction of signalling) to 98 in difference of 5% duty cycle for every 5 minutes and then the duty cycle was reversed from 98 to -98. To study the detail change in temperature changes additional DS18B20+ was integrated on each surface on the heat sink both inside and outside with a DS18B20+ and to know inside box and outside air temperature two other sensors are integrated as shown in the Image 27(a).



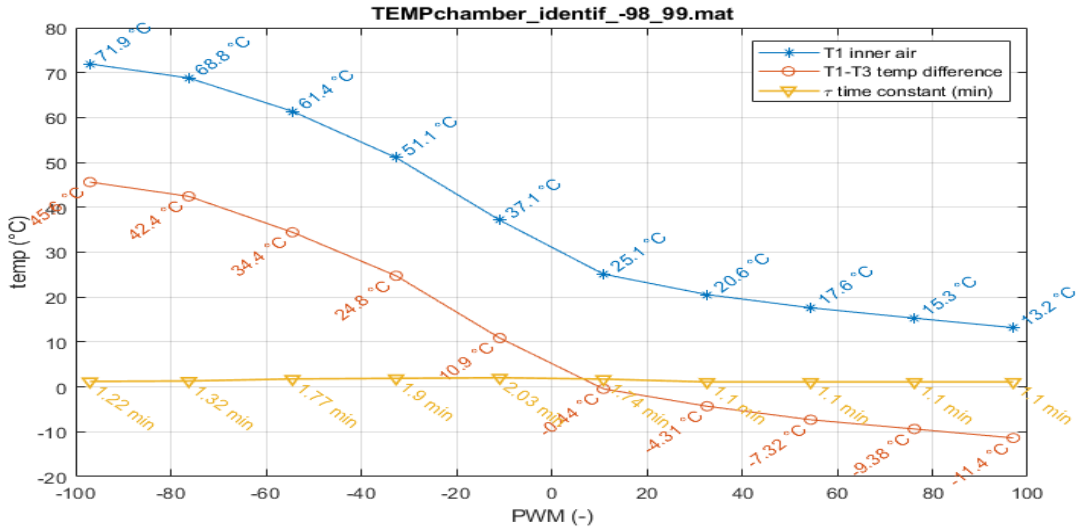
(a)



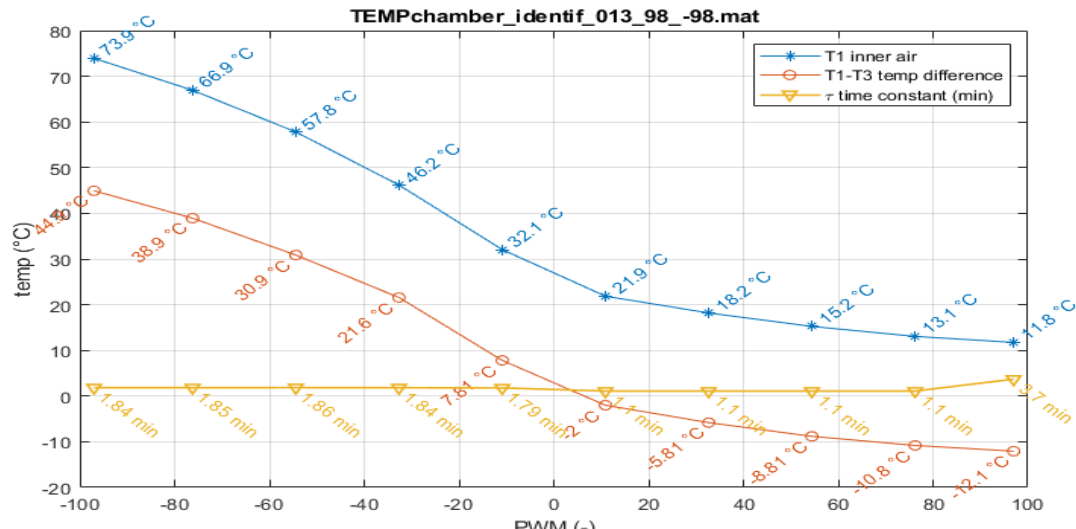
(b)

Graph 4: Temperature VS PWM (a)-98% to 98%, (b) 98% to -98%.

The Graph 4 (a) and (b) shows the reading of the insulated box in change of PWM duty cycle from -98% to 98% and 98% to -98%. Further the data is processed to find the maximum and minimum reaching temperature reached inside insulation box when the PWM is signalled both the direction. The box reached maximum temperature of 71.9 °C and minimum of 13.2 °C from -98% to 98% duty cycle as shown in the Graph 5(a) and initial of the change in the outside heat sink temperature(T3) and inner air temperature(T1) is shown with the time (T -indicated in yellow line) taken to reach when the start of every 5 minute of applied PWM is seen. When the box is subjected to change in 98% to -98% duty cycle starting from atmospheric temperature inside the box, it is found the box could reach maximum of 73.9 °C and minimum of 11.8 °C as shown in the Graph 5(b).



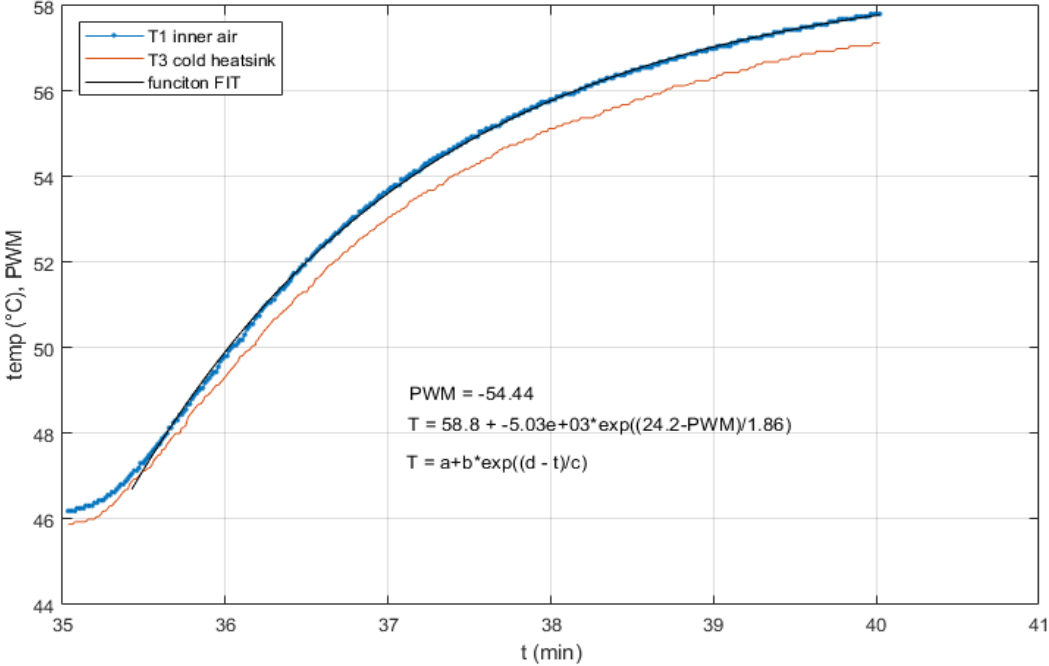
(a)



(b)

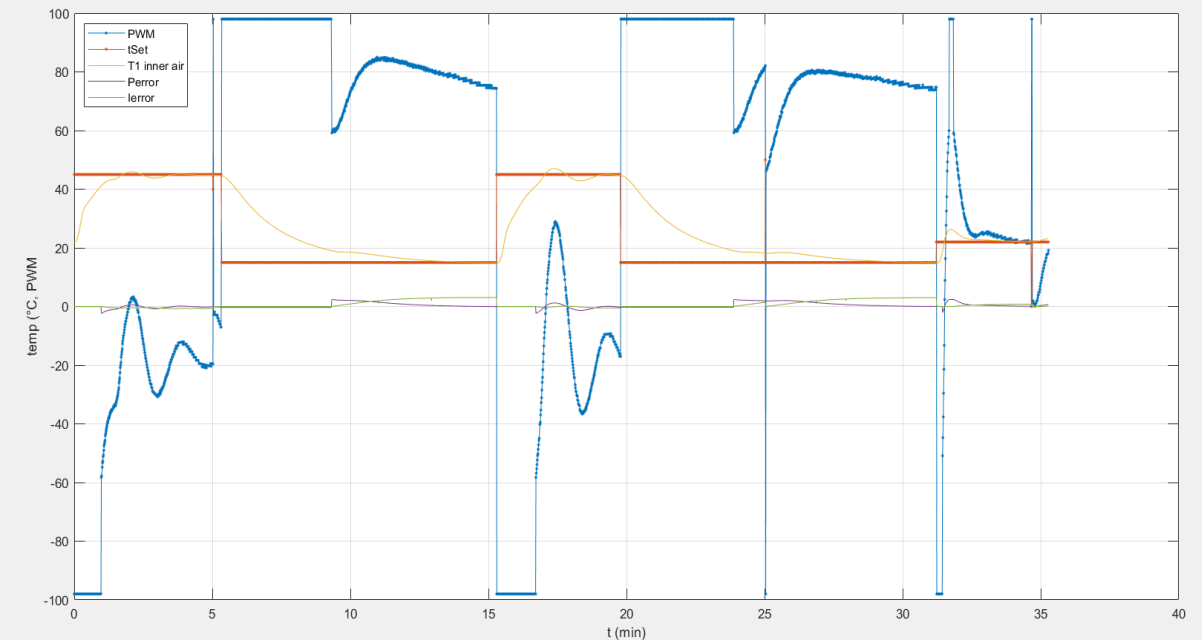
Graph 5: T1, (T1-T3) VS PWM(a) -98% to 98% duty cycle, (b)98% to -98% duty cycle.

The polynomial equation of the generated temperature inside the insulation box is found as shown in the Graph 6.



Graph 6: Function fit.

To check the dynamic behaviour of the box, a set points 22 °C, 45 °C, 15 °C, 45 °C, 25 °C is given from user interface and the PWM change with change in generated temperature is plotted.



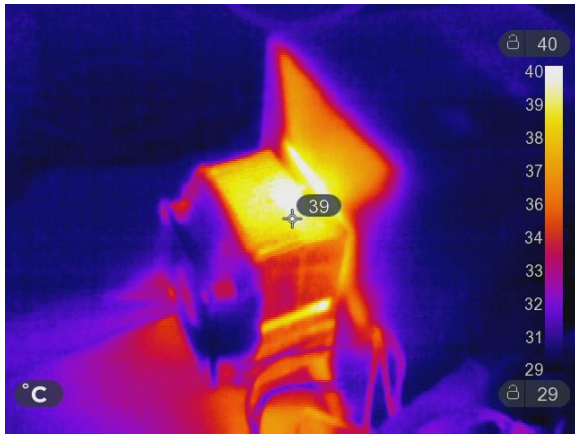
Graph 7: Dynamic behaviour of the box.

By observing the dynamic behaviour as shown in the Graph 7, it is observed the generated temperature can meet the set temperature with 3.5 minutes with an overshoot up to 2-3 °C when

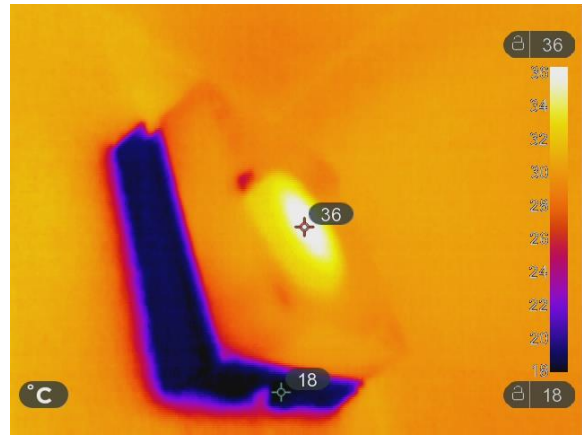
the setpoint temperature is greater than the generated temperature, when the box is subjected for cooling, there was no undershoot observed and the generated temperature would glaze over the setpoint temperature within 7-8 minutes depending the setpoint. The atmospheric temperature also influences the generation of temperature inside box, as the maximum change in temperature across the Peltier module is constant. It is also observed the temperature generation inside the box is very rapid when the box was previously cooled as the temperature in the heatsink would be more suitable to dissipate the temperature.

The heatsink was not fully sufficient to dissipate the heat generated when the box was subjected to cooling, the box could reach up to 8 °C and no more change in temperature is seen. When the system was subjected to 240 W of power supply, the system could reach the setpoint very quick but the existing tuning parameters was not suitable, there was a huge overshoot and undershoot. While the maximum cooling temperature was up to 13 °C as the maximum generated change in temperature by the Peltier module was crossing 68 °C, hence big heatsink is required.

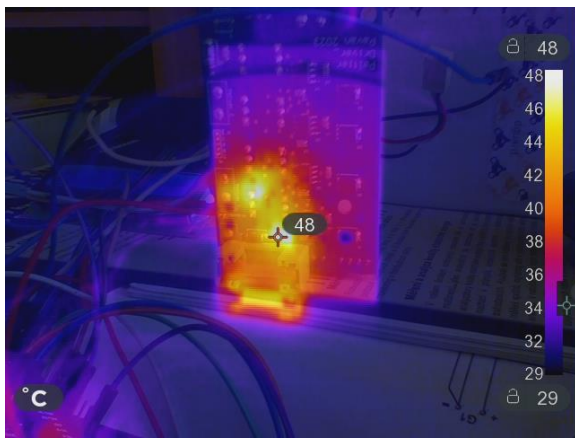
The insulating of the Peltier module also plays significant role in not causing thermal short-circuit among the two faces when subjected to generate change in temperatures. It was observed the internal ventilating motor was generating heat inside as shown in Image 28(b) and in external side the 3D printed box had wide spread of generated heat as shown in Image 28(a). The power circuit also generates the heat as shown in the Image 28(c) and would also affect the box if its integrated.



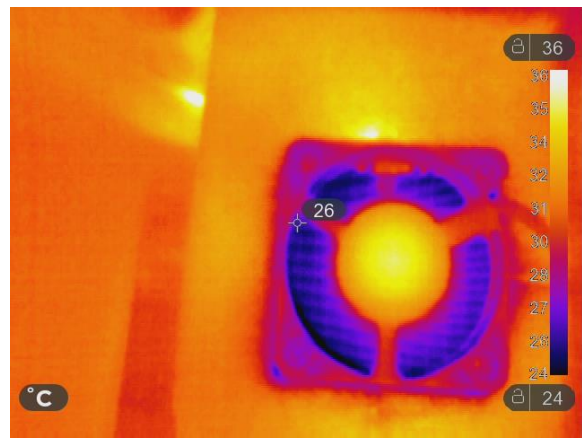
(a)



(b)



(c)



(d)

Image 28: Thermal camera images (a)External side heatsink, (b)Internal side heatsink, (c) Power circuit & (d)Internal side heatsink.

5.1 Conclusion

The detailed study of controllers is done over the period of this project with a clear understanding of components used in the existing available Thermoelectric controllers and its limits in application. The required approaches of controlling the Peltier module are studied and suitable schematics and design of the power circuit for powering two Peltier module is analysed and suitable components are selected to manufacture a PCB for manufacturing. The power circuit was fabricated as expected with the components, some minor modifications were done like adding capacitor for keeping constant power supply and a Schottky diode of 5 V connected to the gate and source of the power MOSFET responsible for controlling heatsink fan.

The prototype for integrating the Peltier module was designed and 3-D printed, thermal insulation was done to reduce the interference of external atmospheric temperature to the inside generated temperature from the Peltier module. The precision of the temperature sensor was seen

while regulating the insulation box. The controlling PWM signals were generated as per the expectations and firmware was programmed for generating PWM signals driving the power circuit. The tuning of the system is performed by switching between heating or cooling as per the error. The dynamic behaviour of the Insulation box was studied and understood then behaviour of the Peltier module which was not linear to control. Further the experiment can be addressed more efficiently and optimised control algorithm with a proper insulation and better heat sink with integrating the power source and microcontroller inside the power circuit.

6 Citations

- [1] VIMALAVATHINI, R a B GITANJALI. Effect of temperature on the potency & pharmacological action of insulin. *The Indian journal of medical research*. 2009, **130**, 166–9.
- [2] YANG, Lei, Zhi-Gang CHEN, Matthew S. DARGUSCH a Jin ZOU. High Performance Thermoelectric Materials: Progress and Their Applications. *Advanced Energy Materials* [online]. 2018, **8**(6), 1701797. ISSN 1614-6840. Dostupné z: doi:10.1002/aenm.201701797
- [3] BOS, Jan-Willem G. 2.3 - Recent developments in half-Heusler thermoelectric materials. In: Ryoji FUNAHASHI, ed. *Thermoelectric Energy Conversion* [online]. B.m.: Woodhead Publishing, 2021 [vid. 2023-05-03], Woodhead Publishing Series in Electronic and Optical Materials, s. 125–142. ISBN 978-0-12-818535-3. Dostupné z: doi:10.1016/B978-0-12-818535-3.00014-1
- [4] PARASHCHUK, Taras, N. SIDORENKO, L. IVANTSOV, A. SOROKIN, M. MAKSYMUK, Bogdan DZUNDZA a Zinovi DASHEVSKY. Development of a solid-state multi-stage thermoelectric cooler. *Journal of Power Sources* [online]. 2021, **496**, 229821. Dostupné z: doi:10.1016/j.jpowsour.2021.229821
- [5] *Thermoelectric Coolers / The World Leader in Thermal Management Solutions* [online]. [vid. 2023-05-03]. Dostupné z: <https://lairdthermal.com/products/thermoelectric-cooler-modules>
- [6] KOSTYUK, Olena, Bogdan DZUNDZA, Ya.S. YAVORSKY a Zinovi DASHEVSKY. Development of Thermal Detector Based on Flexible Film Thermoelectric Module. *Physics and Chemistry of Solid State* [online]. 2021, **22**, 45–52. Dostupné z: doi:10.15330/pcss.22.1.45-52
- [7] *Liquid Series - Liquid heat exchanger peltier cooler assemblies* [online]. [vid. 2023-05-03]. Dostupné z: <https://lairdthermal.com/products/thermoelectric-cooler-assemblies/peltier-liquid-series>
- [8] CARMO, João, L.M. GONCALVES a J.H. CORREIA. Micro and Nanodevices for Thermoelectric Converters. In: [online]. 2011, s. 791–812. ISBN 978-3-642-10496-1. Dostupné z: doi:10.1007/978-3-642-10497-8_25
- [9] DZUNDZA, Bogdan, Olena KOSTYUK a Zinovi DASHEVSKY. Features of Designing Computer Control Systems for Precision Thermoelectric Coolers. *Physics and Chemistry of Solid State* [online]. 2021, **22**, 278–283. Dostupné z: doi:10.15330/pcss.22.2.278-283

- [10] CASTAÑEDA, Manuela, Elkin GUTIÉRREZ-VELÁSQUEZ, C. AGUILAR, Sergio MONTEIRO, Andrés AMELL a Henry COLORADO. Sustainability and Circular Economy Perspectives of Materials for Thermoelectric Modules. *Sustainability* [online]. 2022, **14**, 5987. Dostupné z: doi:10.3390/su14105987
- [11] MAMUR, Hayati, Ömer DILMAÇ, Jahanara BEGUM a Mohammad BHUIYAN. Thermoelectric generators act as renewable energy sources. *Cleaner Materials* [online]. 2021, **2**, 100030. Dostupné z: doi:10.1016/j.clema.2021.100030
- [12] WITTING, Ian T., Thomas C. CHASAPIS, Francesco RICCI, Matthew PETERS, Nicholas A. HEINZ, Geoffroy HAUTIER a G. Jeffrey SNYDER. The Thermoelectric Properties of Bismuth Telluride. *Advanced Electronic Materials* [online]. 2019, **5**(6), 1800904. ISSN 2199-160X, 2199-160X. Dostupné z: doi:10.1002/aelm.201800904
- [13] NOZARIASBMARZ, Amin, Jerzy S. KRASINSKI a Daryoosh VASHAEE. N-Type Bismuth Telluride Nanocomposite Materials Optimization for Thermoelectric Generators in Wearable Applications. *Materials* [online]. 2019, **12**(9), 1529. ISSN 1996-1944. Dostupné z: doi:10.3390/ma12091529
- [14] LALONDE, Aaron D., Yanzhong PEI, Heng WANG a G. JEFFREY SNYDER. Lead telluride alloy thermoelectrics. *Materials Today* [online]. 2011, **14**(11), 526–532. ISSN 1369-7021. Dostupné z: doi:10.1016/S1369-7021(11)70278-4
- [15] RAVINDRA, Nuggehalli, Bhakti JARIWALA, Asahel BANOBRE a Aniket MASKE. Thermoelectric Properties of Silicon-Germanium Alloys: Fundamentals, Materials Selection, Properties, and Performance. In: [online]. 2019, s. 49–67. ISBN 978-3-319-96339-6. Dostupné z: doi:10.1007/978-3-319-96341-9_4
- [16] MASKE, Aniket Annasaheb. Thermoelectric properties of silicon-germanium alloys. nedatováno.
- [17] UHER, Ctirad. 2.2 - Thermoelectric properties of skutterudites. In: Ryoji FUNAHASHI, ed. *Thermoelectric Energy Conversion* [online]. B.m.: Woodhead Publishing, 2021 [vid. 2023-05-03], Woodhead Publishing Series in Electronic and Optical Materials, s. 69–123. ISBN 978-0-12-818535-3. Dostupné z: doi:10.1016/B978-0-12-818535-3.00013-X
- [18] KAREGOUDRA, Shilpa, Md Abdul RAHEMAN, Abdul AABID, Muneer BAIG, Veerasha RK a Nagesh KUDVA. A Systematic Review of Thermoelectric Peltier Devices: Applications and Limitations. *Fluid Dynamics and Materials Processing* [online]. 2022, 1–20. Dostupné z: doi:10.32604/fdmp.2022.020351
- [19] PATIDAR, Shyam. Applications of Thermoelectric Energy: A Review. *International Journal for Research in Applied Science and Engineering Technology* [online]. 2018, **6**, 1992–1996. Dostupné z: doi:10.22214/ijraset.2018.5325
- [20] JOSE, Allwin, Alan D'SOUZA, Sarvesh DANDEKAR, Jitesh KARAMCHANDANI a Pavan KULKARNI. Air conditioner using Peltier module. In: *2015 International Conference on Technologies for Sustainable Development (ICTSD): 2015 International Conference on Technologies for Sustainable Development (ICTSD)* [online]. 2015, s. 1–4. Dostupné z: doi:10.1109/ICTSD.2015.7095879

- [21] LUCAS, Stephen, Romeo MARIAN, Michael LUCAS, Titi OGUNWA a Javaan CHAHL. Employing the Peltier Effect to Control Motor Operating Temperatures. *Energies* [online]. 2023, **16**, 2498. Dostupné z: doi:10.3390/en16052498
- [22] HUANG, B. J, C. J CHIN a C. L DUANG. A design method of thermoelectric cooler. *International Journal of Refrigeration* [online]. 2000, **23**(3), 208–218. ISSN 0140-7007. Dostupné z: doi:10.1016/S0140-7007(99)00046-8
- [23] Interface Multiple DS18B20s with ESP32 & Display Values on Web Server. *Last Minute Engineers* [online]. 31. prosinec 2018 [vid. 2023-08-07]. Dostupné z: <https://last-minuteengineers.com/multiple-ds18b20-esp32-web-server-tutorial/>
- [24] *ESP32 DS18B20 Temperature Sensor with Arduino IDE (Single, Multiple, Web Server) / Random Nerd Tutorials* [online]. 14. červenec 2019 [vid. 2023-08-07]. Dostupné z: <https://randomnerdtutorials.com/esp32-ds18b20-temperature-arduino-ide/>
- [25] BANSAL, P. K. a A. MARTIN. Comparative study of vapour compression, thermoelectric and absorption refrigerators. *International Journal of Energy Research* [online]. 2000, **24**(2), 93–107. ISSN 1099-114X. Dostupné z: doi:10.1002/(SICI)1099-114X(200002)24:2<93::AID-ER563>3.0.CO;2-6
- [26] RAO, K Smriti a Ravi MISHRA. Comparative study of P, PI and PID controller for speed control of VSI-fed induction motor. 2014, **2**(2).
- [27] SALEM, Fawzan, Mohamed I. MOSSAD a M. AWADALLAH. A comparative study of MPC and optimised PID control. *International Journal of Industrial Electronics and Drives* [online]. 2015, **2**, 242. Dostupné z: doi:10.1504/IJIED.2015.076293
- [28] EFHEIJ, Hafed, Abdulgani ALBAGUL a Nabela ALBRAIKI. *Comparison of Model Predictive Control and PID Controller in Real Time Process Control System* [online]. 2019. Dostupné z: doi:10.1109/STA.2019.8717271
- [29] PANG, D., Won-Suk JEON, Kwang-Hoon CHOI, T. KWON, Nam-Gyun KIM a Seong-Cheol LEE. Temperature Control using Peltier Element by PWM Method. In: [online]. 2005 [vid. 2023-04-14]. Dostupné z: <https://www.semanticscholar.org/paper/Temperature-Control-using-Peltier-Element-by-PWM-Pang-Jeon/f83673755d07c5947235f6f57ad-caf1cb0d7cd0f>
- [30] *Bi-Directional Thermostatic Controller* [online]. [vid. 2023-04-20]. Dostupné z: <https://lairdthermal.com/products/temperature-controllers/bi-directional-thermostatic-controllers>
- [31] Choosing and Using Advanced Peltier Modules for Thermoelectric Cooling. *Digi-Key Electronics* [online]. [vid. 2023-04-20]. Dostupné z: <https://www.digikey.com/en/articles/choosing-using-advanced-peltier-modules-thermoelectric-cooling>
- [32] *How to achieve temperature control with a peltier module system* [online]. [vid. 2023-04-20]. Dostupné z: <https://www.electronicsspecifier.com/products/test-and-measurement/how-to-achieve-temperature-control-with-a-peltier-module-system>
- [33] How to Design a Peltier Module System. *CUI Devices* [online]. 2. duben 2019 [vid. 2023-04-20]. Dostupné z: <https://www.cuidevices.com/blog/how-to-design-a-peltier-module-system>

- [34] *MAX1978-MAX1979.pdf* [online]. [vid. 2023-07-31]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX1978-MAX1979.pdf>
- [35] *lt8722.pdf* [online]. [vid. 2023-05-05]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/lt8722.pdf>
- [36] *MAX1978EVKIT.pdf* [online]. [vid. 2023-08-03]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX1978EVKIT.pdf>
- [37] Opt Lasers TEC-8A-24V-PID-HC-RS232 Temp Controller -100°C to +100°C. *Opt Lasers* [online]. [vid. 2023-07-26]. Dostupné z: <https://optlasers.com/tec-controllers/tec-8a-24v-pid-hc-rs232-programmable-temperature-controller-100-100c>
- [38] *SR-54-datasheet.pdf* [online]. [vid. 2023-04-20]. Dostupné z: <https://lairdthermal.com/thermal-wizard/images/tc/SR-54/SR-54-datasheet.pdf>
- [39] TC-48-20. *TE Technology* [online]. [vid. 2023-04-20]. Dostupné z: <https://te-tech.com/product/tc-48-20/>
- [40] TEC-8A-24V-PID-HC-RS232 Programmable Temperature Controller -100°C to +100°C. *Opt Lasers* [online]. [vid. 2023-04-20]. Dostupné z: <https://optlasers.com/tec-controllers/tec-8a-24v-pid-hc-rs232-programmable-temperature-controller-100-100c>
- [41] TEC-10A-24V Temperature Controller. *Opt Lasers* [online]. [vid. 2023-04-20]. Dostupné z: <https://optlasers.com/temperature-control/tec-10a-24v-temperature-controller>
- [42] TEC-10A-24V Temperature Controller. *Opt Lasers* [online]. [vid. 2023-04-20]. Dostupné z: <https://optlasers.com/temperature-control/tec-10a-24v-temperature-controller>
- [43] TEC-20A-25V-PID Temperature Controller. *Opt Lasers* [online]. [vid. 2023-04-20]. Dostupné z: <https://optlasers.com/tec-controllers/tec-pid-7-25v-max-20a>
- [44] *TEC-1123-HV* [online]. [vid. 2023-04-20]. Dostupné z: <https://www.meerstetter.ch/products/tec-controllers/tec-1123-hv>
- [45] *TEC-1167* [online]. [vid. 2023-04-20]. Dostupné z: <https://www.meerstetter.ch/products/tec-controllers/tec-1167>
- [46] KUMAVAT, Mayur a Sushil THALE. Analysis of CSTR Temperature Control with PID, MPC & Hybrid MPC-PID Controller. *ITM Web of Conferences* [online]. 2022, **44**, 01001. Dostupné z: doi:10.1051/itmconf/20224401001
- [47] *Applied Physics_Thermoelectric effect.pdf* [online]. [vid. 2023-05-03]. Dostupné z: https://ddu.collegedu.ac.in/Datafiles/cms/ecourse%20content/Applied%20Physics_Thermoelectric%20effect.pdf
- [48] SLANINA, Zdenek, Martin UHLIK a Vaclav SLADECEK. Cooling Device with Peltier Element for Medical Applications. *IFAC-PapersOnLine* [online]. 2018, **51**(6), 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018, 54–59. ISSN 2405-8963. Dostupné z: doi:10.1016/j.ifacol.2018.07.129
- [49] JUCA, Sandro, Paulo CARVALHO, Renata PEREIRA, D. PETROV a U. HILLER-INGMANN. Design and Implementation of a High Temperature Control Monitoring Applied to Micro Thermoelectric Generators. *Renewable Energy and Power Quality Journal* [online]. 2013, 712–717. Dostupné z: doi:10.24084/repqj11.425

- [50] MISHRA, S. K., S. SATPATHY a O. JEPSEN. Electronic structure and thermoelectric properties of bismuth telluride and bismuth selenide. *Journal of Physics: Condensed Matter* [online]. 1997, 9(2), 461. ISSN 0953-8984. Dostupné z: doi:10.1088/0953-8984/9/2/014
- [51] *esp32-wroom-32_datasheet_en.pdf* [online]. [vid. 2023-08-07]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- [52] ENESCU, Diana, Cesar DIAZ, Alessandro CIOCIA, Andrea MAZZA a Angela RUSSO. Experimental Assessment of the Temperature Control System for a Thermoelectric Refrigeration Unit. In: *2018 53rd International Universities Power Engineering Conference (UPEC): 2018 53rd International Universities Power Engineering Conference (UPEC)* [online]. 2018, s. 1–6. Dostupné z: doi:10.1109/UPEC.2018.8542014
- [53] *Motor Control Pulse Width Modulator (MCPWM) - ESP32 - — ESP-IDF Programming Guide v5.1 documentation* [online]. [vid. 2023-08-18]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/mcpwm.html>
- [54] *Serial - Arduino Reference* [online]. [vid. 2023-04-28]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>
- [55] *SoftwareSerial Library | Arduino Documentation* [online]. [vid. 2023-04-28]. Dostupné z: <https://docs.arduino.cc/learn/built-in-libraries/software-serial>
- [56] UART communication between Arduino Uno and ESP32. *Hackster.io* [online]. [vid. 2023-08-10]. Dostupné z: <https://www.hackster.io/nidhi17agarwal/uart-communication-between-arduino-uno-and-esp32-1170d5>
- [57] LAB, Microcontrollers. ESP32 UART Communication Explained with Example. *Microcontrollers Lab* [online]. 27. červen 2022 [vid. 2023-08-10]. Dostupné z: <https://microcontrollerslab.com/esp32-uart-communication-pins-example/>
- [58] LAB, Microcontrollers. ESP32 UART Communication Explained with Example. *Microcontrollers Lab* [online]. 27. červen 2022 [vid. 2023-04-28]. Dostupné z: <https://microcontrollerslab.com/esp32-uart-communication-pins-example/>
- [59] *ESP32 Dual Core with Arduino IDE | Random Nerd Tutorials* [online]. 4. říjen 2018 [vid. 2023-04-29]. Dostupné z: <https://randomnerdtutorials.com/esp32-dual-core-arduino-ide/>
- [60] *esp-idf/examples/peripherals/mcpwm/mcpwm_bdc_speed_control at master · espressif/esp-idf. GitHub* [online]. [vid. 2023-08-18]. Dostupné z: https://github.com/espressif/esp-idf/tree/master/examples/peripherals/mcpwm/mcpwm_bdc_speed_control
- [61] *Communicating with Four Boards Through UART with Nano Every | Arduino Documentation* [online]. [vid. 2023-04-28]. Dostupné z: <https://docs.arduino.cc/tutorials/nano-every/run-4-uart>
- [62] *MIC4605-Data-Sheet-DS20005853E.pdf* [online]. [vid. 2023-05-25]. Dostupné z: <https://ww1.microchip.com/downloads/en/DeviceDoc/MIC4605-Data-Sheet-DS20005853E.pdf>
- [63] *IRLU024NPBF INFINEON TECHNOLOGIES - Tranzistor: N-MOSFET | unipolární; 55V; 17A; 38W; IPAK | TME Czech Republic s.r.o. - Elektronické součástky* [online].

[vid. 2023-05-25]. Dostupné z: <https://www.tme.eu/cz/details/irlu024npbf/tranzistory-s-kanalem-n-tht/infineon-technologies/>

[64] *irlr024npbf.pdf* [online]. [vid. 2023-05-25]. Dostupné z: <https://www.tme.eu/Document/adabf59c12041ebc2d466e35a3111f05/irlr024npbf.pdf>

[65] SNYDER, G. Jeffrey, Eric S. TOBERER, Raghav KHANNA a Wolfgang SEIFERT. Improved thermoelectric cooling based on the Thomson effect. *Physical Review B* [online]. 2012, **86**(4), 045202. ISSN 1098-0121, 1550-235X. Dostupné z: doi:10.1103/PhysRevB.86.045202

7 List of Supplement Materials.

S. No.	File Name	Comment
1	Code files	Consists of all firmware codes and testing matlab scripts.
2	PCB	Consists of PCB schematics and PCB board design.
3	Images	Project related images.