DEMONSTRATING THERMODYNAMIC LAWS USING AN 8-BIT MICROCONTROLLER MEASUREMENT OF HEAT AND PRESSURE AND DISPLAYING RESULTS ON A PC.

BY

EITVILAS AZELIS

ST.ID: 10522657

A DISSERTATION

SUBMITTED TO THE SCHOOL OF FUTURE TRANSPORT ENGINEERING

OF COVENTRY UNIVERSITY

BENG IN COMPUTER HARDWARE AND SOFTWARE ENGINEERING

APRIL 2023

SUPERVISED BY MR. SHAUN YEATES

Abstract

This work tries to demonstrate thermodynamic laws by measuring heat and pressure with an 8-bit microcontroller and showing the findings on a PC. Thermodynamics and gas laws were the focus of a series of laboratory experiments that provided a greater comprehension of these concepts. This information was then used to construct a system that included a pressure sensor, two temperature sensors—a thermistor and a thermocouple—and these sensors. The sensor readings were sent to a PC for a better graphical user interface, and the microcontroller was coded using the C18 compiler. Real-time data capture and visualisation are made possible by the user-friendly graphical interface that results, which was developed in C# using Windows Presentation Foundation (WPF). This improves students' knowledge of thermodynamic concepts in a practical situation.

1. Introduction

The rapid advancement of technology has led to the increasing demand for accurate and miniaturized pressure and temperature sensor systems. These systems play a crucial role in various industries, including embedded electronics, medicine, and aerospace (Eitvilas, 2022). With the growing need for highly automated, user-friendly, and precise sensor systems, this project aims to investigate current measurement systems, replicate their functionality, and, if possible, improve upon them. A well-designed sensor system can significantly enhance performance, efficiency, and accuracy in applications such as cardiovascular devices, rockets, submarines, refrigeration, and heat pumps.

The primary objective of this project is to gain a deeper understanding of thermodynamic laws, heat engine functionality, and the essential hardware and software tools required for the application (Eitvilas, 2022). Engineers with limited knowledge in these areas will benefit from researching the relevant topics to support the experiment and application development.

The secondary objective involves conducting experiments in a laboratory (See the Image 1.) setting to verify the functionality and hypotheses, document the results, and confirm the validity of Boyle's, Charles', and Avogadro's laws equations. This step is crucial for the successful development of the intended application (Eitvilas, 2022).

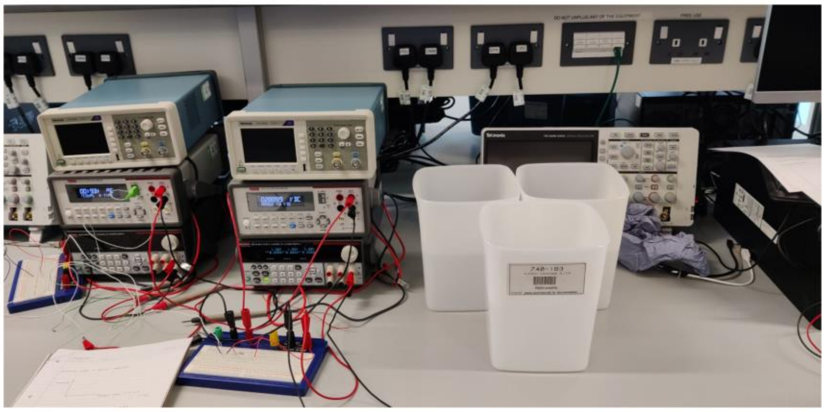


Image no.1

The tertiary objective consists of designing and implementing the system using Proteus, connecting the microcontroller with the power supply Transmitter/Receiver module, and sensors. Programming the calculations and establishing communication between the system and a PC will follow. Furthermore, creating a Graphical User Interface (GUI) for user monitoring and control is essential (Eitvilas, 2022).

Lastly, the project aims to identify potential improvements to the system, such as incorporating additional communication methods or simplifying the GUI. Upon completion, the developed sensor system can be immediately applied in various sectors, including home automation, greenhouses, vehicles, and atmospheric monitoring (Eitvilas, 2022).

2. Literature Review

Understanding of the Heat and Work Equations, as well as the First and Second Laws of Thermodynamics, is necessary for the project. To develop a system that accurately measures heat and pressure in accordance with the project's goals, certain laws and equations are crucial. (Eitvilas, 2022).

The projects also heavily relies on Boyle's, Charles', and Avogadro's Laws. Avogadro's Law deals with gas molecules in equal volumes, while Boyle's Law and Charles' Law relate pressure, volume, and temperature, respectively. For the project's design to be implemented and sensor data to be interpreted, it is vital to comprehend these relationships. (Eitvilas, 2022).

3. Methodology

3.1. Research and Learning Approaches

Thermodynamic laws, notably the first law, and gas laws, such as Boyle's, Charles', and Avogadro's laws, were studied during the course of the project. The microcontroller, temperature and pressure sensors, programming languages, and their frameworks were chosen in considering research. A fundamental grasp of programming the PIC with the C language for the microcontroller, C# for the GUI, and the system for communication was provided via degree coursework as well as video courses from YouTube (Eitvilas, 2022). Project-related journals were examined and used in the report. Through courses from W3Schools, the C# programming language was mastered, and work on the GUI platform was completed. To gain a deeper understanding of the gas and thermodynamic laws materials labaratory experiments were conducted as well the videos from YouTube were obtained.

3.2. Sensor Selection and Integration

Pressure is converted to voltage via the piezoresistive transducer in the pressure sensor. Specifications and calculations for the pressure sensor (NXP Semiconductors, 2018), thermistor (Amphenol Advanced Sensors, 2020), thermocouple aplifier (Analog Devices, 2000) and microcontroller (Microchip Technology Inc., 2007) are provided in its datasheet, which is essential for the project's correctness. (Eitvilas, 2022).
At various temperatures, the thermocouple, a crucial temperature-measuring sensor, creates an electrical connection. Although it only outputs a tiny signal, the design can be improved by including an amplifier and an additional energy source to assure correct data. (Eitvilas, 2022).

Another component that measures temperature is the thermistor, which changes resistance in response to temperature variations. It contributes significantly to the project thanks to its excellent sensitivity, quick response, and low heat capacity. (Eitvilas, 2022).

4. Experiment No.1

4.1. Introduction

For a wide range of applications across multiple fields, reliable and accurate temperature measurement is essential. This project involved building and using a temperature sensor circuit based on thermistors. The sensor was calibrated to assist in creating temperature monitoring and management systems, and several temperature-sensing devices were investigated.

In order to improve the accuracy and dependability of temperature measurements, the experiment attempted to understand better the capabilities and constraints of various temperature sensing devices. The findings might be used in areas like engineering, business operations, and scientific research where precise and reliable temperature readings are essential to the project's success.

The outcomes of this experiment might be helpful for academics and other professionals whose job involves temperature measurements. The accuracy and reliability of temperature readings can be increased, thereby enhancing the performance and effectiveness of temperature measurement systems by better understanding the operation of various temperature sensing devices and calibration techniques. Additionally, this experiment might provide as a starting point for later investigations on the efficiency of various temperature-sensing tools, calibration procedures, and environments.

4.2. Literature Review:

A Simple Thermistor Design for Industrial Temperature Measurement:

A simple and affordable thermistor design for temperature measurement in various industrial applications was put out by Kufre Esenowo Jack et al. (2016). They used an NTC thermistor, a voltage divider circuit, and a microprocessor to process signals and determine temperature. The resulting device was suited for industrial temperature measurement due to its broad temperature range, high sensitivity, and accuracy. This study adds to the body of knowledge on thermistor testing and application by highlighting the adaptability and efficiency of thermistors in various applications.

Design of High Precision Temperature Measurement System Based on LabVIEW:

Zhu et al. (2015) reported on a LabVIEW-based high-precision temperature measurement system. The system used a microprocessor for data processing and acquisition and a thermistor as a temperature sensor. Real-time temperature monitoring and control were made possible by the LabVIEW platform's user-friendly interface and robust analysis features. The proposed system demonstrated the potential of integrating thermistors with LabVIEW for advanced temperature sensing applications by achieving high precision and stability in temperature measurements.

Control and Synchronization in Nonlinear Circuits by Using a Thermistor:

The use of thermistors in regulating and synchronising nonlinear circuits was investigated by Zhang et al. (2020). The authors examined the thermistor-based circuit's dynamic behaviour and presented a cutting-edge control strategy that could synchronise two connected circuits. The work highlighted the potential of thermistors in diverse electronic systems by showing how they could be used in complex control applications beyond temperature monitoring.

4.3. Methods

For applications including both intermittent and continuous temperature monitoring, a biomedical chip thermistor assembly was used in the experiment. The sensor's resistance reduces as the temperature rises because it has a negative temperature coefficient (NTC).

The experiment used an interchangeable thermistor with a temperature range of 0°C to 50°C. Per the thermistor's datasheet, maximum overall stability is maintained at exposure and storage temperatures below 70 °C.

A breadboard was used to build the circuit after the thermistor was attached to the positive and negative source connections (See the Image 2.)

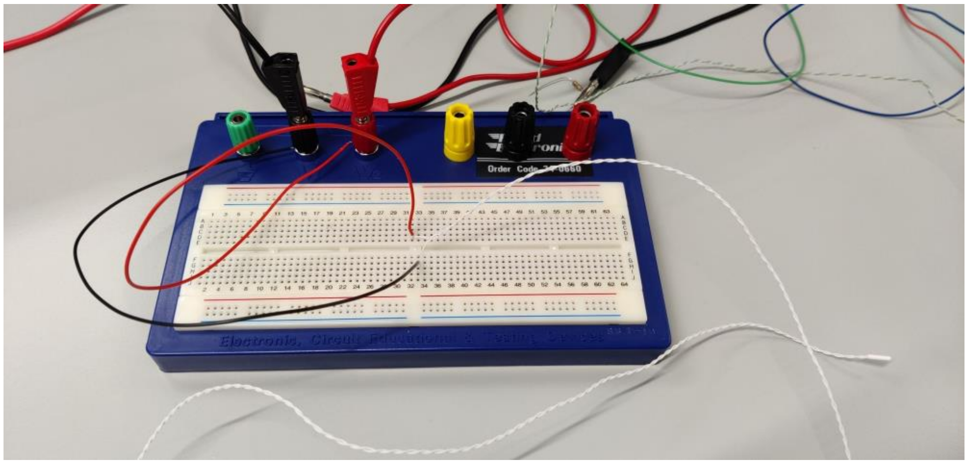


Image No.2

Three buckets were filled with water at varied temperatures to simulate various environments for the experiment. Two temperature sensors were connected, the thermistor connected to an unconfigured breadboard to display voltage and the thermocouple to a digital multimeter to display the temperature. Both sensors were placed into the buckets to measure Celsius's voltage and actual temperature simultaneously. To create a variety of temperature data points, three or four different temperature readings were recorded for each bucket. *See the Image no.3:*

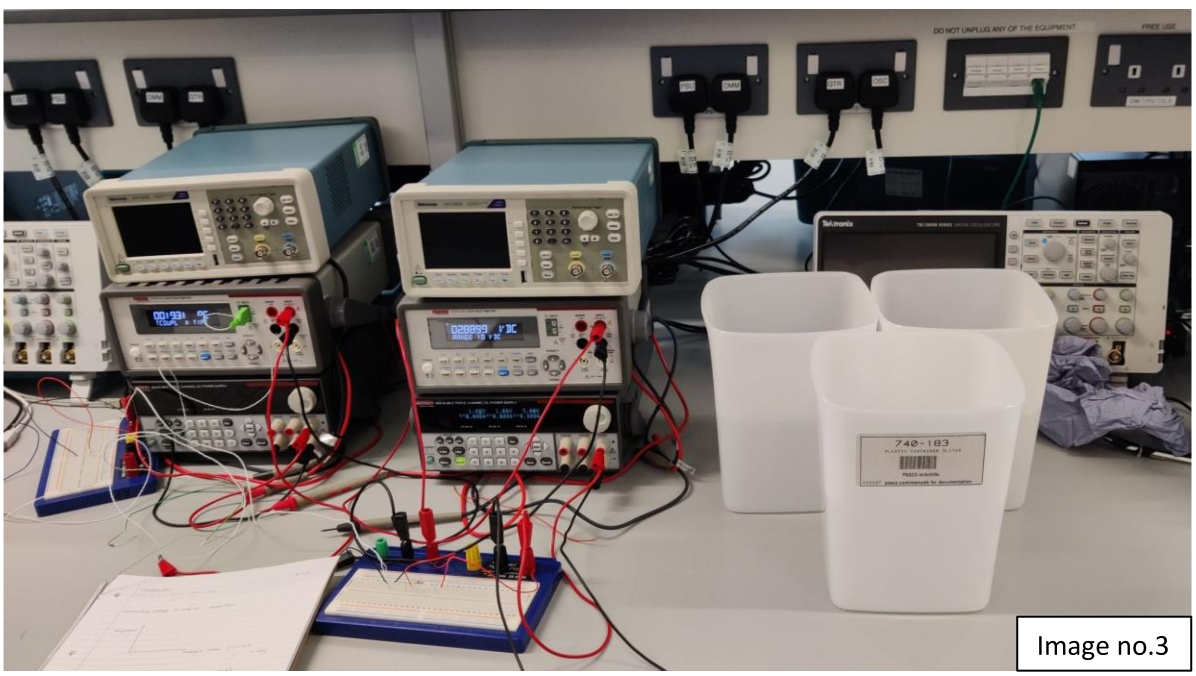


Image no.3

After the temperature readings were gathered, a 10k ohm resistor was added to the circuit (*See the Image no.4*). The exact actions were taken to collect the readings again to compare the circuit's performance before and after the resistor was added.

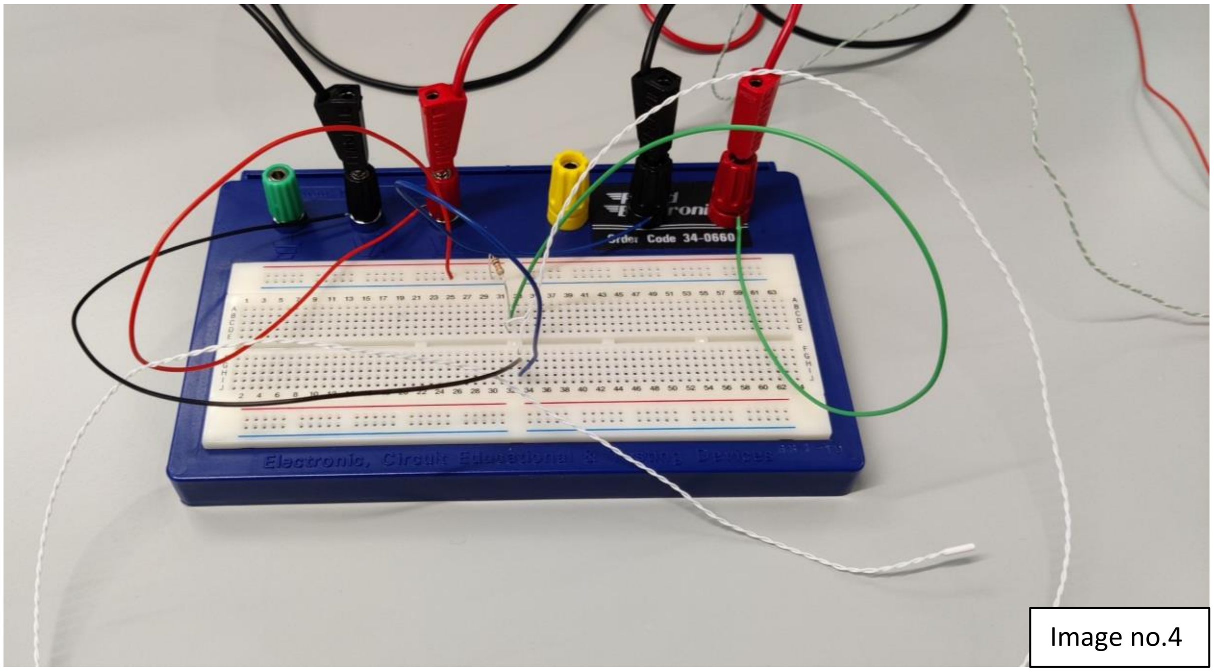


Image no.4

4.4. Results

The digital multimeter recorded the results and visualised the data as resistance versus temperature (*See Tables below*). The sensor results **without** a resistor (See Table *no.1*), and the results **with** a 10k ohm resistor in the circuit (See Table no.2).

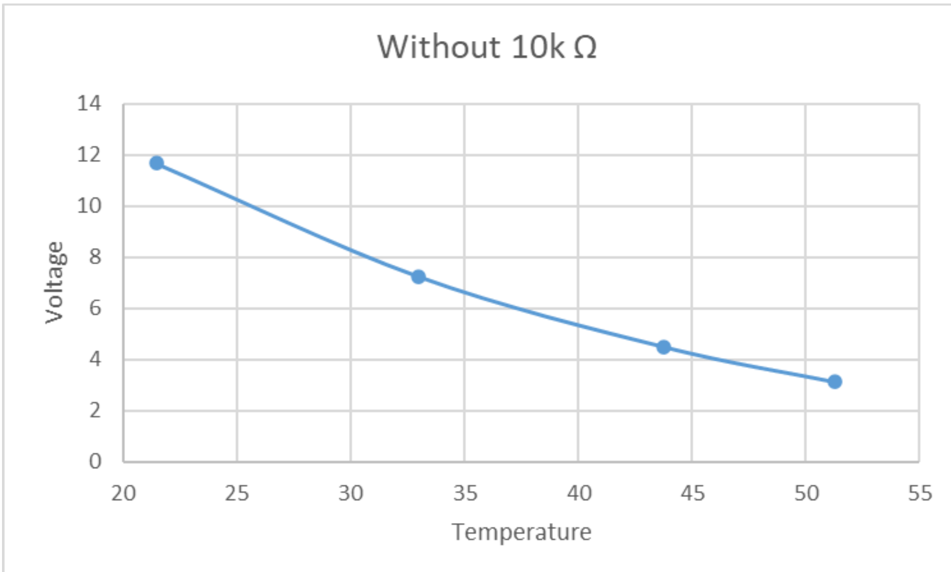
Temperature	Voltage
21.47	11.668
33	7.25
43.77	4.508
51.3	3.13

Table no.1

Temperature	Voltage
21.74	2.6829
32.5	2.14
36.3	1.8162
48.37	1.408

Table no.2

The data was used to make a voltage versus temperature graph to show how voltage and temperature relate to one another. (*See Chart no.1 and See Chart no.2*). The voltage vs temperature plot revealed a non-linear connection between voltage and temperature, and the calibration findings demonstrated that the thermistor was sensitive to variations in temperature.



Char no.1

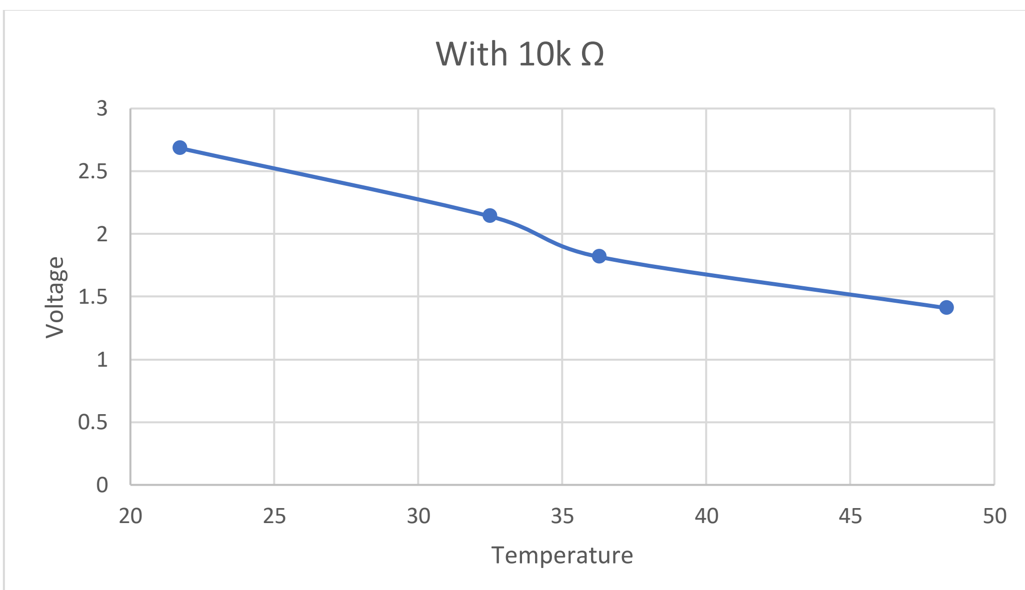


Chart no.2

4.5. Discussion

The experiment investigated the application of thermistors as temperature sensors and the significance of calibration and sensitivity changes based on resistors. The findings demonstrated that the thermistor assembly offered quick responses and great sensitivity when measuring temperature. The 10k ohm resistor, however, improved the circuit's stability and brought attention to the trade-off between sensitivity and accuracy in temperature readings.

4.6. Conclusion

The experiment showed how to use a thermistor as a thermometer and highlighted the importance of calibrating and adjusting sensitivity using resistors. According to the analysis, the thermistor sensor provided excellent stability and quick response for temperature monitoring. The results also showed that sensitivity and stability must be maintained while building sensor equipment for various applications.

Thermistors are suitable for circumstances where the temperature ranges from 0°C to 70°C because they offer a quick reaction and good stability for temperature monitoring. A thermocouple is advised for more precise measurements in the -50°C to 250°C temperature range.

Overall, the experiment provided helpful information about using thermistors to measure temperature and the variables to consider when designing sensor equipment for diverse environments.

5. Experiment No.2

5.1. Introduction

This experiment’s goals are to illustrate the fundamentals of thermodynamics, with a particular focus on the laws of energy conservation and how heat and energy behave in physical systems. The experiment aims to provide students with practical experience building and analysing thermodynamic systems, improving their understanding of the fundamental concepts pointing to the behaviour of heat and energy.

5.2. Literature Review

The "Introduction to the Thermodynamics of Materials" by Gaskell and Laughlin (2017), which focuses on energy conservation and heat behaviour in physical systems, offers a thorough explanation of the core ideas of thermodynamics. The first and second laws of thermodynamics are covered by the writers, who highlight the importance to understand them in order to comprehend energy conservation and heat transmission.

The first law states that energy cannot be created or destroyed, only changed in form. The second law addresses heat transfer direction and the concept of entropy, which measures disorder in a system. To explain heat and energy behaviour in materials science. The Gaskell and Laughlin (2017) provides a solid foundation for comprehending the fundamentals of thermodynamics, energy conservation, and heat behaviour in physical systems.

5.3. Methods

The development of the thermodynamic system included physics class exercises that involved the movement of heat and energy through various physical processes. The precise design of the system depended on the knowledge and expertise of the student, although it might have involved looking at device datasheets and lab sheets, researching the behaviour of a Carnot engine, examining the expansion and compression of a gas, or looking at other thermodynamic processes.

A pressure sensor (MPXA4250AP), a 5V fixed power supply, connecting leads, a digital multimeter (DMM) for measuring voltage and resistance, a breadboard, a heat engine (*See Image no.5*), and various weights were used in the laboratory's thermodynamic system and pressure measurement device.

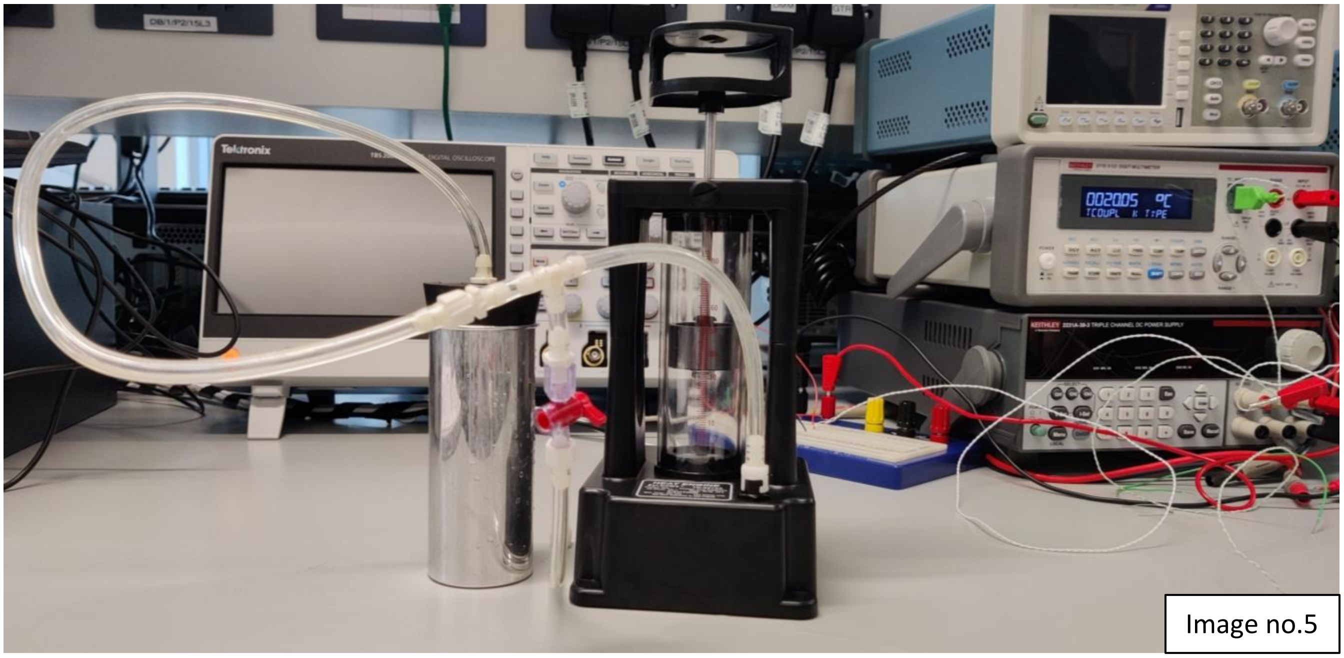


Image no.5

Compressing a closed volume with the heat engine installed vertically, the given sensor was examined in the experiment (*Image no.6*). The pressure gauge needed a 5V DC power supply to function, therefore, it was linked to a DC power supply with black (negative) and red (positive) wires, and the output (yellow) was attached to the DMM.

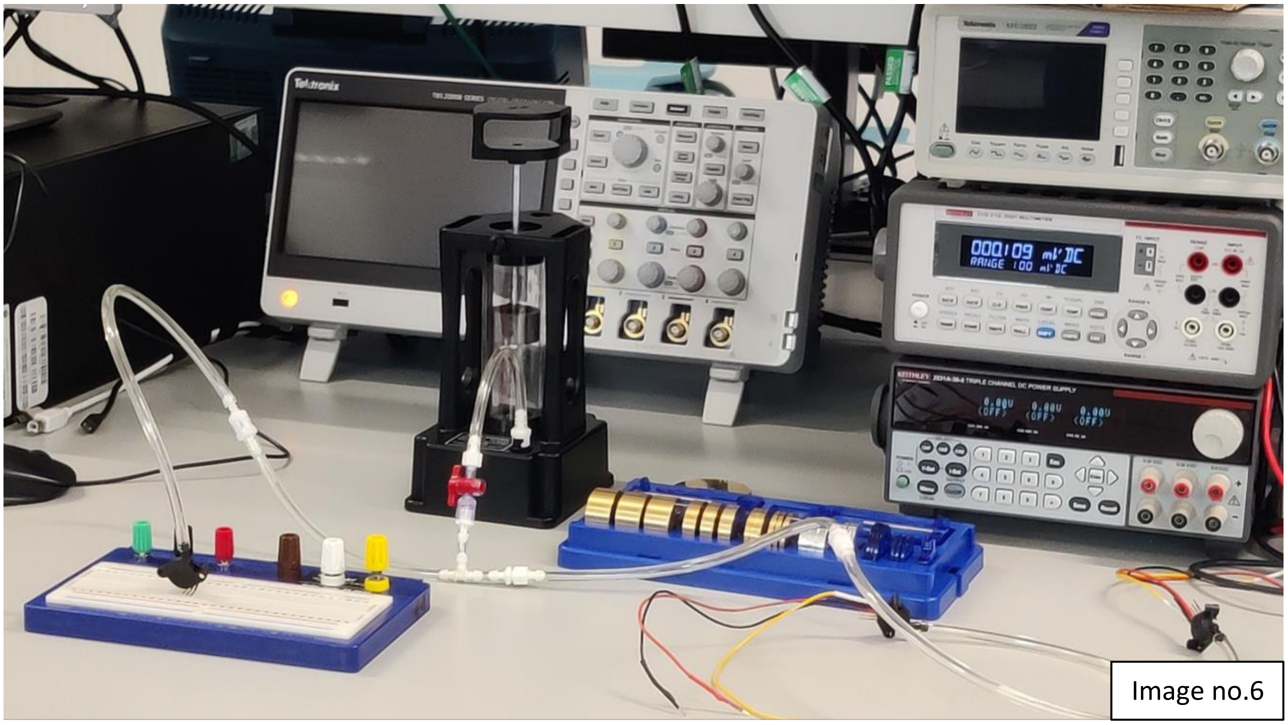
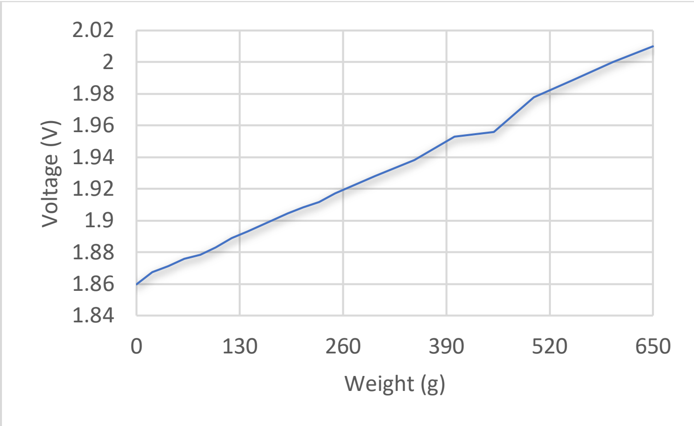


Image no.6

The sensor was connected to the tube, the piston was raised to the proper height and locked into position by closing the valve. Different pressure values could be produced by increasing the weight on the platform (heat engine), and the DMM was used to detect the output values (Voltage). It was noted that the piston's diameter was 32.5mm and that the DMM was configured to show a range of 100mVDC.

5.4. Results

We recorded and plotted the weight, displacement, and voltage. The chart below shows **displacement** versus **weight** (See Chart no.3) and **voltage** versus **weight** (See Chart no.4).



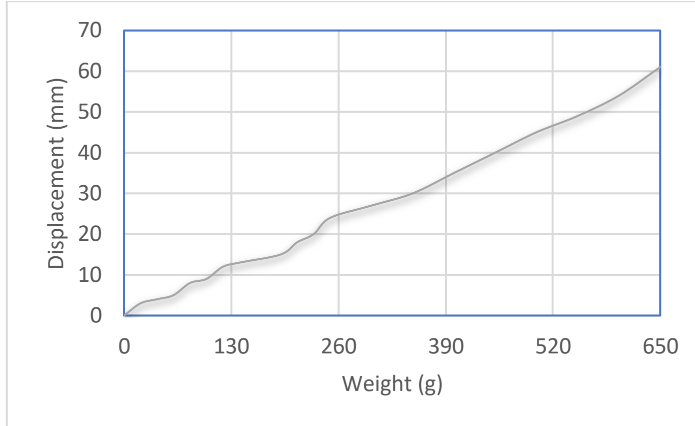


Chart no.3

Chart no.4

The table (See Table no. 3) below shows the numerical data that was presented in the graphical chart above.

Mass (g)	Displacement (mm)	Voltage
0	0	1.8597
20	3	1.8675
40	4	1.8714
60	5	1.876
80	8	1.8784
100	9	1.883
120	12	1.8889
140	13	1.893
190	15	1.9045
210	18	1.9081
230	20	1.9118
250	24	1.9173
300	27	1.9281
350	30	1.9383
400	35	1.9528
450	40	1.95613

500	45	1.9779
550	49	1.9886
600	54	2
650	61	2.01

Table no. 3

With the collected data, the formula was used to calculate:

$$\textit{Pressure} = \textit{Force} / \textit{Area}$$

$$\textit{Force} = m \times g$$

m is the **mass** (converted to kg) on top of the piston, and the **g** is the **gravitational constant** (gravity). The gravitational constant in England is 9.80. It’s important to highlight it that the gravitational constant is not the same everywhere on Earth.

$$\textit{Area} = (\pi / 4) \times (d^2)$$

Where d = 32.5mm = 0.0325m (diameter of the piston). The displacement has to be converted to the volume.

5.5. Discussion:

The experiment described the pressure sensor device and showed how to build and apply a pressure measurement circuit in practice. The results revealed that accurate sensor calibration and pressure measurement were made possible by exploiting the heat engine and various weights. The experiment must be run as slowly as possible to allow carefully applying the weight to the sensor evenly and gradually to avoid damaging the system or risking the accuracy of the measurements. Adding weight to the platform gradually may have been necessary to provide the DMM adequate time to capture the voltage changes accurately.

5.6. Conclusion:

The experiment successfully demonstrated how a pressure measuring circuit works and gave participants valuable practical experience with pressure sensors. The results showed that pressure can be precisely examined with a straightforward circuit and inexpensive instruments.

6. Experiment No.3

6.1. Introduction

The goal of this laboratory experiment is to gain a better grasp of thermodynamics, in particular Charles' Law, which states that, while pressure is held constant, the volume of a gas is precisely proportionate to its temperature. The experiment solves many aspects of thermodynamics and the real-world uses of Charles' Law by examining the relationship between the temperature and volume of a gas using a heat engine device and a thermocouple for temperature monitoring.

6.2. Literature Review

The law of volumes, commonly referred to as Charles' Law, is the basis of thermodynamics and states that, under conditions of constant pressure, the volume of an ideal gas is precisely proportionate to its temperature. Numerous scientific and engineering fields, including meteorology, aerospace engineering, chemical engineering, and the study of thermodynamics and heat transport, all depend on this concept.

French physicist Jacques Charles initially proposed Charles's Law in the 1780s. The equation for the law is:

$$\frac{V_1}{T_1} = \frac{V_2}{T_2}$$

Where V1 and V2 are the gas's initial and final volumes, respectively, and T1 and T2 are their respective initial and final temperatures, measured in Kelvin. (K). Charles's Law has been applied in various domains, according to several studies:

- In meteorology, Charles's Law is used to understand atmospheric processes and predict weather patterns (Mechem, 2013).
- In aerospace engineering, the law is essential for designing propulsion systems and studying the behavior of gases at high altitudes (Hill & Peterson, 1992).

- In chemical engineering, Charles's Law plays a crucial role in optimizing processes such as distillation, gas separation, and chemical reactions (Elliott et al., 2012).

6.3. Methods

The red lock screw was released but left in place when the heat engine setup, the tubes from the pressure vessel was divided and one end was connected to the pressure sensor and another tube end into sealed volume. The pressure vessel was adjusted in the middle (40mm) and sealed with a screws help. The pressure sensor was supplied with the 5V power supply, and connected to the digital multimeter for the output (See Image no.7)

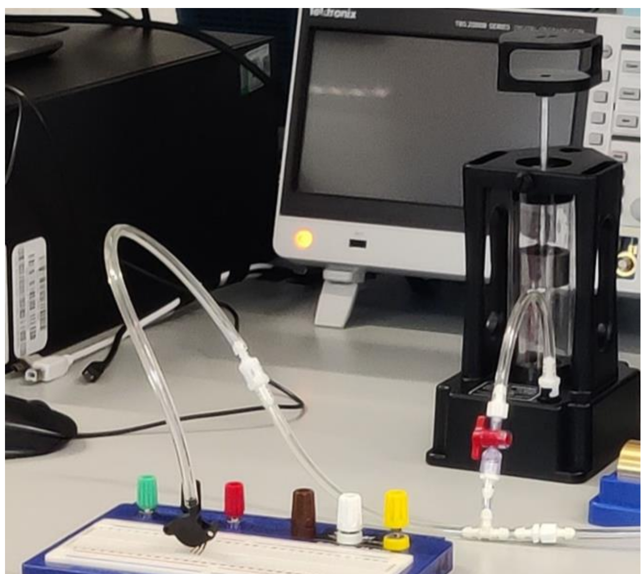


Image no.7

The pressure vessel was positioned on its side to minimize the effects of gravity (See Image no.8)

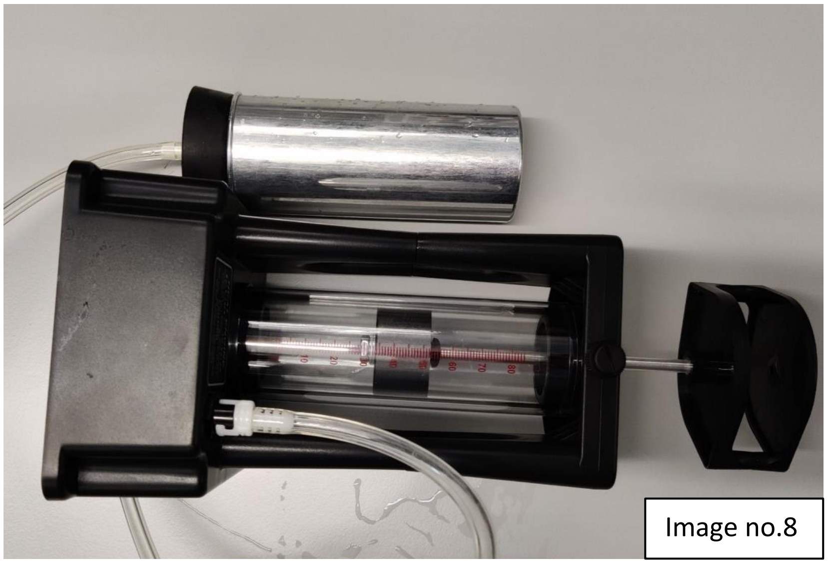


Image no.8

The sealed volume was placed within each of the four different water tanks (The same water tanks from the Experiment No.1), each of which had been preheated to a different temperature. The temperature, voltage from the sensor, and displacement of the pressure vessel were recorded using a calibrated thermocouple. The volume was calculated using the graduations on the side of the pressure vessel, taking into account the piston's diameter of 32.5mm.

6.4. Results

Temperature, displacement and voltage were all the experiment's data that were documented and displayed in the table below:

Temperature (°C)	Displacement (mm)	Voltage
2.46	-15	3.818
23.57	-5	2.631
50.78	+8	1.308

65.92	+21	0.904

Table no.4

The charts below that visually represents the data, which was previously presented as a table. Temperature versus Voltage (Chart no.5) and Temperature over Displacement (Chart no.6):

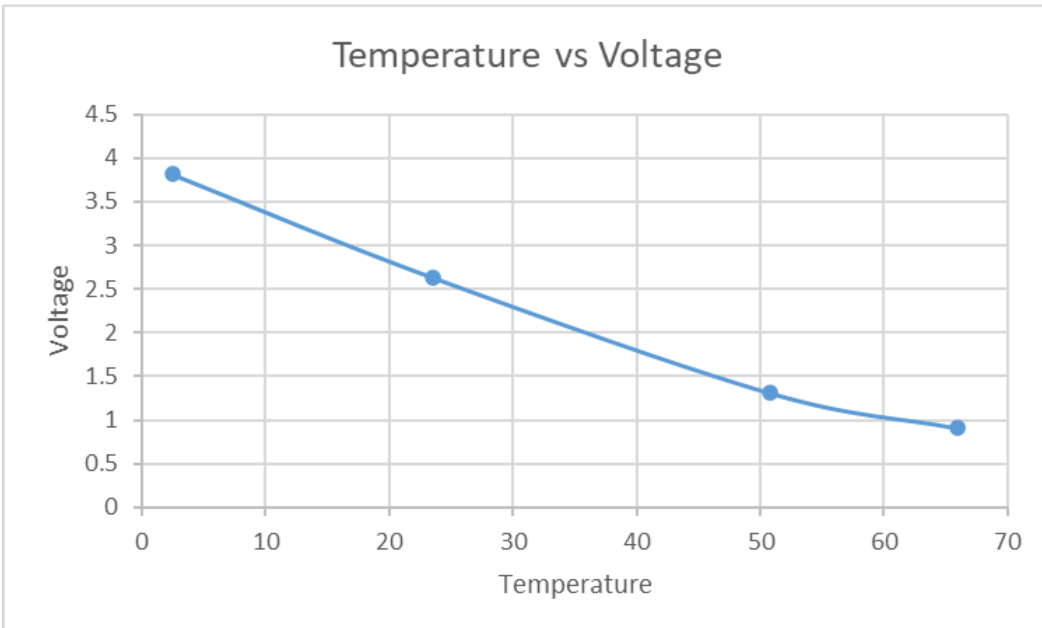


Chart no. 5

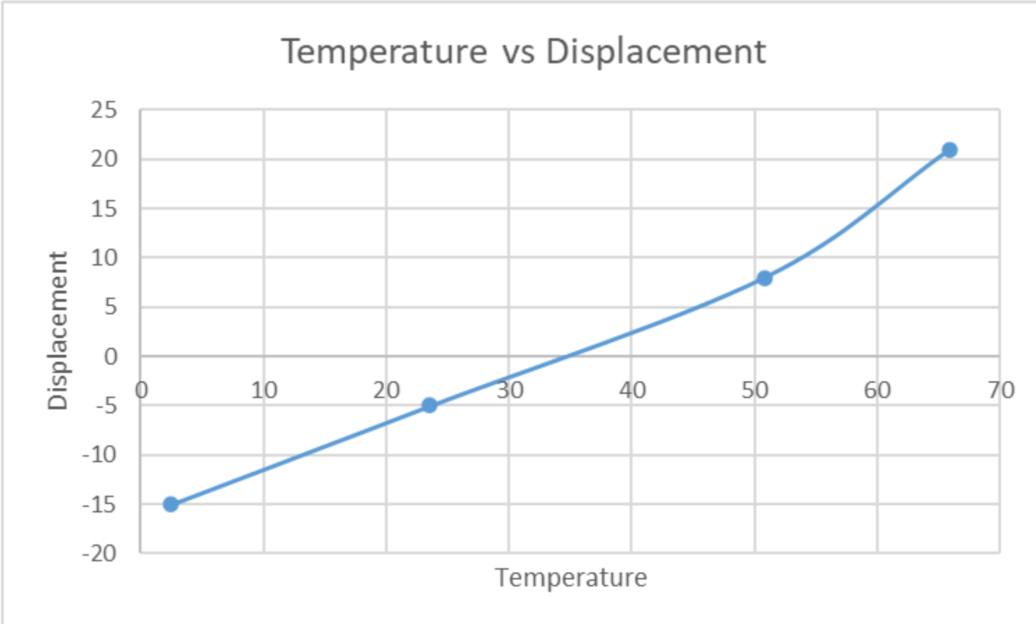


Chart no. 6

6.5. Discussion

The experiment successfully illustrated Charles's Law by showing how, under conditions of constant pressure, the temperature and volume of a gas (air) are directly related. The heat engine device enabled precise volume measurements, while the thermocouple delivered accurate temperature readings. However, certain mistakes or problems might have occurred as a result of the delicate nature of the device or the use of water tanks with different temperatures.

6.6. Conclussion

By examining the relationship between temperature and volume of a gas using a heat engine apparatus, the laboratory experiment enhanced our knowledge of thermodynamics, particularly Charles' Law. The experiment highlighted the significance of precise temperature and volume measurements in addition to the numerous practical uses of Charles' Law in science and engineering. These include meteorology - predicting the weather, simulating climate change, and recognizing atmospheric processes, aerospace engineering - designing and analysing propulsion systems, researching the behaviour of gases at high altitudes, chemical engineering for optimising processes like distillation, gas separation, and chemical reactions; and in the study of thermodynamics and heat transfer for creating energy-efficient heating and cooling systems.

7. System Design

7.1. Hardware Design

7.1.1 Proteus

The project's hardware consists of of a microcontroller (PIC18F4520) running at 8MHz, three sensors, an LCD display, and additional parts. A thermistor, an MPX4250 pressure sensor, and a thermocouple (K-type) make up a set of sensors. The pressure sensor is linked to RA4, the thermocouple to RA1, and the thermistor to RA0 of the microcontroller (See Image no.9).

For precise analogue-to-digital conversion in the microcontroller's ADC, a reference voltage is supplied via a voltage divider circuit connected to RA3. The top resistor of this voltage divider is linked to the 5V power supply, while the bottom resistor is connected to ground (See Image no.9). This voltage divider is made up of two resistors: a 1.5k resistor at the top and a 1k resistor at the bottom. Calculating the reference voltage is as follows: (1k / (1.5k + 1k) x 5V = 2V.

A 8-bit interface is used to link an LCD display to the microcontroller, provides faster data transfer by sends entire byte, simplified code and less overhead for the microcontroller. PortD is linked to the data lines. (RD0-RD7). LCD control lines, that includes RS (register select), RW (read/write), and E (enable) are connected to RC0, RC1 and RCD, respectively. A potentiometer is used to control the V_{SS}, V_{DD}, and V_{EE} connections on the LCD display in order to change the contrast of the screen (See Image no.9).

To enable precise temperature measurements, the thermocouple is coupled to an AD595 component, which is required to amplify the weak voltage signal produced by the thermocouple and provide cold junction compensation. The +IN and -IN pins of the thermocouple are used to connect the AD595 to it. The microcontroller's ADC input is joined to the AD595's output (See Image no.9).

Serial connection between a microcontroller and a PC can only be made possible with the help of the virtual terminal and COMPIM components (See Image no.9). The RC6 (TX) and RC7 (RX) ports on the microcontroller serve as the connection points for the virtual terminal, and the COMPIM component serves as a bridge between the virtual terminal and a real serial port on the PC. With this configuration, data can be sent and received for monitoring and controlling between the microcontroller and PC.

Additional parts, such as power supply circuitry and a printed circuit board (PCB) for mounting and connecting the parts as well as storage space for the complete system would be needed for a practical implementation of the system. To further reduce noise and interference that can impair the system's accuracy and dependability, adequate shielding and grounding measures should be used.

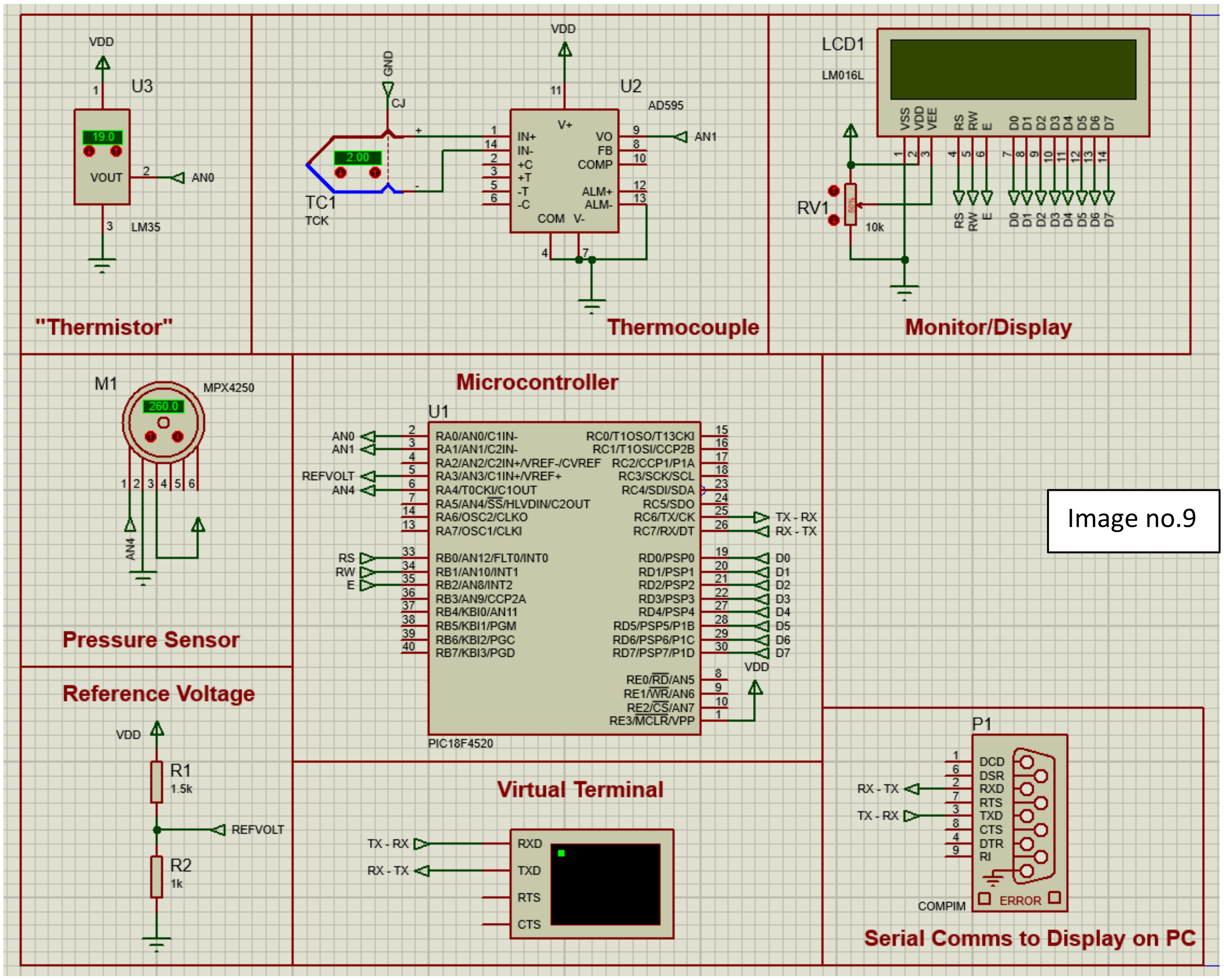


Image no.9

7.2. Software Design

7.2.1. Microcontroller Firmware

This project's microcontroller firmware is built to take sensor data from the pressure, thermocouple, and thermistor, process the data, and then send it over USART to a pc for additional processing and display. The firmware is divided into a number of functions that read sensor data, manage interrupts, and configure the microcontroller and its peripherals. The essential parts of the firmware are described in the sections that follow.

Configuration: The firmware configures the USART, ADC, analog input pins, and Timer0 with the following functions:

- **configureUSART():** This function sets up the USART with the appropriate settings for asynchronous communication, 8-bit data length, continuous reception, and high baud rate generation. The Baud rate is configured to be 9600 bps. (See Image no.10).

To calculate the Baud Rate Generator (BRG):

$$\frac{Fosc}{(DesiredBaudRate\ x\ 16)\ -\ 1}$$

Image no.10

```
void configureUSART(void)
{
    OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNCH_MODE & USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH, 51);
}
```

- **configureADC():** This function sets up the ADC module with external VREF+ and VSS for external VREF-, a FOSC/8 conversion clock, right-justified result format, and 20 TAD acquisition time. (See Image no.11).

Image no.11

```
void configureADC(void)
{
    OpenADC(ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_20_TAD, ADC_CH0 & ADC_INT_OFF & ADC_VREFPLUS_EXT & ADC_VREFMINUS_VSS, 0b1010);
}
```

- **configureAnalogInputPins():** This function configures the analogue input pins RA0, RA1, and RA4 as inputs by setting the TRIS bits for each of those pins. (See the Image no.12).

```
void configureAnalogInputPins(void)
{
    TRISAbits.TRISA0 = 1;
    TRISAbits.TRISA1 = 1;
    TRISAbits.TRISA4 = 1;
}
```

Image no.12

- **configureTimer0():** This function sets up Timer0 in 16-bit mode with a 1:1 prescaler, internal clock source, and interrupt support. (See the Image no.13).

```
void configureTimer0(void)
{
    CloseTimer0 ();
    OpenTimer0 (TIMER_INT_ON & T0_16BIT & T0_SOURCE_INT & T0_PS_1_1);
    WriteTimer0(0);
    INTCONbits.GIE = 1;          // Enable global interrupts
    INTCONbits.TMR0IE = 1;       // Enable Timer 0 Interrupt
    INTCONbits.TMR0IF = 0;
}
```

Image no.13

Interrupt Service Routines (ISR):

An ISR with high priority is built into the firmware to handle Timer0 interrupts. When a Timer0 interrupt occurs, the TMR0_ISR() function is invoked and increases a counter. (timerCount). The periodOneSecond flag is set and the counter is reset when timerCount reaches 30.

Sensor Reading and Data Processing:

Reading and processing of sensor data is done using the readSensor() function, which accepts an ADC channel as an input. To transform the unprocessed ADC values into useful sensor values, it makes the necessary computations based on the channel.

Data Transmission:

The sensor values are passed as parameters to the sendData() function, which transforms them into a string that resembles JSON and looks like this:

{"AN0": sensor1, "AN1": sensor2, "AN4": sensor3}\r\n

To determine the ideal baud rate for sending this data:

- JSON data: 29 characters (including the 3 sensor values, assuming each value has the same length).
- Sensor values: Let's assume 5 characters per sensor value (e.g., "123.4" or "-12.34"), making it a total of 15 characters.
- Total characters: 29 + 15 = 44 characters.

Each character is sent as a byte (8 bits). Assuming you are using the standard asynchronous mode with 1 start bit, 8 data bits, and 1 stop bit, you will need 10 bits per character.

Total bits: 44 characters * 10 bits/character = 440 bits.

To calculate the baud rate:
Baud rate = Total bits / Time = 440 bits / 1 second = 440 bps (bits per second)

However, it's generally a good idea to utilise a greater baud rate to account for unexpected delays and assure stable connection. Microcontrollers often operate at baud speeds of 9600 bps, 19200 bps, or 38400 bps.

Main Loop:

The microcontroller continuously determines whether the periodOneSecond flag is set in the main() function. When set, the sensor values are read and shared over USART while the flag is reset. The readSensor() function, which accepts an ADC channel as an input, reads the sensor data by doing the necessary calculations based on the channel to translate the raw ADC results into useful sensor values.

7.2.2. Graphical User Interface Application

The Graphical User Interface (GUI) application is used to show and track sensor data that was sent to a microcontroller via a serial communication channel. The information consists of pressure readings from an MPX4250 sensor, thermistor and thermocouple readings, and temperature values. The Settings, Front and Graphs pages are the three primary parts of the application:

1. **SettingsPage:** makes it possible for users to set up and preserve the parameters required for creating a serial communication connection with a microcontroller. The COM port, baud rate, parity, and data bits are some of these parameters.
    1.1. SerialCommApply(object sender, RoutedEventArgs e): Validates and saves the user-configured serial communication settings.
    1.2. AreSerialConfigsValid(): Checks if the user-configured serial communication settings are valid.
2. **FrontPage:** Based on the saved settings from the SettingsPage, the FrontPage creates a serial communication connection. It parses the data it receives from the microcontroller and displays the sensor values.
    2.1. ConnectSerialCom(object sender, RoutedEventArgs e): Connects to the microcontroller using the saved serial communication settings.
    2.2. SetupSerialPort(): Sets up the serial port connection based on the saved serial communication settings.
    2.3. HighlightReceivedData(): Highlights the received data in the UI.
    2.4. UpdateSensorData(JObject sensorData): Updates sensor labels and values based on the parsed sensor data.
    2.5. ParseJsonData(string jsonData): Parses JSON-formatted sensor data.
    2.6. SerialPort_DataReceived(object sender, SerialDataReceivedEventArgs e): Handles the serial data received event and processes the received data.
    2.7. DisconnectSerialCom(object sender, RoutedEventArgs e): Disconnects the serial communication connection.
3. **GraphsPage:** The GraphsPage shows graphs of the sensor data that the microcontroller has received in real time. LiveCharts is used to display the data.
    3.1. SetupCharts(): Sets up the chart configurations and initializes the LiveCharts.
    3.2. TimerOnTick(object sender, EventArgs eventArgs): Updates the charts with new data points and removes old data points to keep the chart range limited.
    3.3. SerialComConfigs: A static class that stores the user-configured serial communication settings.

3.4. SerialComReceivedData: A static class that stores the latest received sensor data.

3.5. MeasureModel: A class used for storing data points in the LiveCharts.

7.3. Integration and Testing

7.3.1. Proteus

Microcontollers firmware runs the controller, collects the data from the ADC and transmits the data in the JSON format (See Image no.14) through the Serial Communicaiton port.

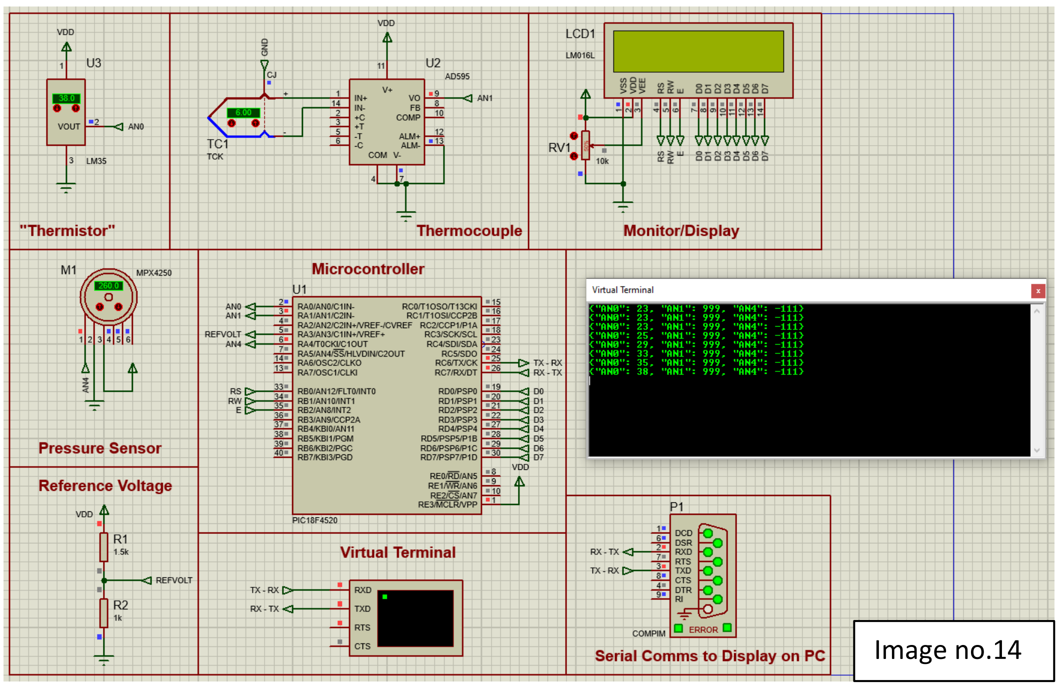


Image no.14

7.3.2. GUI Application

SettingsPage: Users can set and store serial connection options on this page, including the COM port, baud rate, parity, and data bits. To make sure the serial connection is configured correctly, it verifies the input and gives the user feedback (See Image no.15).

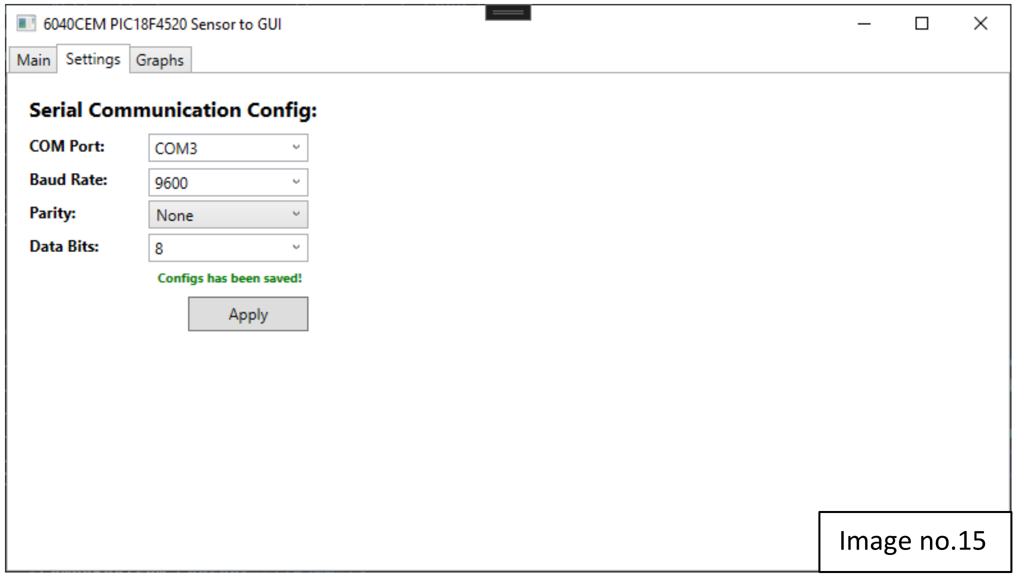


Image no.15

FrontPage: Establishing and maintaining the serial communication connection with the microcontroller is the responsibility of this page. It receives, analyses, and presents the sensor data in a user-friendly way. Options to connect or remove the serial communication are also available on the page (See Image no.16).

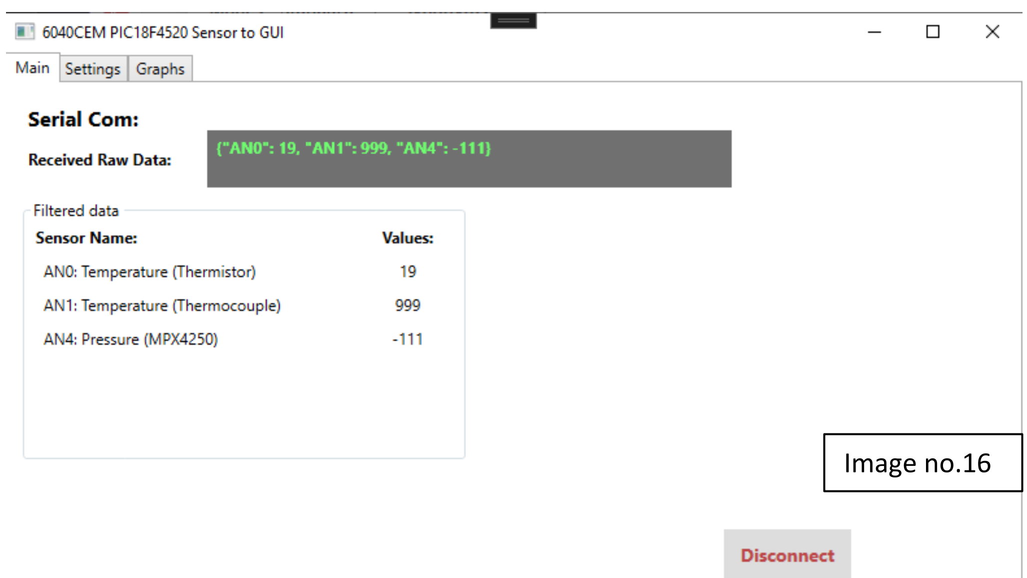


Image no.16

GraphsPage: Line charts are used on this page to display the sensor data in real-time. It provides users with an easy approach to monitor and analyse the sensor data by constantly updating the charts with the most recent data and maintaining a small range to display recent data points (See Image no.17).

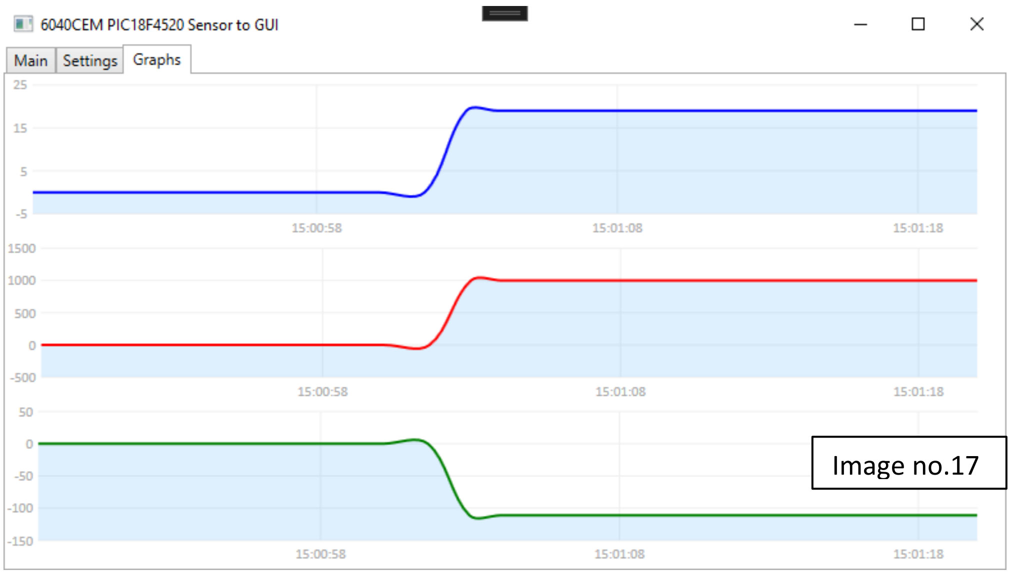


Image no.17

7.3.3. Video demonstration

The link has been provided to show how to use and functionality of application.

Link - https://drive.google.com/drive/folders/1ZNpqKVlOeSZrvwHQNL-m5eYmDCjP944o?usp=sharing

8. Summary and Conclusion

This study has successfully illustrated how to apply thermodynamic rules by measuring heat and pressure with an 8-bit microcontroller and showing the findings on a user-friendly PC interface. The system incorporating temperature and pressure sensors was developed after a series of laboratory experiments to reinforce the understanding of thermodynamics and gas laws. The system improved understanding of thermodynamic concepts in a real-world context by demonstrating the efficient integration of microcontroller programming, sensor data gathering, and real-time data visualisation on a PC. Unfortunately, due to the unknown error, only the first sensor was updated. To further enhance the usability and functionality of such apps, further research may include the incorporation of more sensor types, the improvement of data processing methods, and the creation of more complex visualisation tools.

9. Further Work

Future work can examine different sensor types for more accurate and exact measurements as well as incorporate extra environmental elements like humidity or air flow. Additionally, the analysis and interpretation of sensor data may be enhanced by the development of more sophisticated data processing methods, such as machine learning algorithms. As well, fix the code for the ADC_CH1 and ADC_CH4 to make them functional. The graphical user interface might also be improved to include more complex visualisation capabilities, enabling a more immersive and interesting user experience.

10. Student Reflections

Over the course of the project, I've learned a lot of useful things and developed a variety of abilities, such as a deeper comprehension of thermodynamic principles, competency in microcontroller programming, and the capacity to clearly explain difficult scientific ideas using a user-friendly interface. Numerous problems were offered by the project, including sensor selection and data processing, which called for analytical thinking and problem-solving abilities. I was able to get through these challenges and complete the project's goals thanks to persistence and teamwork with my supervisor. I think that this experience has improved my academic talents and equipped me for future work in engineering and scientific research.

11. References

- Eitvilas, A. (2022). 6039CEM Preparation Progress Report. GG56 Computer Hardware and Software Engineering. Coventry University.
- Blundell, S. J., & Blundell, K. M. (2016). Concepts in thermal physics (2nd ed.). Oxford University Press.
- Adkins, C. J. (2013). Equilibrium thermodynamics (3rd ed.). Cambridge University Press.
- Kufre Esenowo Jack, Emmanuel O. Nwangwu, Israel Agwu Etu, & Ernest Ugwunna Osuagwu (2016). A Simple Thermistor Design for Industrial Temperature Measurement. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), 11(5), 57-66.
- Zhu, W., Liu, J., Yang, H., & Yan, C. (2015). Design of high precision temperature measurement system based on LabVIEW. International Journal of Advanced Computer Science and Applications, 6(6), 153-155.
- Zhang, X., Wang, C., Ma, J., & Ren, G. (2020). Control and synchronization in nonlinear circuits by using a thermistor. Modern Physics Letters B, 34(25), 2050267.
- Gaskell, D. R., & Laughlin, D. E. (2017). Introduction to the Thermodynamics of Materials (6th ed.). CRC Press.
- Hill, P. G., & Peterson, C. R. (1992). Mechanics and thermodynamics of propulsion. Reading.
- Mechem, D. (2013). An introduction to dynamic meteorology. Bulletin of the American Meteorological Society, 94(12), 1917-1920.
- Elliott, J. R., Lira, C. T., & Lira, C. T. (2012). Introductory chemical engineering thermodynamics (Vol. 668). Upper Saddle River, NJ: Prentice Hall.
- Microchip Technology Inc. (2007). PIC18F2455/2550/4455/4550 data sheet. https://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf
- NXP Semiconductors. (2018). MPX4250: Integrated pressure sensor. https://www.nxp.com/files-static/sensors/doc/data_sheet/MPX4250.pdf
- Amphenol Advanced Sensors. (2020). Thermometrics NTC thermistors: Type MA series. https://www.amphenol-sensors.com/hubfs/Documents/AAS-920-321E-Thermometrics-NTC-Type-MA-033020-web.pdf

- Analog Devices. (2000). AD595: Monolithic thermocouple amplifiers with cold junction compensation. https://www.sparkfun.com/datasheets/IC/AD595.pdf