# MIDDLE EAST TECHNICAL UNIVERSITY DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

# EE 447- Term Project – 2023 Fall

# Final Report

# Temperature Controller

**Group 23**

Mehmet Kurt 2443497

Çağlar Umut Özten 2375616

Mehmet Kurt 2443497
Çağlar Umut Özten 2375616

# Table of Contents

**Introduction**........3

**Components**........3

**System Overview**........4

**Operational Description & Code Analysis**........6

**Results and Discussions**........8

**Appendices**........9

**Conclusion**........9

Mehmet Kurt 2443497
Çağlar Umut Özten 2375616

**Introduction**

This report encapsulates the development and implementation of a temperature control system leveraging the TM4C123G microcontroller. The project's core aim involves integrating diverse hardware components, including the BMP280 sensor, Nokia 5110 LCD, resistive heating pad, and peltier, to create a functional temperature control mechanism.

The primary objectives revolve around continuous temperature monitoring within a specified environment, maintaining it within adjusted limits, and presenting real-time data through various output interfaces. Maintaining the environment's temperature within adjusted limits will be provided with peltier and resistive heating pad. Achieving this necessitates adeptness in hardware-software coordination, adhering to specific restrictions in both hardware utilization and programming languages.

This project serves as an exercise in fusing theoretical knowledge with hands-on application, emphasizing the significance of careful design and seamless integration of different hardware elements to craft a cohesive temperature control system.

The next sections elucidate the methodologies, challenges, and outcomes encountered during development, highlighting the strategies employed to fulfill project requisites within stipulated constraints. Furthermore, it explores the use of Serial Peripheral Interface (SPI) and I2C for efficient device communication, underscoring the proficiency in microcontroller-based system implementation.

**Components**

- Nokia 5110 LCD Screen
- BMP280 Temperature Sensor
- TM4C123G Board
- Potentiometer
- DC Supply
- 2 x BD237 Power Transistor
- 2 x 100 Ω resistor

Mehmet Kurt 2443497
Çağlar Umut Özten 2375616

**System Overview**

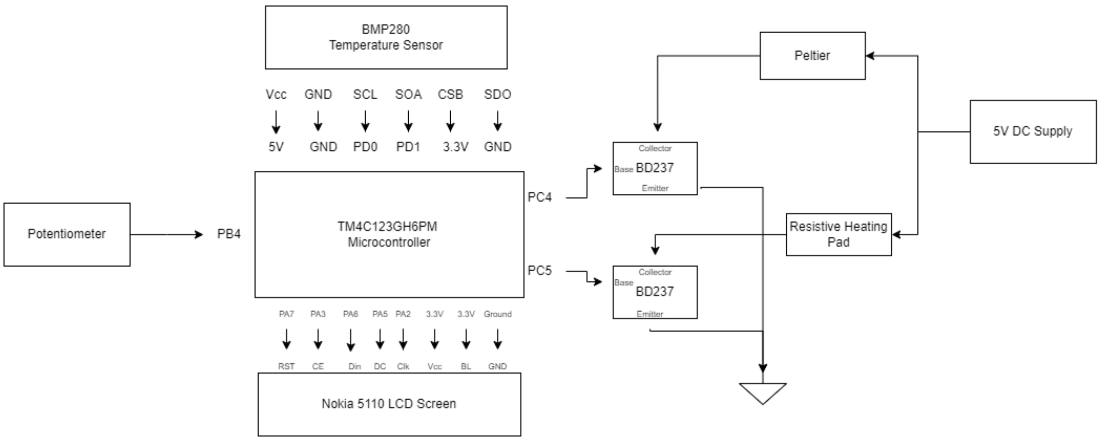



*Figure 1. General System Schematic*

The central objective of our system is the meticulous control and regulation of the ambient environment within a specified temperature threshold. This intricate task is achieved through a multifaceted integration of hardware components and precise software operations.

Continuous and instantaneous temperature monitoring within the environment is facilitated by the BMP280 Temperature Sensor. The sensor's data acquisition mechanism populates an array comprising 128 elements, enabling the calculation of real-time ambient temperature by means of data averaging. Essential pin configurations, imperative for the seamless operation of the sensor, are meticulously detailed within the source code, while a schematic overview of general pin connections is presented in Figure 1.

The determination of the desired ambient temperature range is intricately linked to the functionality of a potentiometer interfaced with the Tiva 4C's Analog-to-Digital Conversion (ATD) module. This potentiometer serves as a crucial interface translating voltage fluctuations ranging from 0 to 3.3V into a digital range spanning 0 to 4095. Further refinement and calibration involve a scaling operation wherein the acquired digital value is normalized and offset to precisely establish the high and low threshold temperature values relative to the desired ambient temperature.

Upon the precise calculation of the instantaneous ambient temperature and the establishment of the threshold parameters, the resultant values are meticulously presented and prominently displayed on the Nokia 5110 LCD Screen. This comprehensive visual representation serves to provide clear and immediate feedback regarding the system's temperature parameters, ensuring user comprehension and oversight.

Integral to the system's functionality are the LED indicators seamlessly integrated within the TM4C, designed to convey the system's temperature status concerning the prescribed range. A discernibly illuminated green LED denotes conformity with the desired temperature range, while the activation of either the blue or red LEDs signifies deviations from the specified range—above or below, respectively. This visual feedback mechanism augments user comprehension regarding the system's ambient temperature status.

Moreover, temperature regulation extends to the engagement of the resistive heating pad and peltier elements, orchestrated through precise and judicious transistor-based switching mechanisms employing BD237 power transistors. Pin assignments crucial for controlling these transistors are explicitly delineated in Figure 1. To ensure controlled current flow from the microcontroller, a 100-ohm resistor is strategically interposed between the base input and the pin. Simultaneously, the substantial current demands of these components necessitate the utilization of an external 5-volt power supply.

In instances demanding heating intervention, activation of the pin linked to the resistive heating pad's transistor base input is meticulously executed upon the system temperature's descent below the stipulated range. Correspondingly, the peltier, functioning as a cooling apparatus, is engaged when ambient temperatures surpass the desired range. Deactivation of these pins ensues upon the attainment of the targeted ambient temperature, ensuring precise and calibrated temperature stabilization. Elaborate circuit diagrams and connection configurations are meticulously outlined in Figures 2 and 3, elucidating the intricate integration of these hardware elements within the system.

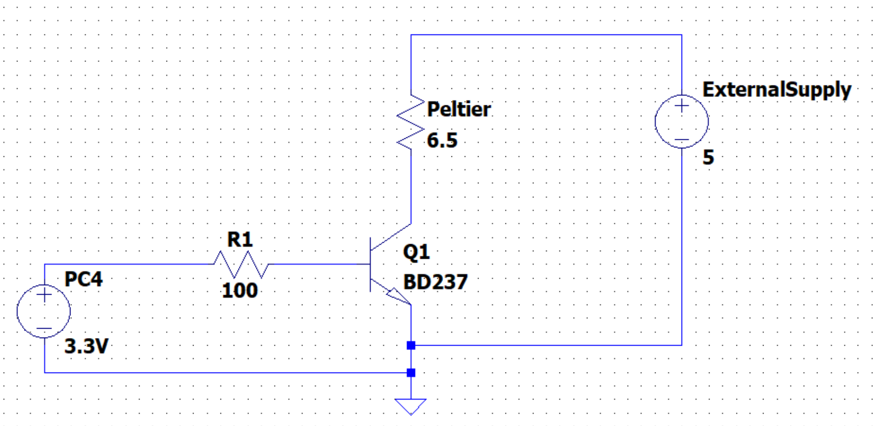



*Figure 2. Schematic of Peltier Device*

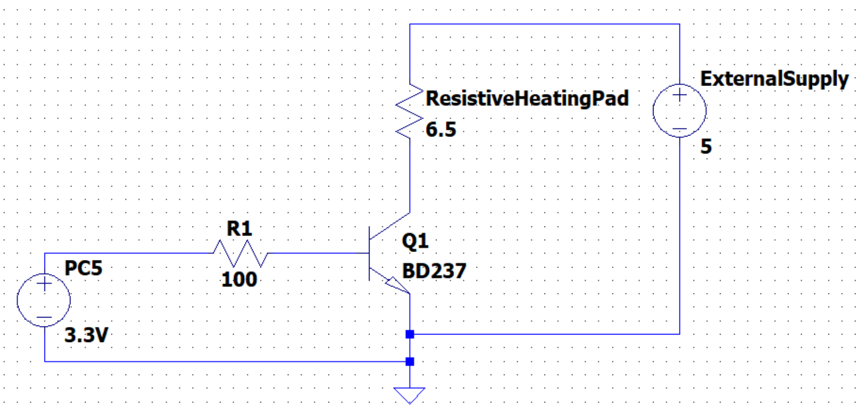
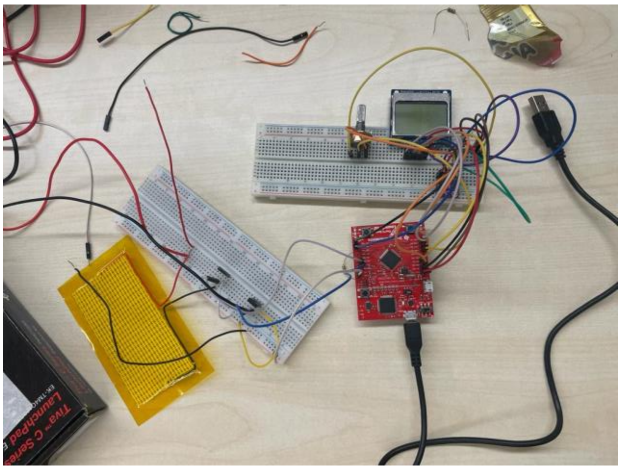



*Figure 3. Schematic of Resistive Heating Pad Device*

Mehmet Kurt 2443497
Çağlar Umut Özten 2375616

In summary, the meticulously detailed functionalities and operational procedures outlined above underscore the system's robustness and efficacy in precise regulation and control of the ambient environment within the preconfigured temperature parameters.



*General Design of the System*

**Operational Description & Code Analysis**

The temperature controller system begins its operation with an initialization phase. During this phase, the system sets up the BMP280 sensor and the Nokia 5110 LCD display. The sensor, crucial for accurate temperature readings, undergoes a calibration process to ensure precision. Simultaneously, the LCD display is prepared to show real-time data and system statuses. Once initialized, the BMP280 sensor continuously monitors the ambient temperature. This data is periodically read and processed by the microcontroller, ensuring up-to-date information on the environment's temperature. The central piece of the system's software, contained in mymain.c, plays a critical role here. It not only reads the sensor data but also applies any necessary adjustments or calibrations to ensure accuracy.

```
  // THESE ARE THE FUNCTIONS NECESSARY FOR CONFIG THE BMP280 AND READ DATA FROM IT
char I2C3_Write_Multiple(int slave_address, char slave_memory_address, int bytes_count, char* data);

char I2C3_Read_Multiple(int slave_address, char slave_memory_address, int bytes_count, char* data);
```

*Figure 4:Write and Read Code Pieces*

Thanks to these two functions, we can set the BMP280 sensor to the desired configuration and operate it. By using the Write_Multiple function, the values that need to be written to the specific slave memory address register inside the sensor are written. By using Read_Multiple, the desired measurement is read.

```
char data[2] = {0x00,0x27};
char datareset[1] = {0xB6};

  I2C3_Write_Multiple(0x76,0X0E,1,datareset); //Could be remove
  I2C3_Write_Multiple(0x76, 0XF5, 1,  data); // MSBdata
  I2C3_Write_Multiple(0x76, 0XF4, 1,  data+1); // MSBdata+1
```

*Figure 5:Usage of Write Multiple Function*

In the above photo, we can see the specific values are written to the specific registers in BMP280.

The user interface of the system, facilitated by the Nokia 5110 LCD, offers real-time insights into the system's performance. The display shows the current temperature, operational status, and possibly allows users to set desired temperature thresholds. This aspect of the system provides an interactive experience, allowing users to directly engage with and control the temperature settings.

The core of the system's functionality lies in its ability to control the temperature. The microcontroller, equipped with logic embedded in mymain.c, continuously compares the real-time temperature data with the user-defined setpoint. If the temperature falls below this setpoint, the system activates the resistive heating pad, gently raising the temperature. Conversely, if the temperature exceeds the setpoint, the Peltier device kicks in, reducing the temperature. This balancing act is crucial for maintaining a stable environment.

The operation of the temperature controller is a cyclic process. The main loop in the software keeps the system in a constant state of monitoring, decision-making, and action. It's a dynamic process, where real-time updates on the LCD display keep the user informed about the current temperature, the operational mode (whether heating or cooling), and any critical alerts.

Considering the code analysis; in the main, we firstly declare our temperature calculation and LCD related variables. Then we use Write functions to be able to configure the BMP280 sensor. Then, we are initializing the onboard RGB LED related modules, which are opening PORTF, setting PF1, PF2 and PF3 as output, enabling the clock for the GPIOF port.

After that, we are initializing the Nokia5110 LCD code. We are defining our LCD related functions in there. In the while loop which always loops, we are reading our ADC value from potentiometer. With this, we can then set our high and low thresholds into desired amounts.

```
//ADC VALUE READING 0-4095
static int get_value(void){
int x = 0;
ADC0->PSSI |= (1<<3);
  while((ADC0->RIS&(1<<3)) == 0){};
  if(ADC0->RIS==0x08){
  x = ADC0->SSFIFO3;
  }
  return x;
}
```

*Figure 6:ADC Reading Code*

Then, in crucial part, which is reading the raw temperature from sensor and converting into Celsius, we are collecting 128 samples from the sensor with the help of Read_Multiple function. Then, we are taking average temperature. But as in the below code it can be seen, some pre operations need to be done, these codes are directly taken from Bosch BMP280 Datasheet.

```
  while(i!=128)
{
  I2C3_Read_Multiple(0x76, 0XFA, 2,  data); // MSB
  value =(uint32_t) (data[0] << 12) | (data[1] << 4) | val3;
  var1=(((double)value)/16384.0-((double)dig_T1)/1024.0)*((double)dig_T2);
  var2=((((double)value)/131072.0 - ((double)dig_T1)/8192.0)*(((double)value
  t_fine=(long signed int)(var1+var2);
  T=(var1+var2)/5120.0;
  sum += T;
  i++;
}
average = sum / 128.0;


sum = 0;
i = 0;
```

*Figure 7:Raw Temperature to Actual Calculation*

After taking the actual average temperature value, we are basically comparing with the threshold values which could be determined by the user and us with the help of potentiometer. Then, if it is higher than the high threshold, we turn on the blue LED, and make the PC5 pin high, which means driving the peltier. If it is in the desired range, both PC5 and PC4 pin gets low output and nothing is driven, except for the green LED. If it is lower than the low threshold, this time we are turning on red led, and make the PC4 pin high, which means driving the heat pad.

Finally, with the help of Nokia5110 special functions, we could print out the temperature and threshold values into LCD screen. In below, SetCursor is determining the position of the written text. Then, we print out the values and texts.

```
Nokia5110_SetCursor(0, 0);
sprintf(str, " Temperature : %d", average);
Nokia5110_OutString(str);

Nokia5110_SetCursor(1, 3);
sprintf(adchigh, "High: %d", High_TH);
Nokia5110_OutString(adchigh);
//Nokia5110_OutString("L: 22 H: 30" );

Nokia5110_SetCursor(2, 4);
Nokia5110_OutString("Low: 22");
//sprintf(adclow, "Low: %d", Low_TH);
//Nokia5110_OutString(adclow);
```

*Figure 8 :LCD Printing Code*

A detailed working explanation can be found in the code comments.

**Results and Discussions**

A standout aspect of this project is the successful integration and utilization of the BMP280 sensor. Initially, one of the notable challenges faced was accurately reading data from this sensor. This difficulty presented a substantial hurdle, as the entire system's efficacy hinges on precise temperature measurements. Overcoming this challenge required a deep understanding of the sensor's workings, its communication protocol I2C and meticulous calibration processes. We received a lot of Hard Fault errors due to my connections, that is, jumper connections. Another difficulty was in translating sensor reading into assembly language. Unfortunately, we did not use it because we could not do it fully, we read the sensor directly in C.

**Appendices**

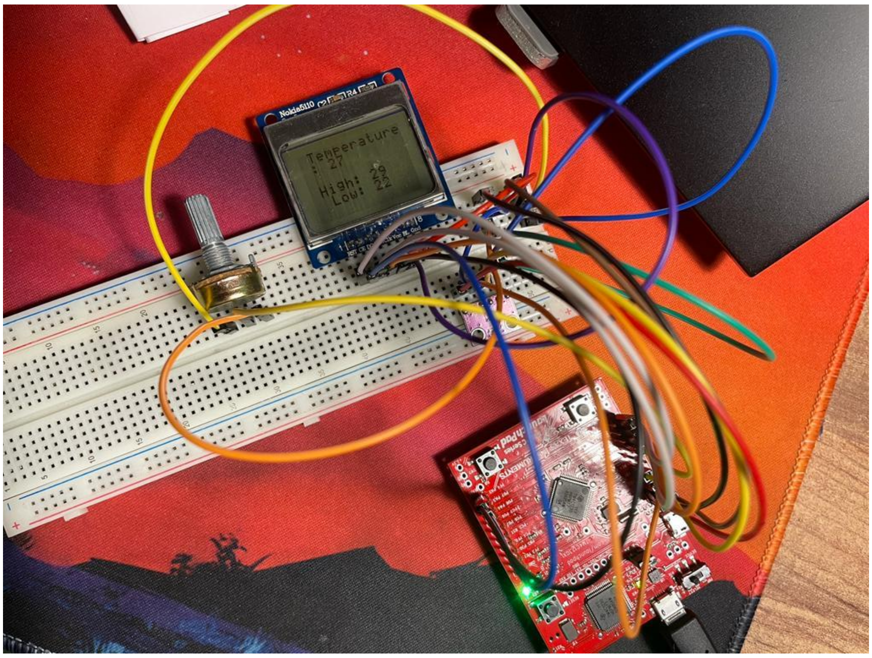



*Figure 9:Actual Photo Tiva-LCD-Pot*

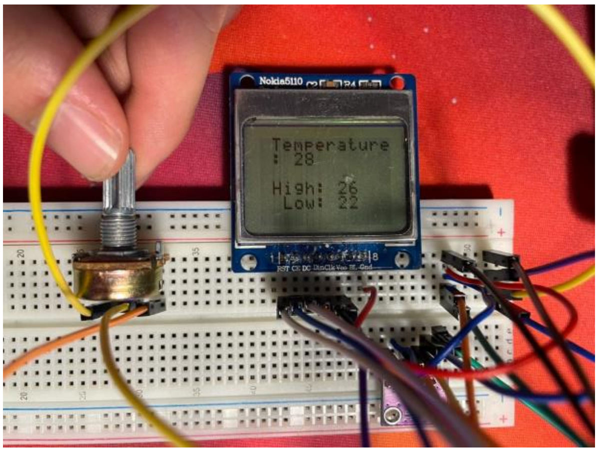



*Figure 10:High Threshold Set with the help of potentiometer.*

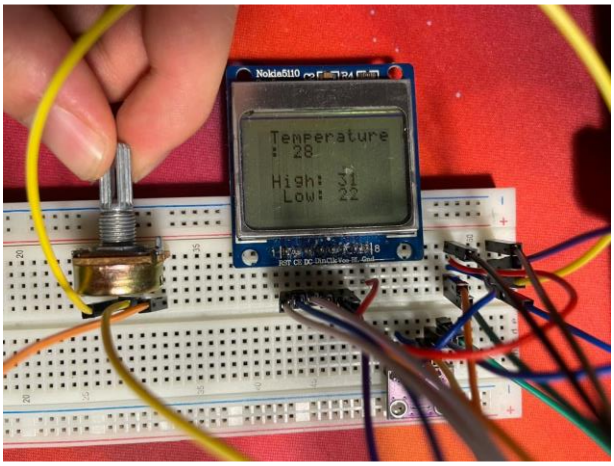



*Figure 11:High Threshold Set with the help of potentiometer.*

**Conclusion**

Our project aimed to create a reliable temperature control system using the TM4C123G microcontroller. Through careful integration of hardware components and software algorithms, we successfully developed a system capable of monitoring and regulating ambient temperature within predefined limits.

The BMP280 Temperature Sensor served as the backbone, providing real-time temperature data. By continuously sampling and averaging this data, we obtained accurate ambient temperature readings critical for system response.

The Analog-to-Digital Conversion (ATD) module coupled with a potentiometer allowed precise scaling and establishment of desired temperature ranges. This translated voltage readings into threshold temperatures, enabling the system to respond effectively to varying conditions.

Our user interface, comprising the Nokia 5110 LCD Screen and LED indicators, provided clear visualization of temperature status. Real-time temperature display and LED feedback empowered users to grasp the ambient temperature situation briefly.

Additionally, our system efficiently controlled the resistive heating pad and peltier elements, ensuring stable temperature regulation. Thoughtful use of external power sources and current regulation mechanisms enhanced system stability and safety.