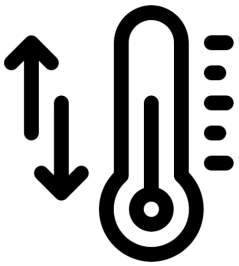
Ahmet Can YULAF 2376234

Bilal YUKSU 2110047

EE447: INTRODUCTION TO MICROPROCESSORS:

TEMPERATURE CONTROLLER DESIGN



INTRODUCTION

Our project is centered around constructing an efficient temperature control system that will employ a precision temperature sensor for real-time temperature data capture. The data will be used as input for the control system, which will regulate the environment within a cardboard box using both a resistive heating pad and a Peltier device. The integration of these two components will allow for flexibility in temperature adjustment and stability maintenance. To facilitate real-time monitoring and a user-friendly interface, we'll use the Nokia 5110 LCD, driven by the SPI module. Additionally, the project incorporates the use of GPIO pins for the onboard RGB LED, which will serve as a visual indicator of the system's operational status. We'll also implement pushbuttons through GPIO for user interactions, allowing for manual adjustments or system control interventions. By amalgamating these components, our goal is to design a robust and versatile temperature control system suitable for a variety of applications.

PROBLEM STATEMENT

In this project, it will be displayed the temperature level measured by the pressure and temperature sensor on the LCD screen and light up the proper LEDs on the microcontroller board. For that purpose, there are four input devices and four output devices in this project. Inputs to the microcontroller board are two push buttons, a BMP280 sensor and a potentiometer. Outputs are a Nokia LCD screen, onboard RGB LEDs, a resistive heating pad and a Peltier. Among those devices, the first one to be inspected is the BMP280 sensor. The initial step in the usage of the sensor is to establish serial communication between the microcontroller and the BMP280 sensor. This operation will be done by using I2C. The sensor will sample the temperature value in a 128 elements array and will send it to our microcontroller board. After the 128-dimensional array is filled, new data is received 3 times per second and the oldest data is deleted, that is, first in first out (FIFO). After attaining that input, we will calculate the mean of the sent data and it will represent the average temperature. Second device to be inspected is the potentiometer. By using the potentiometer, the user will set two threshold values on the temperature level. One of the thresholds is the lower threshold and the other is the higher threshold. In initial configuration, there will be some constant values (given by project members) and the user

can later change those thresholds if he/she wants to. If the average temperature value is below the lower threshold, the red LED in the microcontroller board will light up. If the value is between lower and higher threshold, the green LED in the microcontroller board will light up, and the last interval is above the higher threshold, in which the microcontroller board will light blue LED. Beyond all those, other external outputs which are heating pad and peltier will work based on the current temperature value. If temperature is below the lower threshold, the heating pad will warm up until the temperature is between the lower and higher threshold. Also, the peltier will cool down until the temperature is between the lower and higher threshold, if the temperature is above the higher threshold. Furthermore, the peltier and heating pad will be drived by using mosfet transistors since GPIO pins of the microcontroller board will not produce sufficient current to feed these outputs. Note also that the threshold setting procedure can be done by a potentiometer. The last unit in this project is the Nokia LCD screen. In this screen, the current temperature value and the lower & higher threshold values will be displayed. The current temperature value will be updated at every second, even if other variables do not change in that time interval. If other variables also change, those changes will also be illustrated on the LCD screen. Figure 1 shows the flowchart of the algorithm that will realize the temperature control.

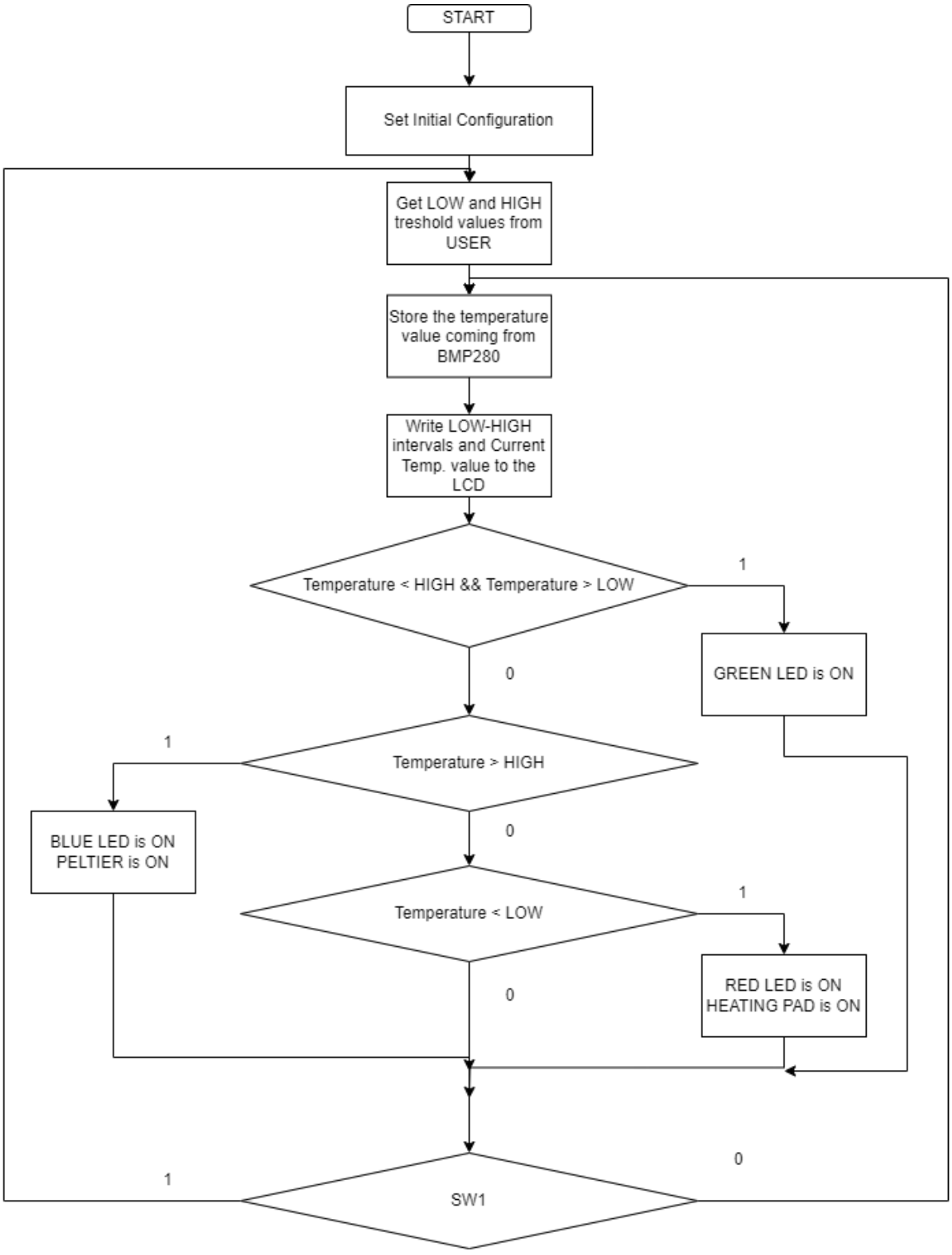


Figure 1. Flowchart of the algorithm

Sensing with BMP280

BMP280 is a temperature and pressure sensor manufactured by Bosch. It can communicate in two different interface, namely SPI and I2C. The communication between master, microcontroller -TM4C123GH6PM in this application- and the slave,BMP280. This sensor has a set of registers that holds data in it. For proper communication, the sensor is adjusted using its ctrl_meas with address 0xF4 and config with address 0xF5. By writing 0x23 and 0x3C respectively, the sensor is configured to operate at normal mode and be 16-bit sampling, 62.5ms of standby time and filter coefficient of 22. This values are determined according to the datasheet.

Temperature is measured by reading the addresses of 0xFA, 0xFB and most significant nibble of 0xFC. The read value is processed by the following the equations to obtain the temperature in Celsius.

```
var1 = (((double)adc_T)/16384.0 - ((double)dig_T1)/1024.0) * ((double)dig_T2);
var2 = ((((double)adc_T)/131072.0 - ((double)dig_T1)/8192.0) * (((double)adc_T)/131072.0 - ((double) dig_T1)/8192.0)) * ((double)dig_T3);
t_fine = (BMP280_S32_t)(var1 + var2);
T = (var1 + var2) / 5120.0;
```

Temperature read is done periodically with the help of Timer0A timer module. This timer takes around 3x20 bit ADC data from associated registers per second. The read ADC values are stored in a FIFO with size 128. Temperature value to be shown on the screen is determined by the average of ADC values in the FIFO buffer.

Temperature Control Mechanism

Control is quite simple. Operation start with the fulling the FIFO buffer with 20 bit ADC readings. Based on the average of this readings, celsius equivalent of the temperature is calculated. Initially, green led is on unless the thresholds are provided. Those thresholds can be set by the user by using the on-board push buttons. Based on the thresholds, system decides with subunit to operate. If measured average temperature is less than the low threshold, red led is turn on, this means that system needs to be heated. Thus, heating pad module is turned on. If it is higher than the high threshold, this means that the system needs to be cooled down. Thus, blue led is turned on. If these are not the cases, green led is turned on and no other module, except display and sensing is active.

Displaying Outputs

To facilitate the optimal performance of the Nokia 5110, a series of three key configurations are essential. Initially, GPIO settings are applied, with GPIO Port-B specifically utilized. Pins pB4, pB5, and pB7 are designated for Clk, CE, and Din, respectively. Additionally, pB2 and pB3 are configured for RST and DC, with VCC sourced from the microcontroller's 3.3V pin and GND connected to the microcontroller's ground.

Subsequently, adjustments to the SPI module on the TM4C123 are made to ensure seamless compatibility with the LCD. Since we used port-B for SSI, we started to SPI configuration by enabling the clock for SSI2 and then enters a loop to wait for the SSI2 peripheral to be ready, checking the status through the PRSSI register. Once the peripheral is confirmed ready, the SPI interface is disabled (CR1 set to 0x00) to allow for subsequent configuration. The clock rate of the SPI is then set by configuring the Clock Prescale Register (CPSR) to 0x02. The Control Register 0 (CR0) was adjusted (0x09C7) to specify parameters such as data size (8-bits), clock phase, and polarity. In this case, the Serial Clock Rate (SCR) was set to 9, indicating the desired clock frequency. The SPI is configured in Freescale mode. The Clock Configuration (CC) register is set to use the Precision Internal Oscillator (PIOSC). SSI operation is enabled (CR1 set to 0x02), and a series of no-operation instructions (__asm("NOP")) are included to provide additional time for stable configuration.

Lastly, during the NOKIA 5110 configuration phase, the display undergoes initialization for communication, positioning it to receive data for subsequent presentation. Initially, the reset pin (PB2) is set low, initiating a reset for the display, followed by a delay of 100 milliseconds to ensure a stable reset. Subsequently, the reset pin is set high, bringing the display out of reset. To enable command mode, the command/data pin (PB3) is set low. Extended command mode is activated by sending 0x21, followed by setting various parameters such as voltage operation, temperature limit, voltage bias value, and basic command mode activation using the SendData function. The display is configured for normal mode with a specified position (X=0, Y=0), and then the command/data pin is set high, indicating the switch to data mode. This sequence ensures proper initialization and configuration of the Nokia 5110 LCD for subsequent data display.

After the necessary adjustments are made, common mode is enabled by first setting the pB3 pin low to send the bits in the desired location of the screen. Then, the X and Y axis are set for the origin. Finally, the pB3 pin is made high again and data mode is enabled and the desired data is sent to be printed on the screen as 8 bits.Using the left side of the screen, Low Threshold, High Threshold and Current Temperature were printed on the LCD screen as LT, HT and CT respectively. On the right side of the screen, a graph of temperature, T versus time, s was drawn and temperature data was taken and recorded every second on the graph. Some outputs are shown in Figure 2.

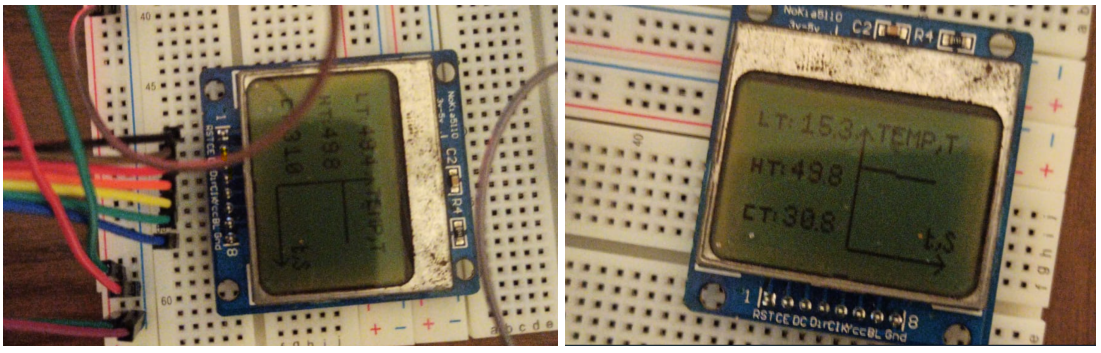


Figure 2. displaying temperature vs time graph on Nokia5110 LCD Screen

Conclusion

To sum up, the integrated temperature controller system using TM4C123GH6PMI combines a precision temperature sensor, resistive heating pad, and Peltier device to ensure adaptable and stable temperature regulation. Employing a Nokia 5110 LCD for real-time monitoring and onboard RGB LEDs for visual cues enhances user interaction. Users have the flexibility to set lower and upper-temperature thresholds using a potentiometer, with the microcontroller processing this input to calculate the average temperature. Distinctive LEDs in red, green, and blue signify temperature intervals, while the heating pad and Peltier respond to temperature changes, skillfully regulated by MOSFET transistors. The LCD screen dynamically updates temperature values and thresholds, providing a comprehensive overview. In conclusion, the system delivers a user-friendly interface and versatile functionality, ensuring effective temperature control across various applications.