i. Front page

Coursework 2 – Human Machine Interface

Coventry University

KH5048CEM

Embedded System Design and Development

Adham Abdallah

CU1901014

BEng Electrical and Electronics Engineering

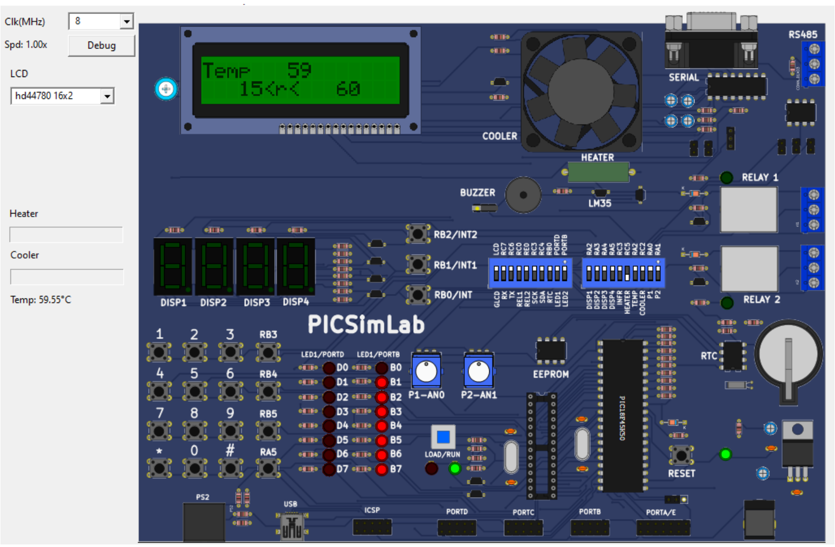
2021 - 2022

ii. Table of contents

1.0 Introduction

“The EasyPIC v8 is a development board designed for the rapid development of embedded applications. Using it with a series of other materials such as resistors, breadboards, etc.. One can achieve multiple application with it. One industrial application that can be created is the human machine interface. Where we could use the LCD to display temperatures gained from the temperature sensor and then compare the gained temp. between the different limits that you set. Which is precisely what will be shown in the following report.

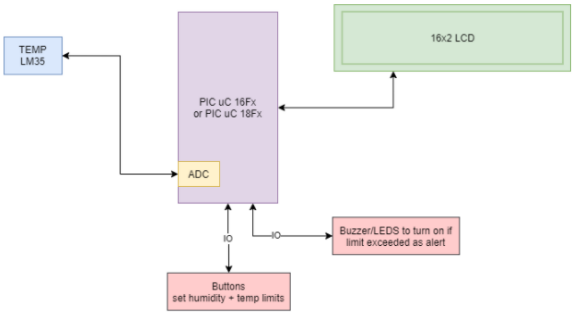
2.0 System Design Architecture



Clk(MHz) 8
Spd: 1.00x
LCD hd44780 16x2
Temp: 59.55°C

```
Temp     59
  15<r<    60
```

As we can see in the diagram above of the Picsimlab, we have a lot of different components, such as the LCD which is the green rectangle with the temp displayed on it, the heater button that could be turned on and off, the different RB buttons that control the increase and decrease of the specified limit. The PICSIMLab takes the HEX file saved from the MIKROCPRO program and loads it on the system and runs the code in sequence. In addition, the code is written on MIKROCPRO program where it first checks for any errors and then you could build and program it on the PICSIMLab and all the libraries such as conversions and LCD that are necessary are selected. A layout of the code logic system can be seen in the below diagram..



TEMP
LM35
16x2 LCD
PIC uC 16Fx
or PIC uC 18Fx
ADC
IO
IO
Buzzer/LEDS to turn on if limit exceeded as alert
Buttons
set humidity + temp limits

2.1 Bill of Materials

Item: Arduino Jumper Wire Set

Quantity bought: (40 Jumper)

Quantity used: 6

Price per set: 22 Egyptian pound

Description:

- Handy for making wire harnesses or jumping between headers on PCB's
- Current Rating up to 1A
- Mixed Colors
- Length 20 cm
-

SKU: PH61.MM.20CM

Link to website: https://ram-e-shop.com/product/ph61-mm-20cm/

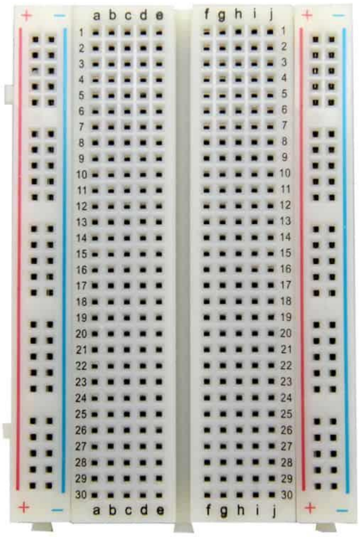
Item: Bread Board 630-Tie Point "BB-01"

Quantity: 1

Price per unit: 30 Egyptian pounds

Description:

- 1 Terminal Strip
- Tie Point 630
- 2 Distribution Strips
- Tie Points 200.
- ABS Plastic material
- Pitch: 2.54mm
- Rated: 300V/3-5A
- Wire Range: 20-29AWG



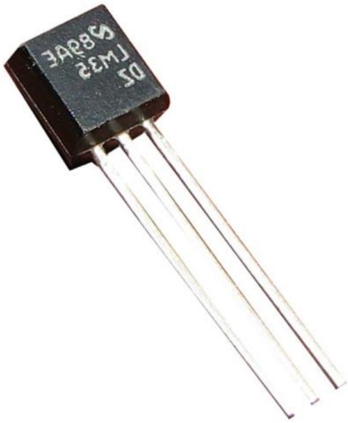
SKU: BB01.BREAD.BOARD

Link to website: https://ram-e-shop.com/product/bb01-bread-board/

Item: LM35dz "Temperature Sensor"

Quantity: 1

Price per unit: 25 Egyptian pounds



Description: Linear + 10.0 mV/˚C scale factor

0.5˚C accuracy guaranteeable (at +25˚C)

Rated for full −55˚ to +150˚C range

Suitable for remote applications

Low cost due to wafer-level trimming

Operates from 4 to 30 volts

SKU: LM35DZ

Link to website: [LM35dz "Temperature Sensor" - RAM Electronics (ram-e-shop.com)]

Item: EasyPIC v8

Quantity: 1

Price per unit: 249 USD

Description: The EasyPIC v8 is a development board designed for the rapid development of embedded applications, based on 8-bit PIC microcontrollers (MCUs). • The development board is divided into several sections, arranged so that all the related interactive components such as switches, buttons, indicators, and connectors, are logically positioned and grouped together

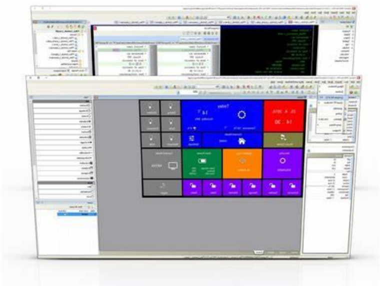
Manufacturer: Mikro Elektronika

Link to website: https://www.mikroe.com/easypic

Item: mikroC PRO for PIC

Quantity: 1

Price per unit: 269 USD



Description: a full-featured ANSI C compiler for PIC devices from Microchip. It is the best solution for developing code for PIC devices.

It features intuitive IDE, powerful compiler with advanced optimizations, lots of hardware and software libraries, and additional tools

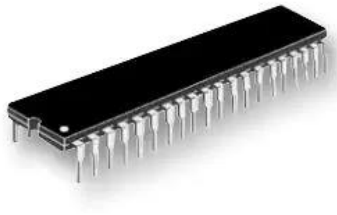
Manufacturer: Mikro Elektronika

Link to website: https://www.mikroe.com/mikroc-pic

Item: PIC18F47K42 Microcontroller Chip

Quantity: 1

Price per unit: $3.17



Description: DIP40 microchip (has 40 pins). Clock frequency: 64MHz. Program memory: 128kB. Supply voltage: 2.3 - 5.5V DC. Harvard 8bit Architecture.

Find the rest of the info attached in the data sheet: https://www.tme.eu/Document/475e508ddf619bca87f4a27ea9509268/PIC18F47K42.pdf

Manufacturer: Microchip Technology

Link to website: https://www.tme.eu/en/details/pic18f47k42-i_p/8-bit-pic-family/microchip-technology/

Item: USB-C to USB-C 2.0 cable with adapter to USB 3.0 type A Male

Quantity: 2

Price per unit: 9.00 USD

Description: This is a USB 3.1 cable with USB Type C male connectors on both sides. Cable comes with USB Type C Female to USB Type A Male adapter, making it compatible with USB 2.0 and USB 3.0 host slots as well. Cable is 1.5 meters long and compatible with the new 8th generation of Mikroe development tools.

Manufacturer: Mikro Elektronika

Link to website: https://www.mikroe.com/usb-c-to-usb-c-20-cable-with-adapter-to-usb-30-type-a-male

Item: LCD mini click

Quantity: 1

Price per unit: 15.20 USD

Description: LCD mini click displays 2x16 monochrome characters on an LMB162XFW LCD display. It features the MCP23S17 port expander and the MCP4161 digital potentiometer, both from Microchip.

Manufacturer: Mikro Elektronika

Link to website: LCD mini click — displays 16x2 monochrome characters on a LCD display (mikroe.com)

Item: Buzzer

Quantity: 1

Price per unit: 7.20 USD

Description: BUZZ Click is an accessory board in mikroBus™ form factor. Board features a piezo speaker capable of emitting audio signals.Buzzer's resonant frequency is 3.8kHz (where you can expect it's best performance).

Manufacturer: Mikro Elektronika

Link to website: Buzz click - Breakout board for Piezo Buzzer (mikroe.com)

There are multiple ways to have a cost effective system, one of which is the use of cheaper microcontrollers that would decrease the cost drastically, controllers such as Arduino are a good alternative. In addition, to using a different temperature sensor that don't have a very high limit like the one used here.

3.0 Software Design, Testing, and Implementation

3.1 Software Explanation

The software can be broken down into multiple segments. Starting of with the main LCD module connections that is used to assign the required pins on the PICSIMLab. Then, comes the role of defining all the variables that are going to be used throughout the code. Variables such as high (which is the upper limit), low (which is the lower limit), e (the variable that stores the data gained from the temperature sensor) and finally b (which converts the value gained from the temperature sensor to degrees Celsius). In addition to, empty space being left under high1,low1,etc. to store the different temp inputs such as (the changeable upper and lower limit).

Following these connections, the Void main which is the main function is created. Where the different ports that are used are configured as either analog or digital and set to be an input or an output. The LCD and the ADC (analog digital converter) are initialized, and the heater is turned on using the function RC5_BIT=1. Then, the LCD is cleared and displays the different texts set. In addition, to the upper and lower limit being set at 15 degrees Celsius and 60 degrees Celsius.

Lastly, the different conditions are implemented. Where, for example if the temperature exceeds the upper limit. The buzzer turns on.The code used to implement the system is written below along with the comments in each line, explaining the code precisely (written in green). Furthermore, please refer to the video for the implementation of this code on the simulation.

https://elsewedyedu1-my.sharepoint.com/:v:/g/personal/adham_abdallah_tkh_edu_eg/EfNJROp6uvpEjWyWOT2c16UBOTLCk4pOQrpk1-c6btlU8Q?e=Qa5VyG

```
// LCD module connections  (set for the simulation kit)

sbit LCD_RS at RE2_bit;

sbit LCD_EN at RE1_bit;

sbit LCD_D0 at RD0_bit;

sbit LCD_D1 at RD1_bit;

sbit LCD_D2 at RD2_bit;

sbit LCD_D3 at RD3_bit;

sbit LCD_D4 at RD4_bit;

sbit LCD_D5 at RD5_bit;

sbit LCD_D6 at RD6_bit;

sbit LCD_D7 at RD7_bit;


sbit LCD_RS_Direction at TRISE2_bit;

sbit LCD_EN_Direction at TRISE1_bit;

sbit LCD_D0_Direction at TRISD0_bit;

sbit LCD_D1_Direction at TRISD1_bit;

sbit LCD_D2_Direction at TRISD2_bit;

sbit LCD_D3_Direction at TRISD3_bit;

sbit LCD_D4_Direction at TRISD4_bit;

sbit LCD_D5_Direction at TRISD5_bit;

sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
// End LCD module connections

unsigned char e;       //defining the variable e
unsigned char b;       //defining the variable b
unsigned char low;     //defining the variable low
unsigned char high;    //defining the variable high

char temp[10];        // you leave a space for a text under the value temp
char low1[10];        // you leave a space for a text under the value low1
char high1[10];       // you leave a space for a text under the value high1

 void main()         // main function
{
ANSELA=0xff;         //configure port A pins as analog
TRISA=0xff;          // port A is set to be an input
ANSELB=0;            //configure port B pins as digital
TRISB=0xff;          // port B is set to be an input
ANSELC=0;            //configure port C pins as digital
TRISC=0;             // port C is set to be an output
ANSELD=0;            //configure port D pins as digital
TRISD=0;             // port D is set to be an output
ANSELE=0;            //configure port E pins as digital
TRISE=0;             // port E is set to be an output

Lcd_Init();          // Initializing the LCD
ADC_Init();          // Initializing the analog digital converter
Lcd_Cmd(_LCD_CLEAR);      // all text on the LCD display is cleared
Lcd_Cmd(_LCD_CURSOR_OFF);  // cursor is turned off


RC5_bit=1;                //turn heater on

low = 15;                //setting the lower limit

high = 60;               //setting the upper limit


Lcd_Out(1,2,"Embedded Task#2");      //write text in first row

Lcd_Out(2,1,"by AdhamAbdallah");     //write text in second row

Delay_ms(2500);            // delay of 2500 milliseconds

Lcd_Cmd(_LCD_CLEAR);       // all text on the LCD display is cleared

Lcd_Out(1,1,"Temp");      // Prepare and output static text on LCD

Lcd_Out(2,6,"<r<");      // Prepare and output static text on LCD



while (1) {                // continuous endless loop

e= ADC_Read(2);           // read temperature from the sensor pin and save in variable z

b= 5*e/10.24;        // converting the temperature into degree Celsius and saving it in c

WordToStr(b, temp);   // converting the variable to sting

Lcd_Out(1,5,temp);     // Prepare and output static text on LCD


WordToStr(low, low1);     // converting the variable to sting

Lcd_Out(2,1,low1);       // converting the variable to sting

WordToStr(high, high1);   // converting the variable to sting

Lcd_Out(2,9,high1);       // converting the variable to sting

 if (b >= high){      // setting if conditions.

   RC1_bit=1;  // if the value of temp increases more than the +limit turn buzzer on

 }

 if (b <= high){      // setting if conditions.
```

```
   RC1_bit=0;  // if the value of temp is less than the +limit turn buzzer off
 }
 if (b <=low){               // setting if conditions.
   RC1_bit=1;  // if the value of temp decreases more than the -limit turn buzzer on
 }
 if (RB2_bit==1){   //if RB2 is pressed on
  high = high-1;     // add +1 to the upper limit
 }
   if (RB1_bit==1){    //if RB1 is pressed on
  high= high+1;      //Subtact-1 from the upper limit
 }
 if (RB0_bit==1){       //If RB0 is pressed on
  low=low-1;            //Add +1 from the lower limit
 }
 if (RB3_bit==1){        //if RB3 is pressed on
  low=low+1;               //subtract -1 from the lower limit
 }
 Delay_ms(50) ;         //delay of 50 milliseconds


}
}
```

3.2 Software Implementation on GIT

Using my personal account on GitHub. I created a new repository under the name of Embedded-Task#2 and the owner name AdhamAbdallah. Where I made the access to it public and copied the link down below”

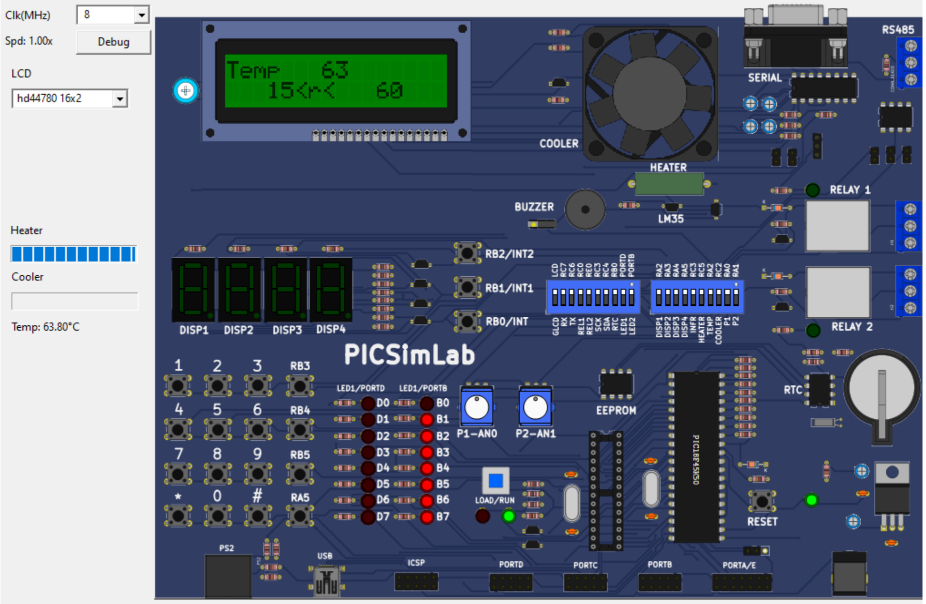
AdhamAbdallah/Embedded-Task-2 (github.com)

3.3 Software Test Cases

TEST CASE 1

Condition 1 : Value detected by the temperature sensor is more than the upper limit set.

As seen in the figure below, the current temperature read by the sensor is 63 degrees Celsius which is more than the specified upper limit initially set to 60 degrees Celsius. When this occurs, the buzzer is turned on meaning RC1_BIT=1. Bear in mind, that the temperature has risen due to the heater being turned on.



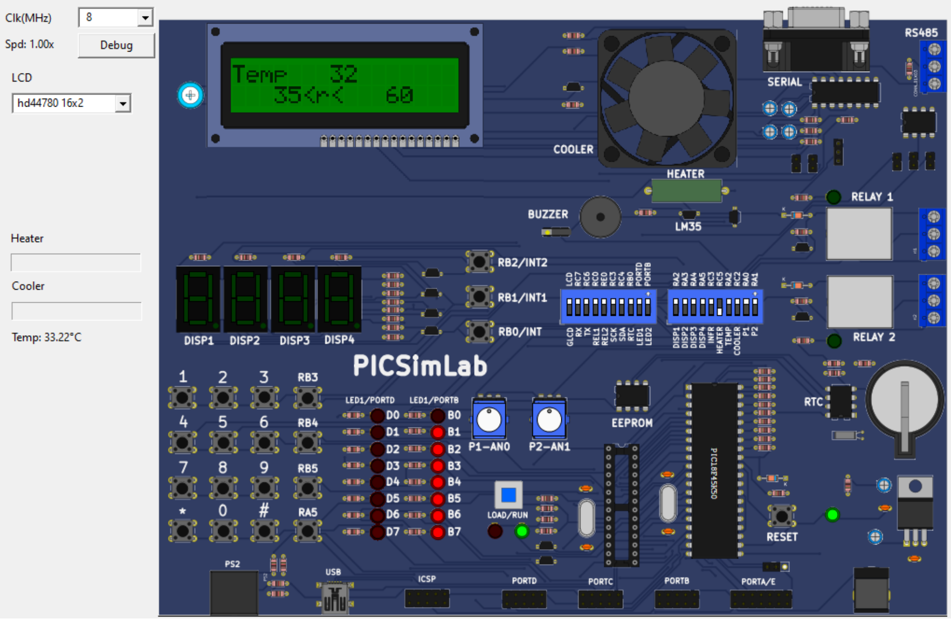
Clk(MHz) 8
Spd: 1.00x
LCD hd44780 16x2
Heater
Cooler
Temp: 63.80°C

```
Temp    63
 15<r<    60
```

TEST CASE 2

Condition 2: Value detected by the temperature sensor is less than the lower limit set.

As seen in the figure below, the current temperature read by the sensor is 32 degrees Celsius which is less than the specified lower limit set to 35 degrees Celsius (manipulated the lower limit by pressing on RB0 which adds to the lower limit. So instead of the lower limit being 15 it is now 35) (This is discussed in the below test cases). When this occurs, the buzzer is turned on meaning RC1_BIT=1.



Clk(MHz) 8

Spd: 1.00x

Debug

LCD hd44780 16x2

Heater

Cooler

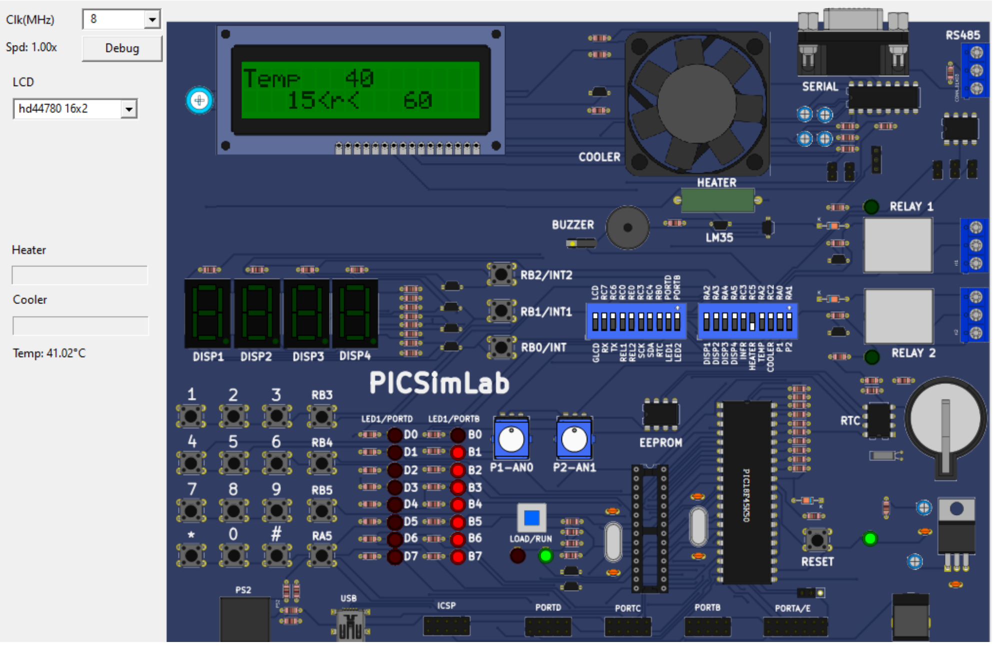
Temp: 33.22°C

```
Temp     32
  35<r<    60
```

TEST CASE 3

Condition 3: values detected by the temperature sensor is within the specified range

As seen in the figure below, the current temperature read by the sensor is 40 degrees Celsius which is in the specified range of 15<temp<60. When this occurs, the buzzer is off meaning RC1_BIT= 0. Furthermore, if the temp was more than the upper limit or less than the lower limit, the buzzer will still be on until it reaches the specified range. When it does buzzer=off= no sound emitted.



Clk(MHz) 8

Spd: 1.00x

LCD hd44780 16x2

Temp: 41.02°C

```
Temp    40
 15<r<    60
```

TEST CASE 4

Condition 4: Value of the specified upper limit need to be increased

The upper limit is always initially set to 60 degrees Celsius, but, in the case that its value needs to be increased. The RB2_BIT is pressed on. Each push will add 1 degree Celsius to the upper limit or one could press once continuously, and the value of the upper limit will increase exponentially until you take of your finger when you reach your desired upper limit.

Please refer to the video below to see the implementation of test case 4 and 5.

https://elsewedyedu1-my.sharepoint.com/:v:/g/personal/adham_abdallah_tkh_edu_eg/ERbVU04uLLZHpNJ4hOn3WaQBlTH1vhGnZyGwErXXruL2Fw?e=OQzbPV

TEST CASE 5

Condition 5: Value of the specified upper limit need to be decreased

The upper limit is always initially set to 60 degrees Celsius, but, in the case that its value needs to be decreased. The RB1_BIT is pressed on. Each push will subtract 1 degree Celsius from the upper limit or one could press once continuously, and the value of the upper limit will decrease exponentially until you take of your finger when you reach your desired upper limit.

Please refer to the video below to see the implementation of test case 4 and 5.

https://elsewedyedu1-my.sharepoint.com/:v:/g/personal/adham_abdallah_tkh_edu_eg/ERbVU04uLLZHpNJ4hOn3WaQBlTH1vhGnZyGwErXXruL2Fw?e=OQzbPV

TEST CASE 6

Condition 6: Value of the specified lower limit need to be increased

The lower limit is always initially set to 15 degrees Celsius, but, in the case that its value needs to be increased. The RB0_BIT is pressed on. Each push will add 1 degree Celsius to the lower limit or one could press once continuously, and the value of the lower limit will increase exponentially until you take of your finger when you reach your desired lower limit.

Please refer to the video below to see the implementation of test case 6 and 7.

https://elsewedyedu1-my.sharepoint.com/:v:/g/personal/adham_abdallah_tkh_edu_eg/EaP6XRtUe2lDiUihH4IkvegBT4jcocpWZVFxHMG5xTOauw?e=74nTXg

TEST CASE 7

Condition 7: Value of the specified lower limit need to be decreased

The lower limit is always initially set to 15 degrees Celsius, but, in the case that its value needs to be decreased. The RB3_BIT is pressed on. Each push will subtract 1 degree Celsius from the lower limit or one could press once continuously, and the value of the lower limit will decrease exponentially until you take of your finger when you reach your desired lower limit.

Please refer to the video below to see the implementation of test case 6 and 7.

https://elsewedyedu1-my.sharepoint.com/:v:/g/personal/adham_abdallah_tkh_edu_eg/EaP6XRtUe2lDiUihH4IkvegBT4jcocpWZVFxHMG5xTOauw?e=74nTXg