UNIVERSITI MALAYSIA SABAH

FACULTY OF ENGINEERING (FKJ)

ELECTRONIC ENGINEERING (COMPUTER)

UH6523002/HK20

KS30503 EMBEDDED SYSTEM

SEMESTER I SESSION 2021/2022

INDIVIDUAL MINI PROJECT

Lecturer: Dr. Mazlina Mamat

NAME	MATRIC NO.
ABDUL RAHIM BIN SAIDI	BK19110293

TABLE OF CONTENTS

	Page
TASK 1	**1**
1.1 Control Panel's Picture	1
1.2 Optimal Microcontroller(MCU) for the Control System	2
TASK 2	**3**
2.1 Suitable device/component based on Block Diagram	3
2.2 Datasheet, Estimation Total Application Power, Estimation Heat Produced & System Connectivity's	4
TASK 3	**9**
3.1 Circuit of Electric Thermos Pot 3.0L using Proteus	9
3.2 Flowchart & C Programming	9
3.3 Simulation	14
3.4 Improvement of the Embedded System	18
TASK 4	**27**
4.1 Constructed Real Hardware of the System	27

TASK 1

1.1 Control Panel's Picture

The control panel's picture of an embedded system that is available in my house is Electric Thermos Pot 3.0L which shown below:

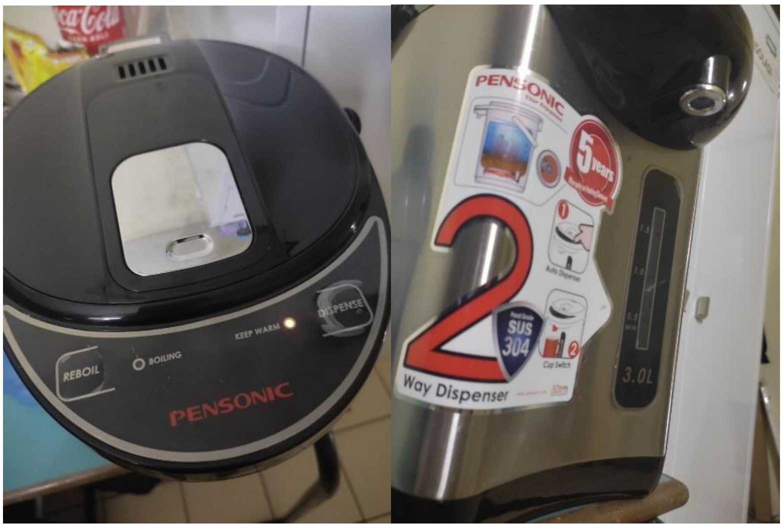

Figure 1.0 Control Panel's Electric Thermos Pot 3.0L

The derived block diagram is shown below:

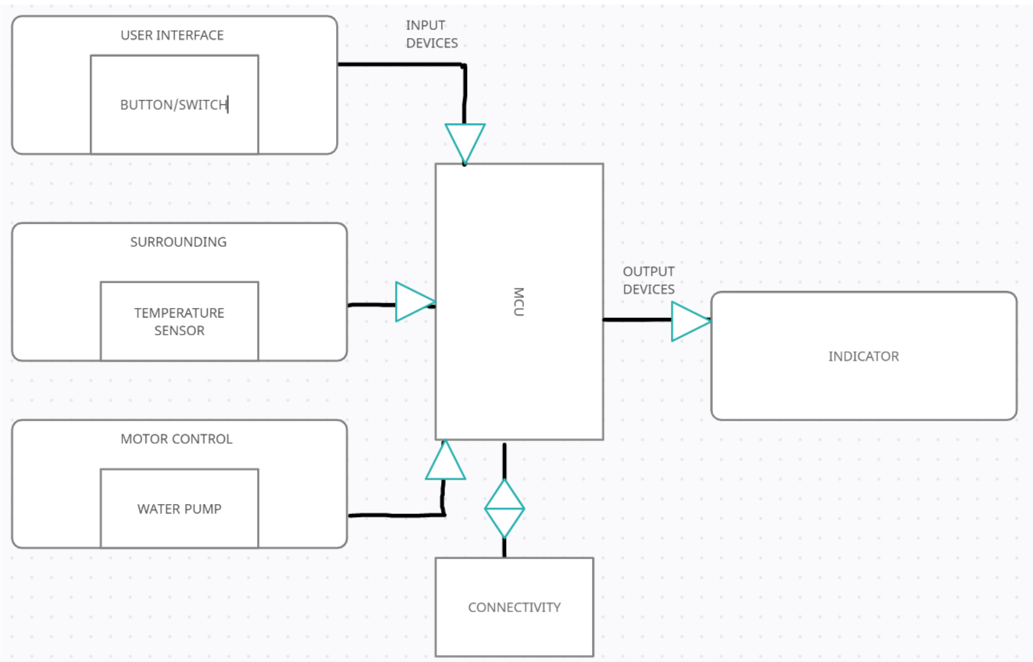

Figure 1.1 Block Diagram Electric Thermos Pot 3.0L

1.2 Optimal Microcontroller(MCU) for the Control System

There a lot of microcontroller out there but there are one MCU that always I used on sharping my skills on embedded circuit of a certain projects which is PIC16F887.

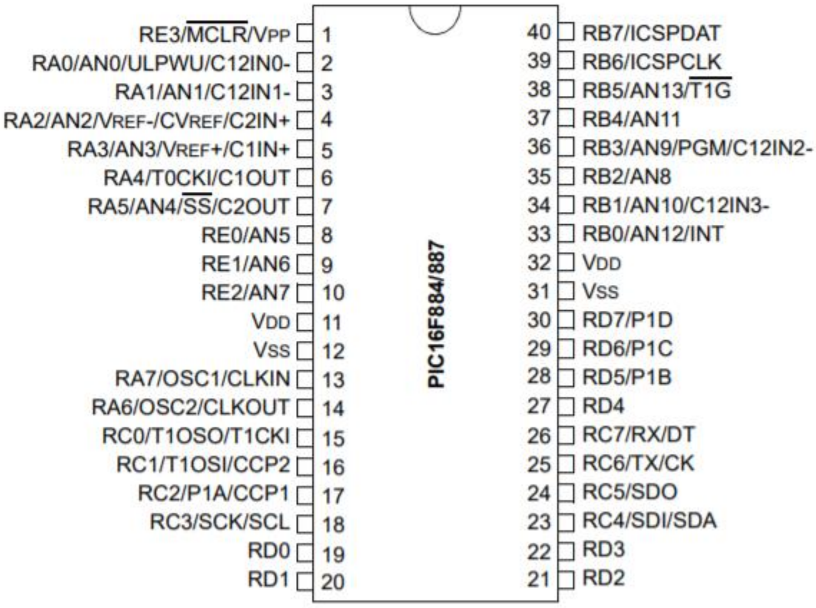

Figure 1.3 Pin Diagrams PIC16F887

As you can see, 35 I/O pins can be used in PIC16F887 which is more than enough if compare with **Figure 1.0** Control Panel's **and Figure 1.1** Block Diagram which use only a *Minimum* of 9 pins only on the MCU. The datasheet, states that the MCU can run on a high sink/source current which is up to 25mA. It also has a Power-on Reset (POR) which is important for the safe measurement of the systems. The temp range is between -40 and 125 degrees Celsius which is suitable enough for a heating system that is up to 100 degrees Celsius usually. We have already touched on some features that may require for the Electric Thermos Pot 3.0L systems. For more detail of the PIC16F887 MCU open the link provided below.

Datasheet: https://ww1.microchip.com/downloads/en/DeviceDoc/40001291H.pdf

TASK 2

2.1 Suitable device/component based on Block Diagram

- For the user interface part which is the keypad, I suggest using the push button such as 'Tactile Switches'. The reason behind it is commonly used anywhere and can be easily found. On technical features, we just need to push the button to turn on some features on the systems not to mention the low power rating of the components.
- For the surrounding part which is the temperature sensor, I suggest using the temperature sensor LM35 which is small in physical also cheaper, and commonly used in the industries. The low power rating of the components is also one of the reasons to choose the components.
- For the motor control part which is the water pump, I suggest using 12V 225RPM 1.3kgfcm Brushed DC Geared Motor because of the smaller size also cheaper than the other 12V DC motor. It also has a low power rating.
- For connectivity connection, I suggest using AC Cord because the power source travels farther very efficiently. It's also very important to have enough power supply to run the system efficiently especially for the heating element which consumes a little bit larger in power rating.
- For the output device which is the only one, the Indicator, we use the LEDs. As from the datasheet of PIC16F887 where the pins are specified to drive (source) up to 20mA and sink (pull the output to ground) 25mA. These current capabilities easily allow the MCU to drive LEDs. From the control panel, we need two different LEDs with different colors which indicates boiling & dispense.

Figure 2.0 (From Left to Right) Tactile Switch, Temperature Sensor LM35, DC Motor, AC Cord, LEDs Light

2.2 Datasheet, Estimation Total Application Power, Estimation Heat Produced & System Connectivity's

Assume that all value that was taken for calculation below are in max value.

Estimation Total Power

- Datasheet for Tactile Switch is shown below:

■Specifications

Item	Specification	Unit	Note
Operating Temperature Range	-20～+70	℃	
Storage Temperature Range	-40～+85	℃	
Type of Operation	Tactile Feedback		
Circuit Configuration	Push-On Momentary, 1 pole-1 throw		
Power Rating	MAX 50mA 24VDC		
Contact Resistance	MAX 100	mΩ	
Insulation Resistance	100MΩ Min. at 100VDC		
Dielectric Withstanding Voltage	250VAC for 1 minute		
Contact Bounce	MAX 5	ms	
Operating Force	0.98±0.49	N	A type
	1.57±0.49	N	B type
Return Force	MIN 0.10	N	A type
	MIN 0.49	N	B type
Travel	0.25 +0.2/-0.1	mm	

Figure 2.1 Specification of Tactile Switch

As we can see from the **Figure 2.1,** the power rating of Tactile Switch is at Max 50mA when DC supply voltage at 24V.For more info of the datasheet click the link below:

Datasheet: https://components101.com/sites/default/files/component_datasheet/Push-Button.pdf

Tactile Switch Power = (24V x 50mA) = 1.2W

Since there are 3 buttons from **Figure 1** for the control panel's hence,

Total Tactile Switch Power = 1.2W x 3 = 3.6W

- Datasheet for Temperature Sensor LM35 is shown below:

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)(2)

		MIN	MAX	UNIT
Supply voltage		–0.2	35	V
Output voltage		–1	6	V
Output current			10	mA
Maximum Junction Temperature, $T_J$max			150	°C
Storage Temperature, T_{stg}	TO-CAN, TO-92 Package	–60	150	°C
	TO-220, SOIC Package	–65	150	°C

Figure 2.2 Specification of Temperature Sensor LM35

From **Figure 2.2**, at maximum value, the output current is 10mA and the supply voltage is 35V.For more info of the datasheet click the link below:

Datasheet:
https://www.ti.com/lit/ds/symlink/lm35.pdf?ts=1640063252518&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM35

Temperature Sensor LM35 Power = 10mA x 35V = 0.35W

- Datasheet for 12V 225RPM 1.3kgfcm Brushed DC Geared Motor is shown below:

参数 Details:	减速电机参数(Geared motor specification):	
	空载电流(No Load Current)(mA):	≤90
	空载转速(No Load Speed)(r.p.m):	225±22
	额定力矩(Rated Load Torque)(kgf.cm):	1.3
	额定电流(Rated Current)(mA):	≤570
	额定转速(Rated Load Speed)(r.p.m):	170±17
	堵转电流(Stall Current)(mA):	
	堵转力矩(Stall Torque)(kgf.cm):	
	旋转方向(Rotation Direction):	CW/CCW
备注(Remarks):		

注意事项(Notes):
1,焊接作业请在短时间内实行。（推荐：焊接头的温度340－400度，2秒内实行） Please conduct the welding work in a short time (Recommendation:With the soldering iron tip at a temperature of 340－400℃， within 2 seconds
2,在安装过程中不应使减速电机受到冲击、跌落。不允许直接在减速器出轴上敲打或压配。 Prevent gear motor from being shocked or dropping in mounting process.It is prohibited to strike and crush directly on reducer output shaft.
3,安装螺丝请前先确认外观图所标示之形式及长度尺寸。 Install the screws after first checking the taps and tap depth against the dimensions entered on the exterior drawing.
4,请小心使用，勿使黏接剂借由输出轴流入轴内。 It is necessary to be careful that the adhesive does not spread along the shaft and flow into the bearing.
5,请不要在安装零件时为了调整位置而从输出轴方向去回转齿轮马达。 Do not turn a geared motor by its output shaft when,for example,arranging its position so as to install it.

Figure 2.3 Specification of 12V 225RPM 1.3kgfcm Brushed DC Geared Motor

The specification from **Figure 2.3** tells us the rated current 570mA in value. The datasheet link provided below:

Datasheet: https://drive.google.com/file/d/1Dmn5dxFfEKLwMLhDiL9wTeDIEwwnsN2O/view

12V 225RPM 1.3kgfcm Brushed DC Geared Motor Power = 570mA x 12V = 6.84W

- Datasheet for AC Cord (RS PRO 3M Power Cable) is shown below:

Electrical Specifications

Current Rating	10A
Voltage Rating	250V
Insulation Resistance	>10MΩ
Cable Retention	0.5mm²-1mm²>30N 1.25mm²-1.5mm²>60N
Electrical Strength	2.5kV 50Hz
Safety Test	4.4kV 50Hz
Capacitance	95.08pF/m
Inductance	0.43μH/m

Figure 2.4 Specifications of AC Cord (RS PRO 3M POWER CABLE)

Electrical specifications **Figure2.4** tells that the current rating of 10A and the voltage rating 250V. For more info of the datasheet click the link below:

Datasheet: https://docs.rs-online.com/789e/A700000007928240.pdf

AC Cord Power = 10A x 250V = 2500W

As stated that the systems of Electric Thermos Pot 3.0 L is using AC power which supply power of 2500W. Hence, all the total power of the system components must not exceed the AC Cord Power of 2500W when running the systems to avoid over loaded and overheat.

- Datasheet for 5mm Round White LED is shown below:

Absolute Maximum Ratings (Ta=25℃)

Parameter	Symbol	Rating	Unit
Continuous Forward Current	I_F	30	mA
Peak Forward Current(Duty /10 @ 1KHZ)	I_{FP}	100	mA
Reverse Voltage	V_R	5	V
Operating Temperature	T_{opr}	-40 ~ +85	℃
Storage Temperature	T_{stg}	-40 ~ +100	℃
Soldering Temperature (T=5 sec)	T_{sol}	260 ± 5	℃
Power Dissipation	P_d	100	mW
Zener Reverse Current	Iz	100	mA
Electrostatic Discharge	ESD	4K	V

Figure 2.5 Specification of 5mm Round White LED

As from **Figure 2.5** state that the peak current is 100mA while the reverse voltage is 5V. For more info of the datasheet click the link below:

Datasheet: http://www1.futureelectronics.com/doc/EVERLIGHT%C2%A0/334-15__T1C1-4WYA.pdf

5mm Round White LED Power = 100mA x 5V = 0.5W

Since using 2 5mm Round White LED Power,

0.5W x 2 = 1.0W

- Hence all the total power of each component except the AC Cord,

Total Power of the Electric Thermos Pot 3.0L = 3.6W + 0.35W + 6.84W + 1.0W = 11.79W

Estimation Heat Produced

To calculate heat produced, as we know,

$$Power = \frac{Heat}{time}$$

Hence,

$$Heat = Power \times time$$

We assume all time = 1s, 1W = $\frac{Joules}{second}$. Hence,

Tactile Switch Heat Produced = 3.6W x 1s = 3.6 J

Temperature sensor LM35 Heat Produced = 0.35W x 1s = 0.35 J

12V 225RPM 1.3kgfcm Brushed DC Geared Motor Heat Produced = 6.84W x 1s = 6.84 J

5mm Round White LED Heat Produced = 1.0W x 1s = 1.0 J

Hence,

Total heat produced = 3.6 J + 0.35 J + 6.84 J + 1.0 J = 11.79 J

System Connectivity's

For the system connectivity's it is recommended to use AC power as the systems use right now. It is possible to use battery but is not very convenient overtime also for the user. For example:

- The total estimation for the 3L water to boil is around 10 minutes,

Total minutes in second = 10 min x $\frac{60s}{1\ minute}$ = 600s

Consider the total power is produce every 1 second. Hence,

Total power of the Electric Thermos Pot 3.0L after 10 minutes =11.79W/s x 600s = 7074W

Consider we taking a very high Ah battery and voltage where 36 V and 4.4Ah battery.

Total power of the battery = 36 V x 4.4A = 158.4W

Hence the battery produces 156.4W for each hour only. In addition, do we need around 45 batteries for a single boil only? Not worthy right? Hence, using the battery is not necessary or really not recommended for this system.

TASK 3

3.1 Circuit of Electric Thermos Pot 3.0L using Proteus

By following the **Figure 1.1** Block Diagram, figure below shows the constructed circuit using Proteus:

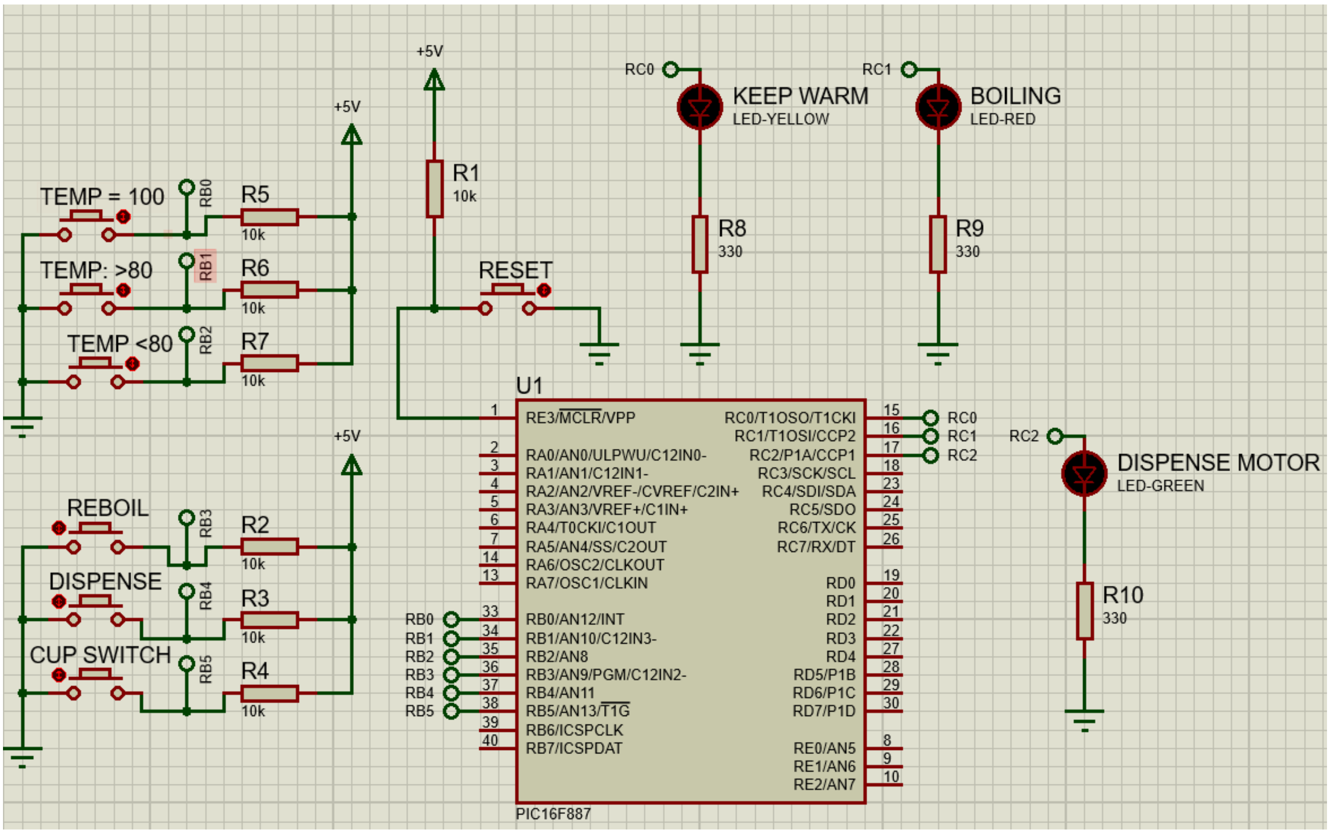

Figure 3.0 Thermos Pot 3.0L Circuit on Proteus

Because some parts cannot be used by doing the hardware for the demo, I changed some parts which are the LM35 to button to control the temperature for certain value & DC Motor to LED just to indicate the circuit is working properly.

3.2 Flowchart and C Programming.

We will derive a flowchart on how the Electric Thermos Pot 3.0L work and based on the flowchart, the C programming will be written for embedded it on PIC16F887.

Flowchart

The flowchart to present the operation of the embedded system is shown below:

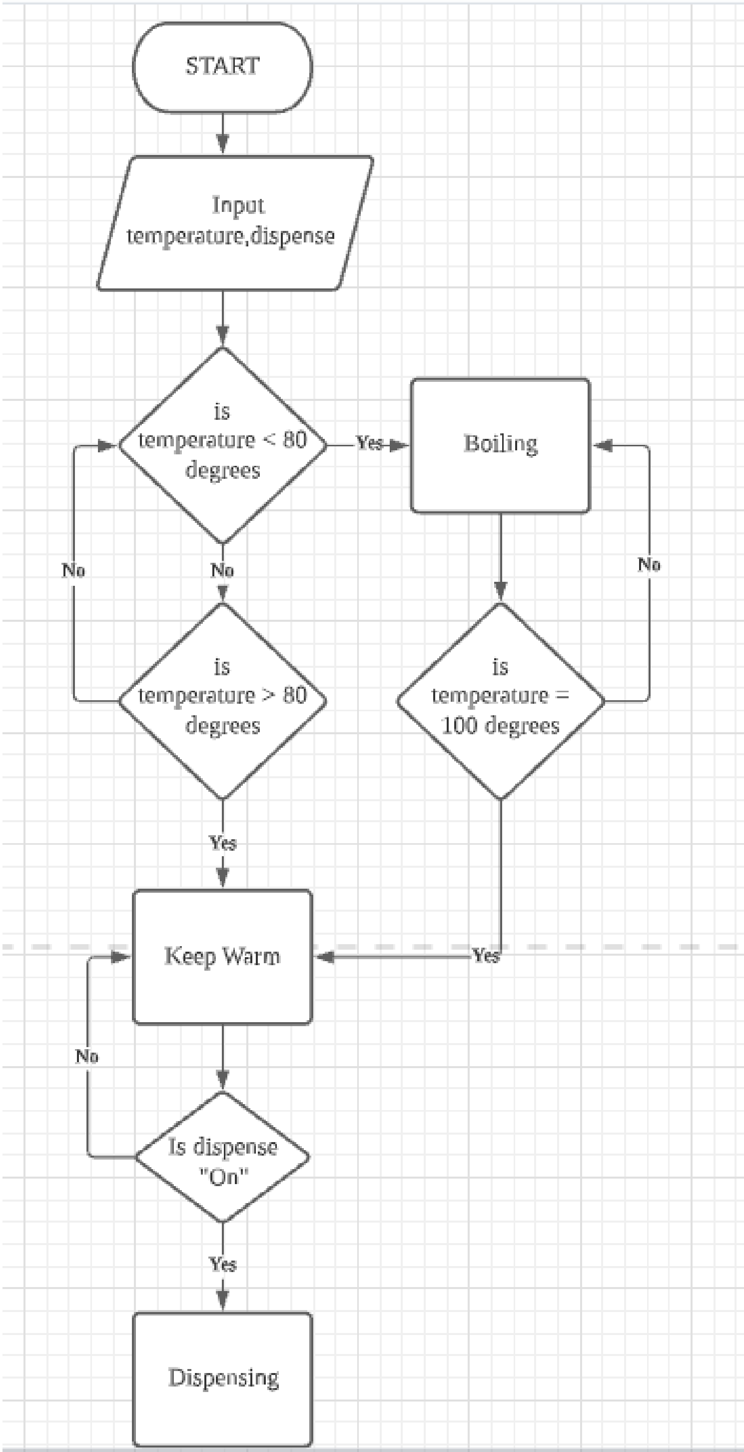

Figure 3.1 Flowchart Thermos Flask Operation Embedded System

C Programming Code

```
// CONFIG1

#pragma config FOSC = INTRC_NOCLKOUT// Oscillator Selection bits (INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled and can be enabled by SWDTEN bit of the WDTCON register)

#pragma config PWRTE = ON // Power-up Timer Enable bit (PWRT enabled)

#pragma config MCLRE = OFF // RE3/MCLR pin function select bit (RE3/MCLR pin function is digital input, MCLR internally tied to VDD)

#pragma config CP = OFF // Code Protection bit (Program memory code protection is disabled)

#pragma config CPD = OFF // Data Code Protection bit (Data memory code protection is disabled)

#pragma config BOREN = ON // Brown Out Reset Selection bits (BOR enabled)

#pragma config IESO = ON // Internal External Switchover bit (Internal/External Switchover mode is enabled)

#pragma config FCMEN = ON // Fail-Safe Clock Monitor Enabled bit (Fail-Safe Clock Monitor is enabled)

#pragma config LVP = OFF // Low Voltage Programming Enable bit (RB3 pin has digital I/O, HV on MCLR must be used for programming)

// CONFIG2

#pragma config BOR4V = BOR40V // Brown-out Reset Selection bit (Brown-out Reset set to 4.0V)

#pragma config WRT = OFF // Flash Program Memory Self Write Enable bits (Write protection off)

// #pragma config statements should precede project file includes.

// Use project enums instead of #define for ON and OFF.


#include <xc.h>

#include <stdlib.h>

//#include <stdio.h> //header file for sprintf


#define _XTAL_FREQ 8000000

#define ON 1

#define OFF 0


void DelaySeconds();
```

```
void main()
{
   TRISA=0X07;
   TRISB=0X3F;
   TRISC=0XF8;

   OSCCON=0X75;
   ANSELH=0X00;

   PORTBbits.RB0=1;
   PORTBbits.RB1=1;
   PORTBbits.RB2=1;

   PORTBbits.RB3=1;
   PORTBbits.RB4=1;
   PORTBbits.RB5=1;

   PORTEbits.RE3=1;

   PORTCbits.RC0=OFF;
   PORTCbits.RC1=OFF;
   PORTCbits.RC2=OFF;

while(1)
   {
      //Temp button
      if (PORTBbits.RB0==0)
      {
         PORTCbits.RC0=ON;
         PORTCbits.RC1=OFF;
         PORTCbits.RC2=OFF;
      }
```

```
if (PORTBbits.RB1==0)
      {
         PORTCbits.RC0=ON;
         PORTCbits.RC1=OFF;
         PORTCbits.RC2=OFF;
      }
      if (PORTBbits.RB2==0)
      {
         PORTCbits.RC0=OFF;
         PORTCbits.RC1=ON;
         PORTCbits.RC2=OFF;
      }

      //Main button
      if(PORTBbits.RB3==0)
      {
         PORTCbits.RC0=OFF;
         PORTCbits.RC1=ON;
         PORTCbits.RC2=OFF;
      }
      if(PORTBbits.RB4==0)
      {
         PORTCbits.RC0=ON;
         PORTCbits.RC1=OFF;
         PORTCbits.RC2=ON;
      }
      if (PORTBbits.RB5==0)
      {
         PORTCbits.RC0=ON;
         PORTCbits.RC1=OFF;
         PORTCbits.RC2=ON;
      }
```

```
if (PORTEbits.RE3==0)
      {
         PORTCbits.RC0=OFF;
         PORTCbits.RC1=OFF;
         PORTCbits.RC2=OFF;
      }
   }
}
```

3.3 Simulation

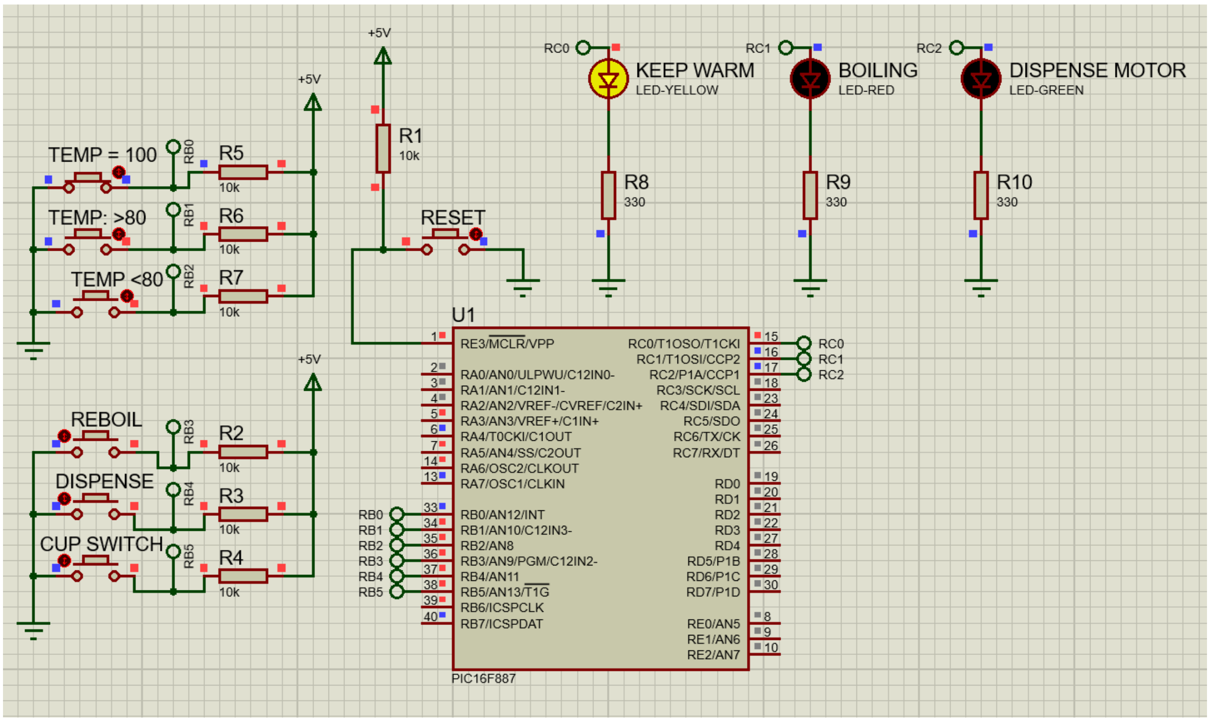

Figure 3.3 If Temperature Reach 100 °C

From **Figure 3.3** if we pressed the "TEMP=100" meaning the temperature reaches 100°C the LED "KEEP WARM" will be ON which means the thermos flask is ready to use.

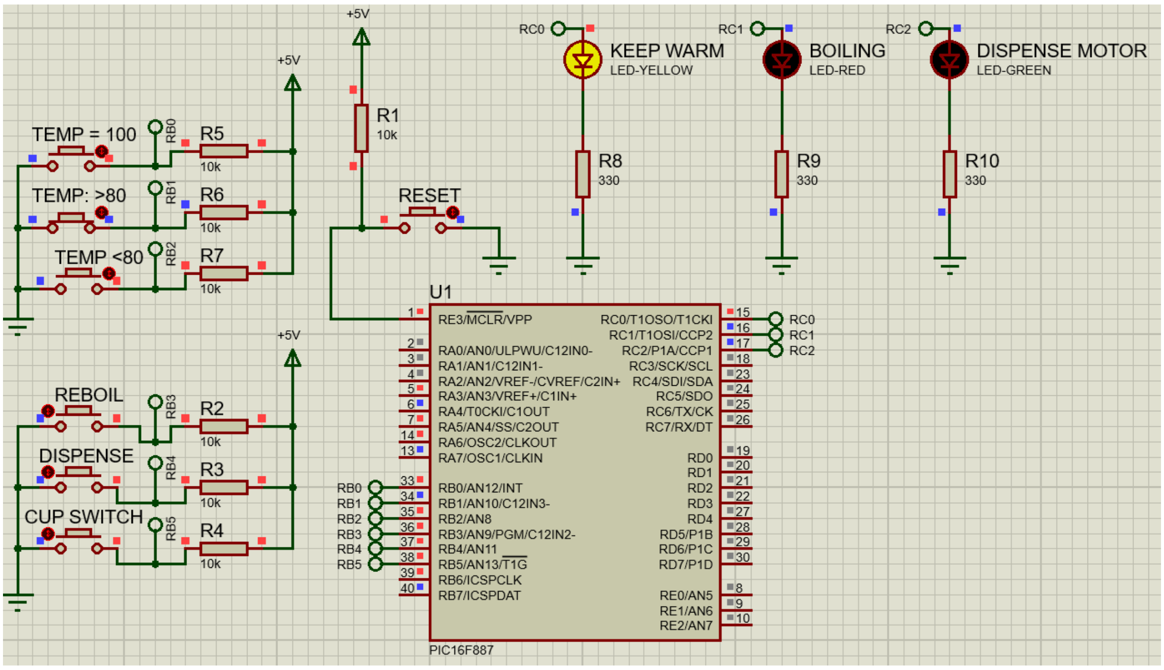

Figure 3.4 If Temperature Reach 80 °C after reach 100 °C

From **Figure 3.4**, if we pressed the "TEMP>80" button meaning the temperature is greater than 80°C (temperature reference to keep water warm) after boiling, the LED "KEEP WARM" will ON which means the thermos flask is ready to use.

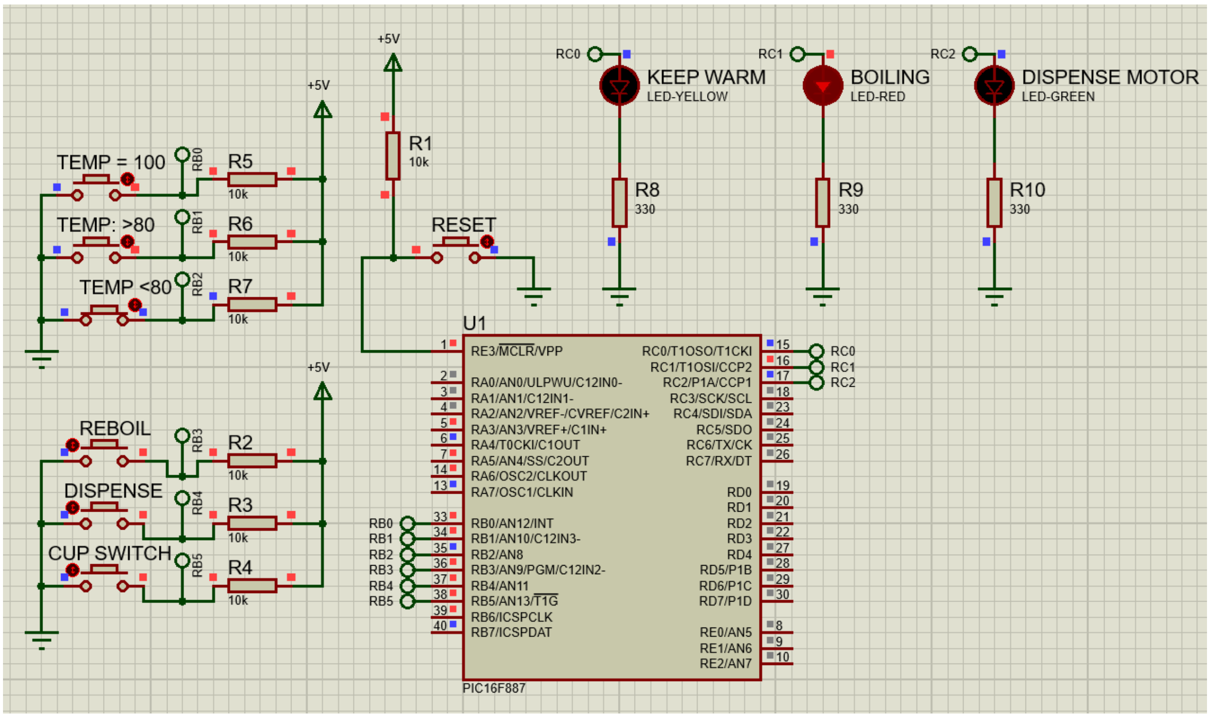

Figure 3.5 If Temperature below than 80°C

From **Figure 3.5,** if we pressed the "TEMP<80" button meaning the temperature is below the temperature reference and needs to be boiled. The LED "BOILING" will be ON which means the water in the thermos flask is in the boiling process and not ready to be used.

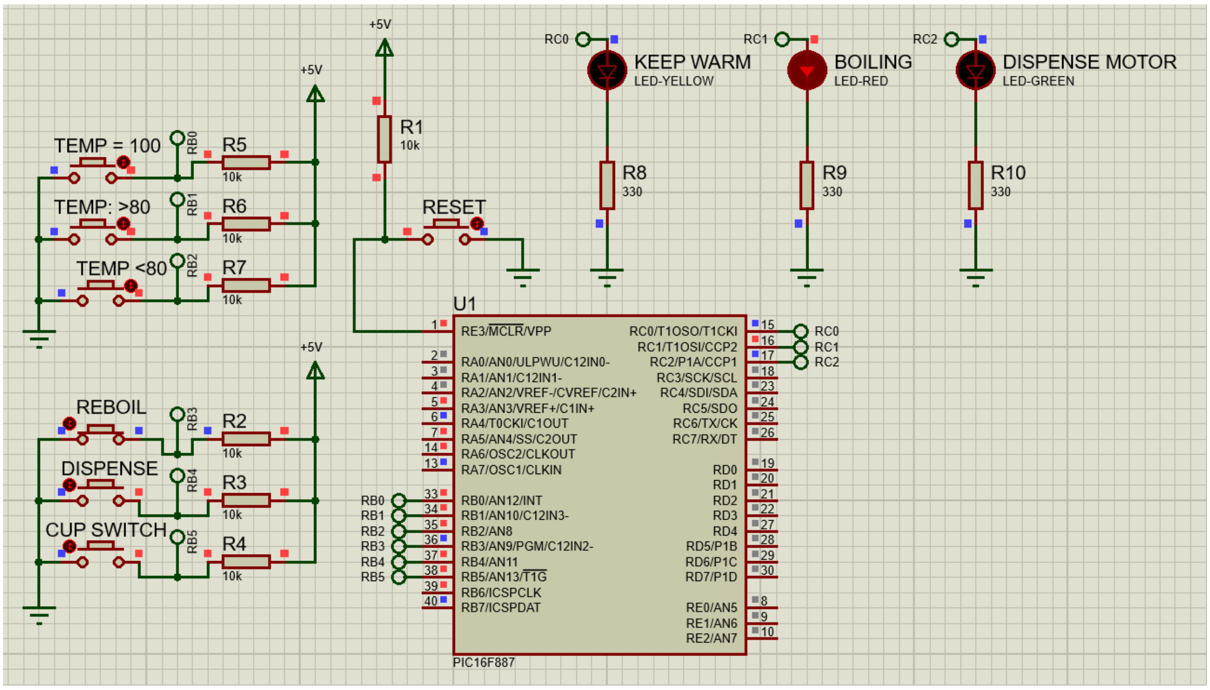

Figure 3.6 If Re-Boil Button Pressed

From **Figure 3.6**, if we pressed the "REBOIL" button meaning the water in the thermos flask will be boiling back ignoring the temperature references. The LED "BOILING" will be ON which means the water in the thermos flask is in boiling the process and not ready to be used.

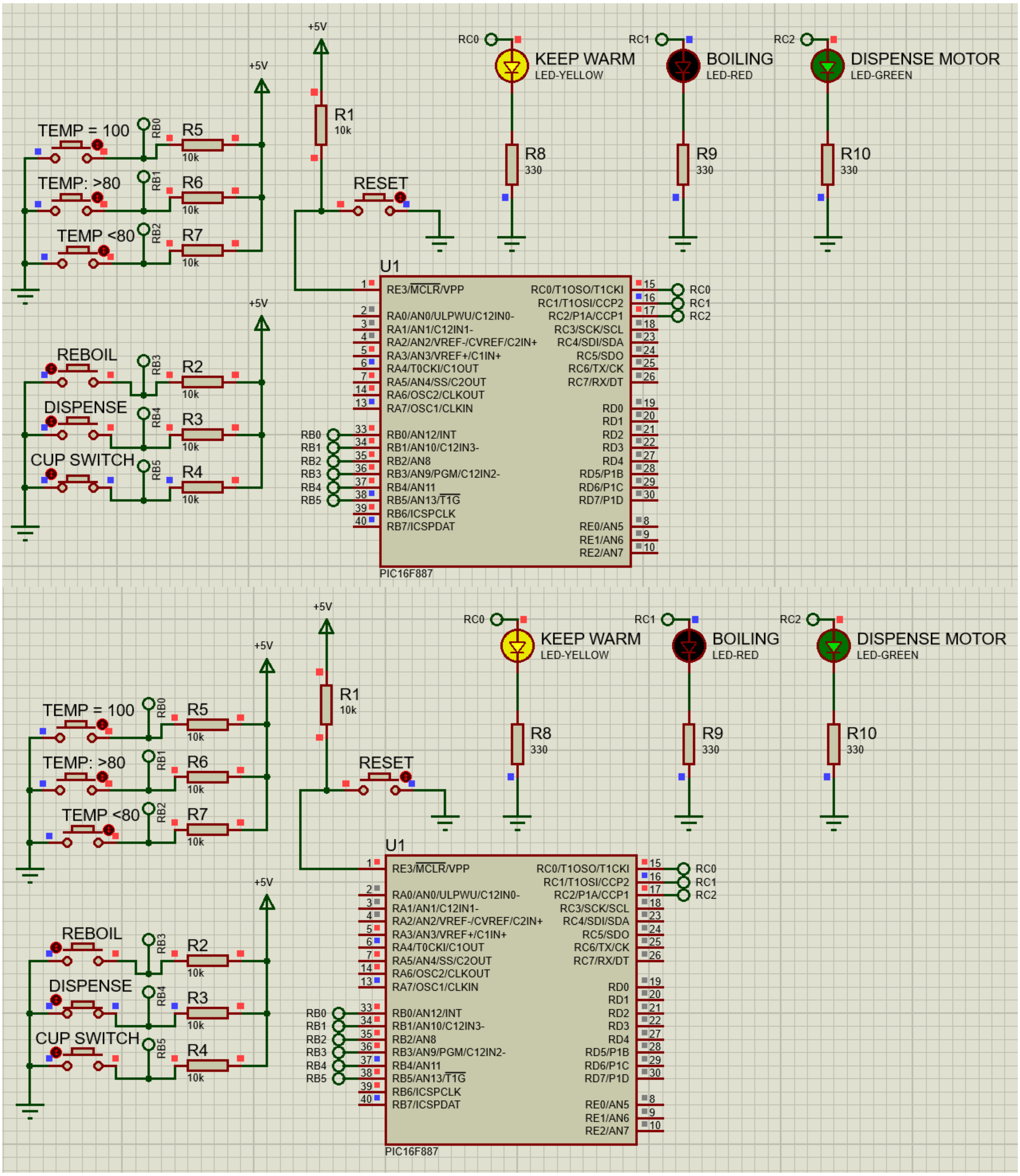

Figure 3.7 If Pressed Dispense or Cup Switch Button

The pressed button and cup switch have the same working mechanism which lets the water out from the thermos flask from the water outlet helped by the motor inside. If the "DISPENSE" or "CUP SWITCH" button is pressed, LED "KEEP WARM" will on meaning the thermos flask is ready to use and the LED "DISPENSE MOTOR" will also ON meaning the water is being pushed out from the thermos flask.

The simulation of the circuit **Task 3.1** can be view on the link provided below:

Link:
https://drive.google.com/file/d/1o78lrUVy1kqyJ5eREOyRuZeUoCsRV7M3/view?usp=sharing

3.4 Improvement of the Embedded System

The improvement that can be considered to the current Embedded System is additional of the LCD screen.

i. The reason why the system needs improvement is that the user sometimes cannot indicate what they are doing without double-checking on it. If there's an LCD screen, they can easily what they are doing correctly and can check additional features on the screen like showing how many minutes more for the water boil completely.
ii. The type of LCD screen that will be use is 16 X 2 LCD Screen. Below is the figure of the improvement circuit.

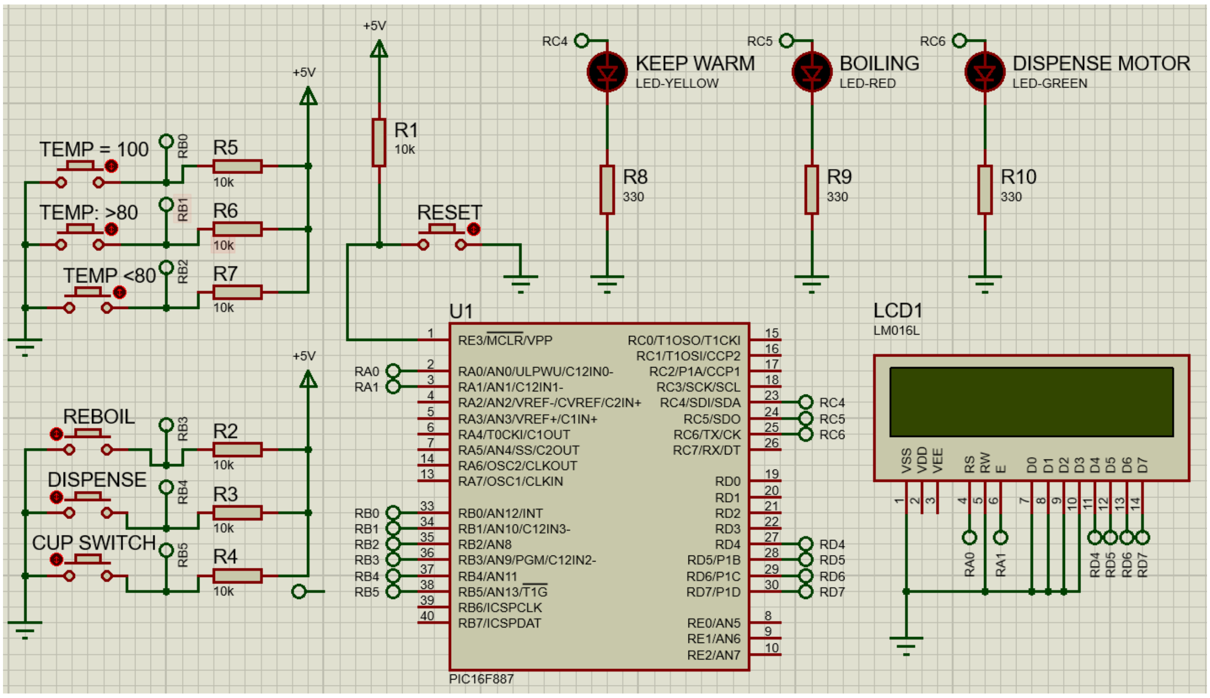

Figure 3.1 Modified Improvement Circuit

The C programming code for the circuit are shown below:

```
// CONFIG1

#pragma config FOSC = INTRC_NOCLKOUT// Oscillator Selection bits (INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled and can be enabled by SWDTEN bit of the WDTCON register)

#pragma config PWRTE = ON // Power-up Timer Enable bit (PWRT enabled)

#pragma config MCLRE = OFF // RE3/MCLR pin function select bit (RE3/MCLR pin function is digital input, MCLR internally tied to VDD)

#pragma config CP = OFF // Code Protection bit (Program memory code protection is disabled)

#pragma config CPD = OFF // Data Code Protection bit (Data memory code protection is disabled)

#pragma config BOREN = ON // Brown Out Reset Selection bits (BOR enabled)

#pragma config IESO = ON // Internal External Switchover bit (Internal/External Switchover mode is enabled)

#pragma config FCMEN = ON // Fail-Safe Clock Monitor Enabled bit (Fail-Safe Clock Monitor is enabled)

#pragma config LVP = OFF // Low Voltage Programming Enable bit (RB3 pin has digital I/O, HV on MCLR must be used for programming)

// CONFIG2

#pragma config BOR4V = BOR40V // Brown-out Reset Selection bit (Brown-out Reset set to 4.0V)

#pragma config WRT = OFF // Flash Program Memory Self Write Enable bits (Write protection off)
```

```
// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#include <xc.h>
#include <stdlib.h>
#include <stdio.h> //header file for sprintf

#define _XTAL_FREQ 8000000
#define ON 1
#define OFF 0
#define RS PORTAbits.RA0
#define EN PORTAbits.RA1
#define D4 PORTDbits.RD4
#define D5 PORTDbits.RD5
#define D6 PORTDbits.RD6
#define D7 PORTDbits.RD7

#include <xc.h>
#include <stdlib.h>
#include <stdio.h> //header file for sprintf
#include "lcd.h"

int i;

void main()
{
    TRISA=0X00;
    TRISB=0X3F;
    TRISC=0X00;
    TRISD=0X00;
```

```
   OSCCON=0X75;
   ANSELH=0X00;
   ANSELH=0X00;

   PORTBbits.RB0=1;
   PORTBbits.RB1=1;
   PORTBbits.RB2=1;

   PORTBbits.RB3=1;
   PORTBbits.RB4=1;
   PORTBbits.RB5=1;
   PORTBbits.RB6=1;

   PORTEbits.RE3=1;

   PORTCbits.RC4=OFF;
   PORTCbits.RC5=OFF;
   PORTCbits.RC6=OFF;

    Lcd_Init();
      __delay_ms(500);
   Lcd_Clear();

while(1)
   {
      __delay_ms(200);
      Lcd_Clear();
      Lcd_Set_Cursor(1,1);
      Lcd_Write_String("WELCOME..");

      if (PORTBbits.RB0==0)
      {
         Lcd_Clear();
         Lcd_Set_Cursor(1,1);
```

```
Lcd_Write_String("WELCOME..");

    if (PORTBbits.RB0==0)
    {
       Lcd_Clear();
       Lcd_Set_Cursor(1,1);
       Lcd_Write_String("KEEP WARM...");
       PORTCbits.RC4=ON;
       PORTCbits.RC5=OFF;
       PORTCbits.RC6=OFF;
       __delay_ms(800);
    }

    if (PORTBbits.RB1==0)
    {
       Lcd_Clear();
       Lcd_Set_Cursor(1,1);
       Lcd_Write_String("KEEP WARM...");
       PORTCbits.RC4=ON;
       PORTCbits.RC5=OFF;
       PORTCbits.RC6=OFF;
       __delay_ms(800);
    }

    if (PORTBbits.RB2==0)
    {
       Lcd_Clear();
       Lcd_Set_Cursor(1,1);
       Lcd_Write_String("BOILING...");
       PORTCbits.RC4=OFF;
       PORTCbits.RC5=ON;
       PORTCbits.RC6=OFF;
       __delay_ms(800);
    }
```

```
if(PORTBbits.RB3==0)
{
  Lcd_Clear();
  Lcd_Set_Cursor(1,1);
  Lcd_Write_String("BOILING..");
  PORTCbits.RC4=OFF;
  PORTCbits.RC5=ON;
  PORTCbits.RC6=OFF;
  __delay_ms(800);
}

if(PORTBbits.RB4==0)
{
  Lcd_Clear();
  Lcd_Set_Cursor(1,1);
  Lcd_Write_String("DISPENSING..");
  PORTCbits.RC4=ON;
  PORTCbits.RC5=OFF;
  PORTCbits.RC6=ON;
  __delay_ms(800);
}

if (PORTBbits.RB5==0)
{
  Lcd_Clear();
  Lcd_Set_Cursor(1,1);
  Lcd_Write_String("DISPENSING..");
  PORTCbits.RC4=ON;
  PORTCbits.RC5=OFF;
  PORTCbits.RC6=ON;
  __delay_ms(800);
}
```

```
if (PORTEbits.RE3==0)
      {
         Lcd_Clear();
         __delay_ms(1000);
         PORTCbits.RC4=OFF;
         PORTCbits.RC5=OFF;
         PORTCbits.RC6=OFF;
      }
   }
}
```

- The power requirement of the circuit will be changed considering the improvement also some small changes of the modified circuit because of lack of electronic part. This power requirement will be calculated and divided into 2 parts which are from continuing from **TASK 2.2 Estimation Total Power** and from **Figure 3.1 Modified Improvement Circuit.**

<u>Total Power Based on Real World Circuit</u>

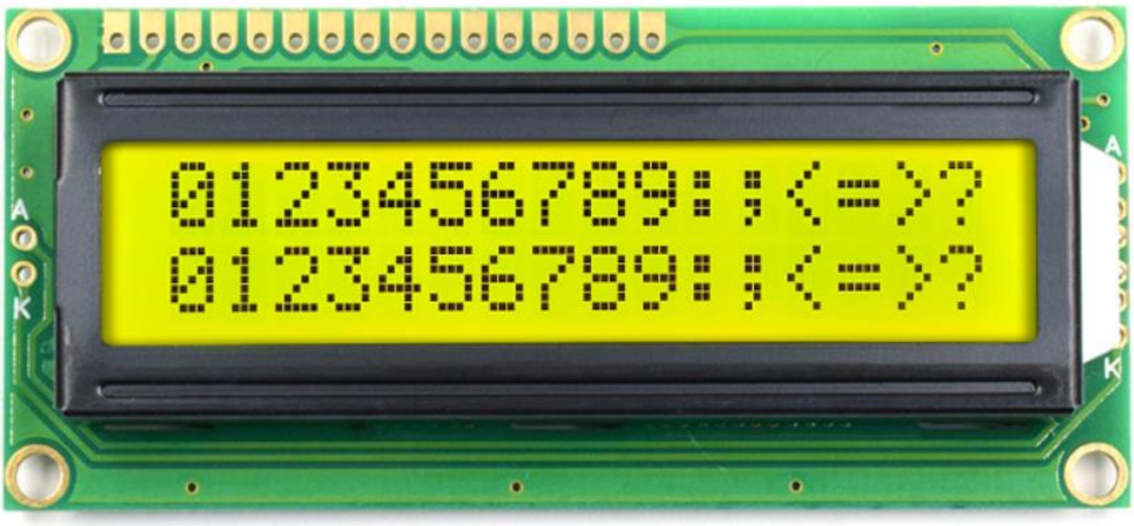

Figure 3.2 16 x 2 LCD Display

The datasheet of the 16 x 2 LCD Display is shown below:

② FEATURES/特征

Feature(特征)		UNIT(单位)
LCD type & Colour(类型及颜色)	STN Yellow-green ;STN Blue	
View Angle(视角)	6 0' clock	
Operating temperature(工作温度)	-0℃ ---- 50℃	
Storage Temperature(贮存温度)	-10℃ ---- 60℃	
Control IC & Package(控制芯片及封装)	SPLC780D , COB	
Weight(重量)	32 (With B/L)	g
Backlight(背光类型)	LED	

Ta=25℃		MIN(最小)	TYP(典型)		MAX(最大)	UNIT(单位)
Supply current don't contain backlight(无背光时电流)		0		4		mA
VDD(电源)		-0. 3	3. 3	5. 0		V
VLCD-V0(LCD驱动电压)		0	5. 0	5. 0		V
Backlight LED (背光电压,电流)	侧背光	0	3. 3	5. 0		V
		0	20	30		mA
	底背光			5. 0		V
				80		mA

Figure 3.3 Specifications of 16 x 2 LCD Display

As stated on **Figure 3.3** the peak current is 5V while the current is 80mA and 30mA for backlight LED and 4mA for supply current don't contain backlight. The datasheet of 16x2 LCD Display can be obtaining from the link below:

Datasheet:
http://www.tgklcd.com/index.php?m=content&c=index&a=show&catid=44&id=19

16 x 2 LCD Display Power = 5V x (80Ma + 30mA + 4mA) = 0.57W

Hence,

Total Power Application = Total Power Task 2.2 + 16 x 2 LCD Display Power

= 11.79W + 0.57W = 12.36W

Total Power Based on Modified Circuit

Based on **Task 2.2**, Tactile Switch power is 1.2W each. Hence from **Figure 3.1**,

Tactile Switch Power = 6 x 1.2W = 7.2W

While the LED light is 0.5W each.

LED Light Power = 3 x 0.5W = 1.5W

Hence the total power of the modified circuit is:

Total Power = 7.2W + 1.5W + 0.57W = 9.27W

TASK 4

4.1 Constructed Real Hardware of the System

Figure below is the constructed real hardware circuit of the system on the breadboard:

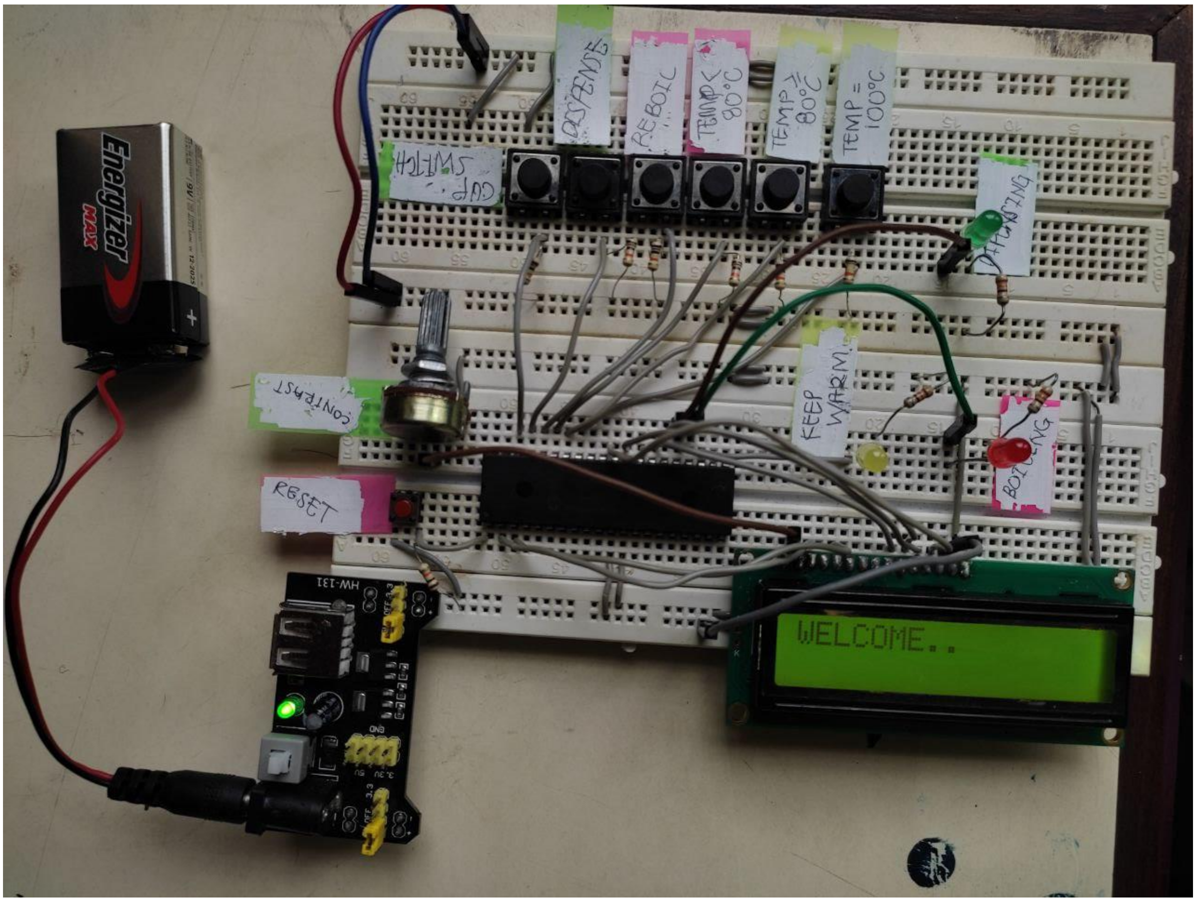

Figure 4.1 Real Hardware of the Thermosflask System

The simulation of the real hardware Thermosflask system can be view on the link below:

https://drive.google.com/file/d/1agMngkY93Ybxrm-8yObpWzCPImtplHg_/view?usp=sharing

Conclusion

The design of the embedded system on the Thermosflask is influenced by several circumstances trends such as the complexity of the circuit which in a real thermos flask need a real temperature sensor, water, and motor also design that does not include in the mini-project above. As mentioned that the purpose of this mini-project is just needed to know the basic step of creating the system. The process includes building the block diagram, calculating the power requirement, building the source code, testing the simulation, building the system, embedding the system, and testing the real hardware constructed of the system. Throughout the process, the designer has learned a lot of things that need to be considered especially when troubleshooting the real hardware circuit of the system which takes more time for me as the designer to make the system work properly. The troubleshooting part can take you back to the beginning where need to adjust the source code and test it back till the end. No matter how complex and a lot of consideration need to look at the system, the joy built a successful system achieved will bring the joy of our creation at the last moment.