Embedded Systems Project

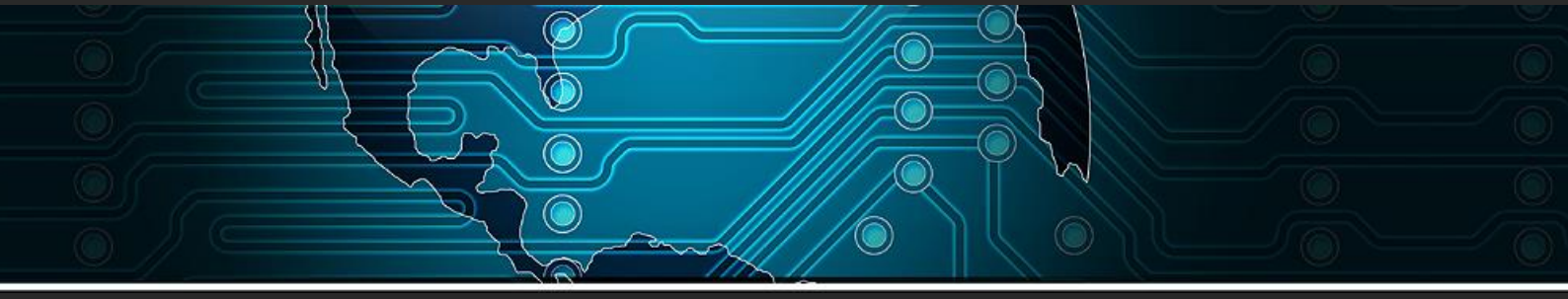
Mohammad AbuJrai – 0172241

Omar Mesheh – 0174034

Yousef Al-Hersh – 0175551

Dr. Ramzi Saifan

Summer 2020



Project Overview

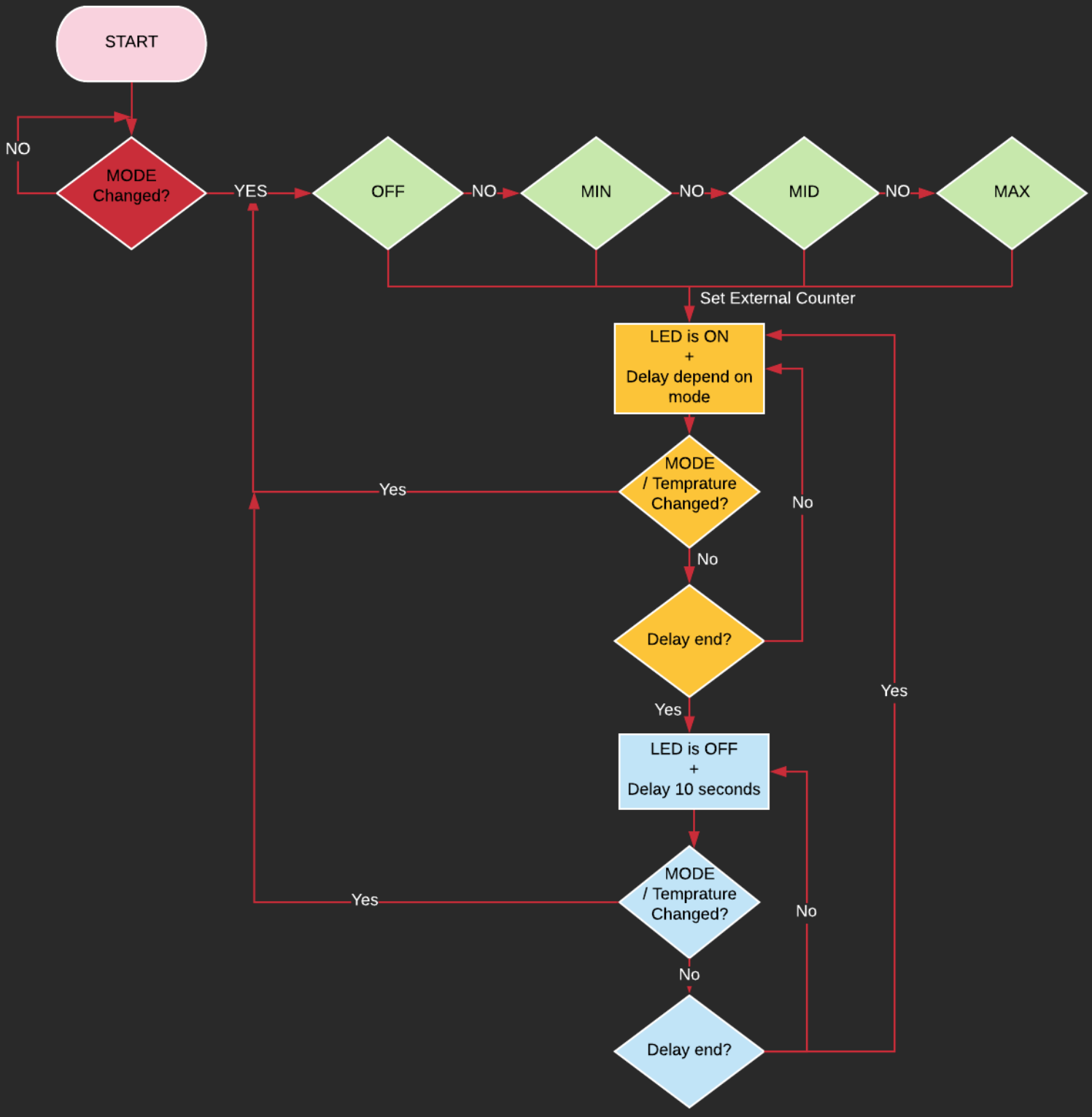
This Project is a digital simulation of a fridge cooling system. It consists of two parts: PIC1 & PIC2.

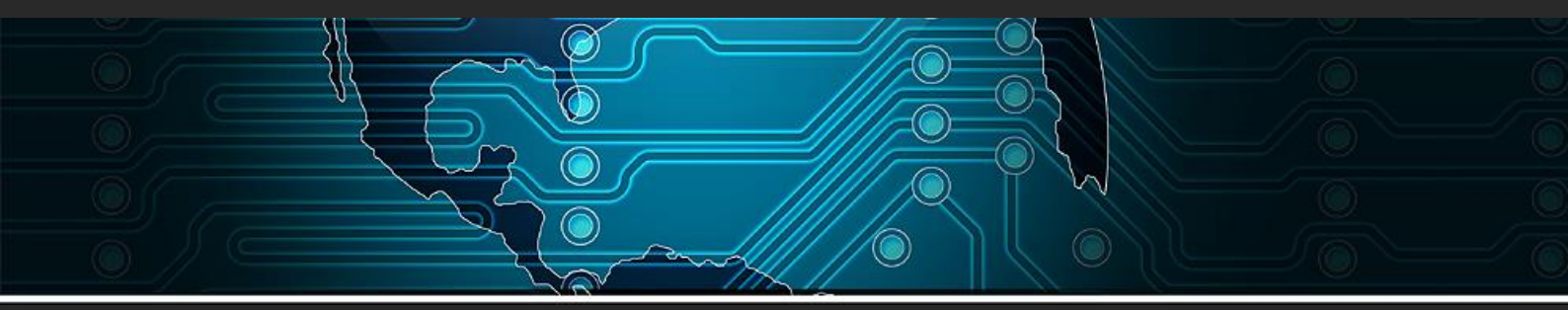
PIC1 act as the temperature sensor, it reads the temperature continuously, save it and then pass it to PIC2.

PIC2 is the main part of this project; it receives the temperature from PIC1, and then process the temperature depending on the mode of operation to decide if the temperature is too high so we start flashing a LED, or if the temperature is too low so we turn the LED Off.

In this project, PIC1 & PIC2 are PIC16F877a Series

Program Flowchart

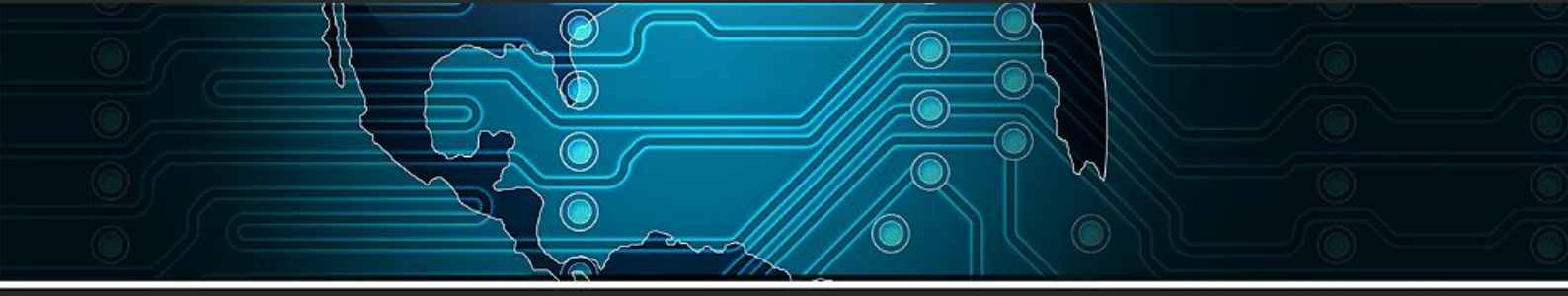




PIC 1

PIC1 is used to read the temperature that is set by the user.
It has five pins act as inputs. In addition to five pins act as outputs.
The input pins are (PORT B [0:4]), PIC1 reads the value at each pin, and then send them by the output pins (PORT C [0:5]) to PIC2.

```
movlw b'00011111'
movwf TRISB
movlw b'00000000'
movwf TRISC
```



PIC 2

PIC 2 is connected to the compressor system, so it tells the compressor (if it is ON) when to cool and when to turn Off.

PIC 2 has seven input pins, they are (PORT C [0:4]) which represent the temperature sent by PIC1.
In addition, PORT B [6:7] act as inputs to represent the mode of operation of this system.

To tell the compressor that the temperature is high and it must turn cooling system on, PIC2 has one single output, which is indicated by RC7

```
movlw b'00011111'
movwf TRISC
movlw b'11000000'
movwf TRISB
```

In this project, PIC 2 has one enabled interrupt, which is PORT B change interrupt.

```
movf  PORTB,0
bcf  INTCON,RBIF
bsf  INTCON,GIE
bsf  INTCON,RBIE
```

Now, the program will enter an infinite loop until the user change the mode of operation.

```
LOOP  btfss FLASH,0
goto LOOP
```

Therefore, since the mode of operation is connected to rb6-rb7 pins, once the mode of operation is changed, the ISR will wake up.
The ISR will convert the first bit of ‘FLASH’ register to indicate that the LED can start flashing now.

```
ISR
    movlw 0x01
    xorwf FLASH,1
    movf PORTB,0
    bcf INTCON,RBIF
    RETFIE
```

When the mode of operation is changed, the program will decide the correct mode by subtracting a copy of that mode with a previously defined register. For example: to check MIN mode, put b'01000000' in register MIN, subtract the value in PORT B and check the zero flag.

```
movf MIN,0
subwf PORTB,0
btfsc STATUS,Z
CALL CASE_MIN
```

Each mode has its own "Temp1" and "Delay Counter" as the following table:

Mode of operation		Counter
OFF	OFF all the time	-----
MIN	Temperature >= 20 → LED is On 10 seconds , Off 10 seconds.	20
	Temperature < 20 → LED OFF.	
MID	Temperature >= 15 → LED is On 15 seconds , Off 10 seconds.	30
	Temperature < 15 → LED OFF.	
MAX	Temperature >= 5 → LED is On 20 seconds , Off 10 seconds.	40
	Temperature < 5 → LED OFF.	

NOTE :

Counter is used in a loop that gives delay of 0.5 seconds.

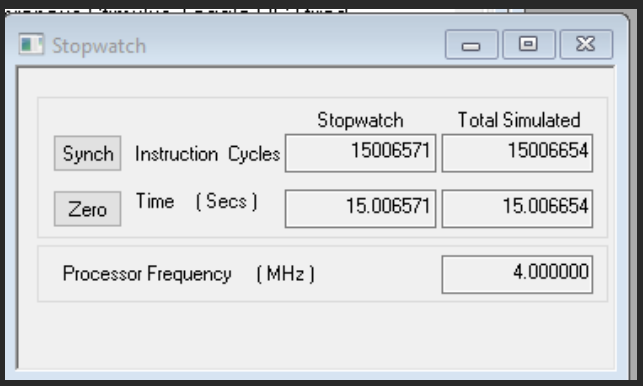
So MIN will give delay of 10 seconds, MID will give 15 seconds, and MAX will give 20 seconds.

After each case delay, the counter will be automatically changed to 20, to give delay of 10 seconds. Calculations are next >>

Delay Calculation

The following Code is used to generate a 0.5 seconds Delay.
It is then inserted in another loop with a counter depends on the mode of operation as explained in the previous table.

```
     movlw D'0'
     movwf EXTERNAL_COUN
     movlw D'244'
     movwf INTERNAL_COUN
LOOP_NOP
    NOP
     NOP
     NOP
     NOP
     NOP
     decfsz EXTERNAL_COUN,1
     goto LOOP_NOP
     decfsz INTERNAL_COUN
     goto LOOP_NOP
     RETURN
```



ex: MID mode delay gives 15sec

Calculations:

$\#T_{inst}$ =

(1 + 1 + 1 + 1) = 4 ➔ initialization

(5 + 1 + 2) × 255 × 243 =495720→All external iterations except last

Nops, decfsz, goto, External Counter, Internal Counter

+ (5 + 2 +1+ 2) × 243 = 2430 ➔ Last external iteration

Nops, decfsz, decfsz, goto, Internal Counter

+ (5 + 2 + 2 + 2) = 11 ➔ Last internal iteration

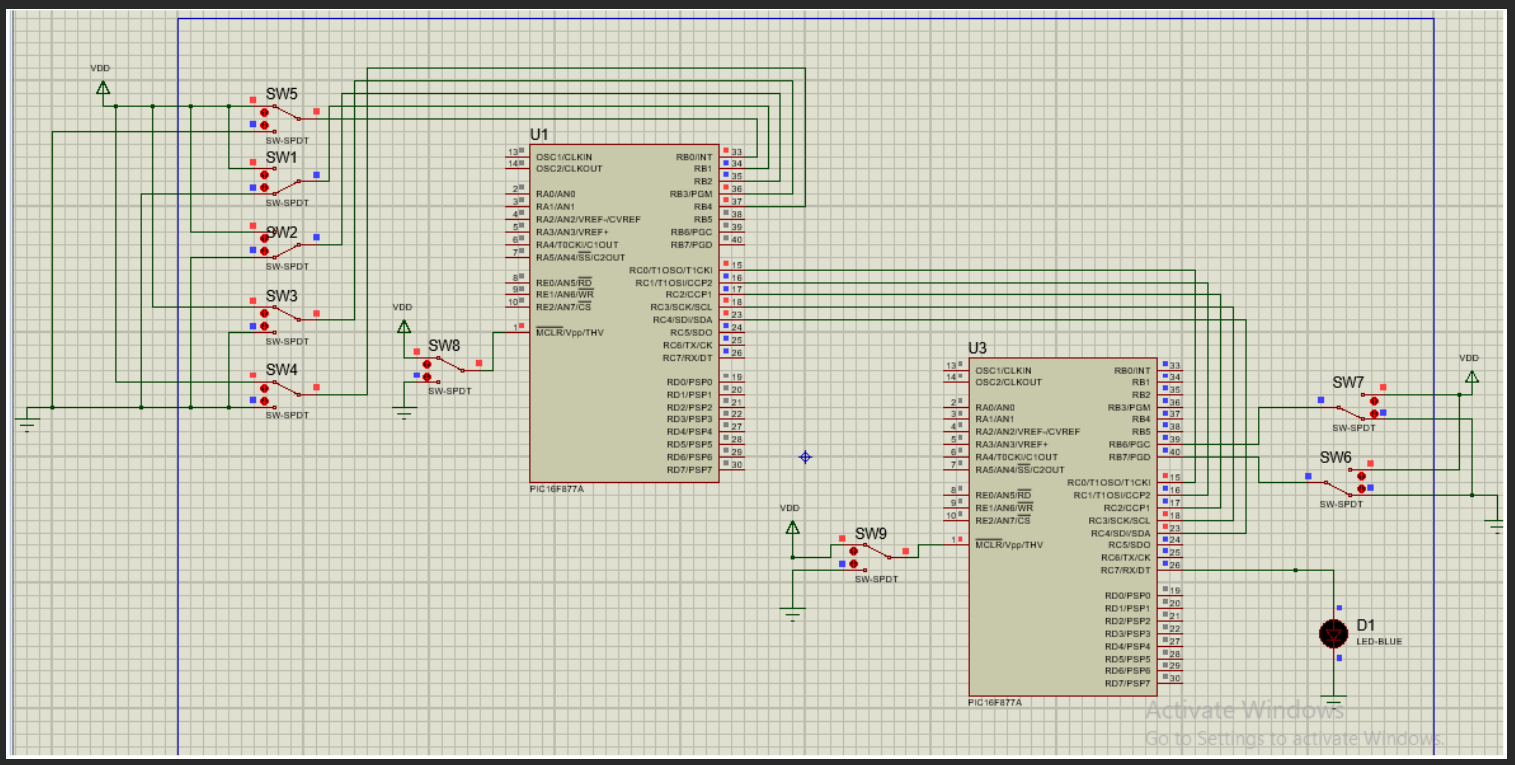
Nops, decfsz, decfsz, return

T_{inst} = 4 + 495720 + 2430 + 11 = 498165

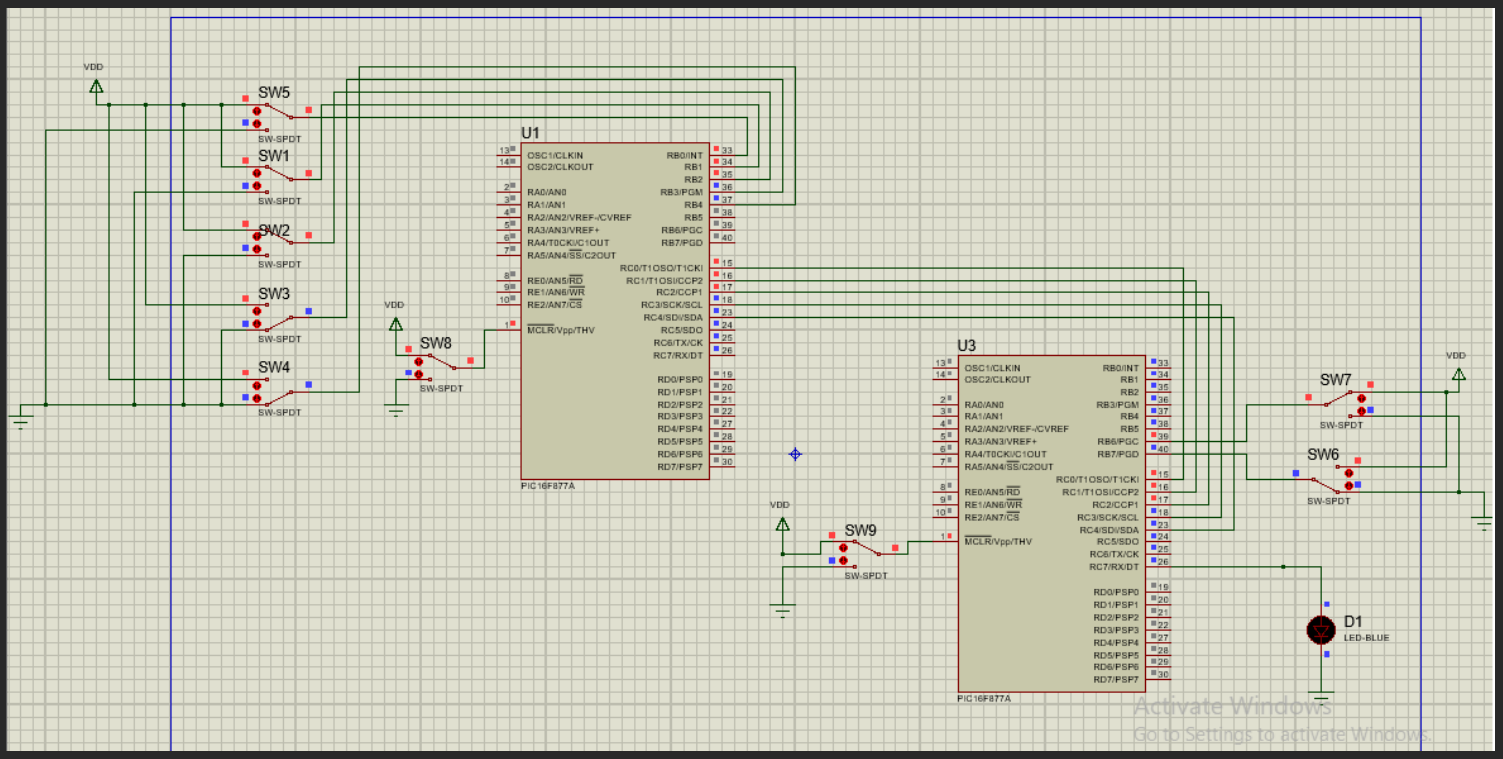
$$\text{Delay} = \frac{4}{4 * 10^6} * 498165 = 0.498 \text{ sec}$$

Final Result and Screenshots from proteus:

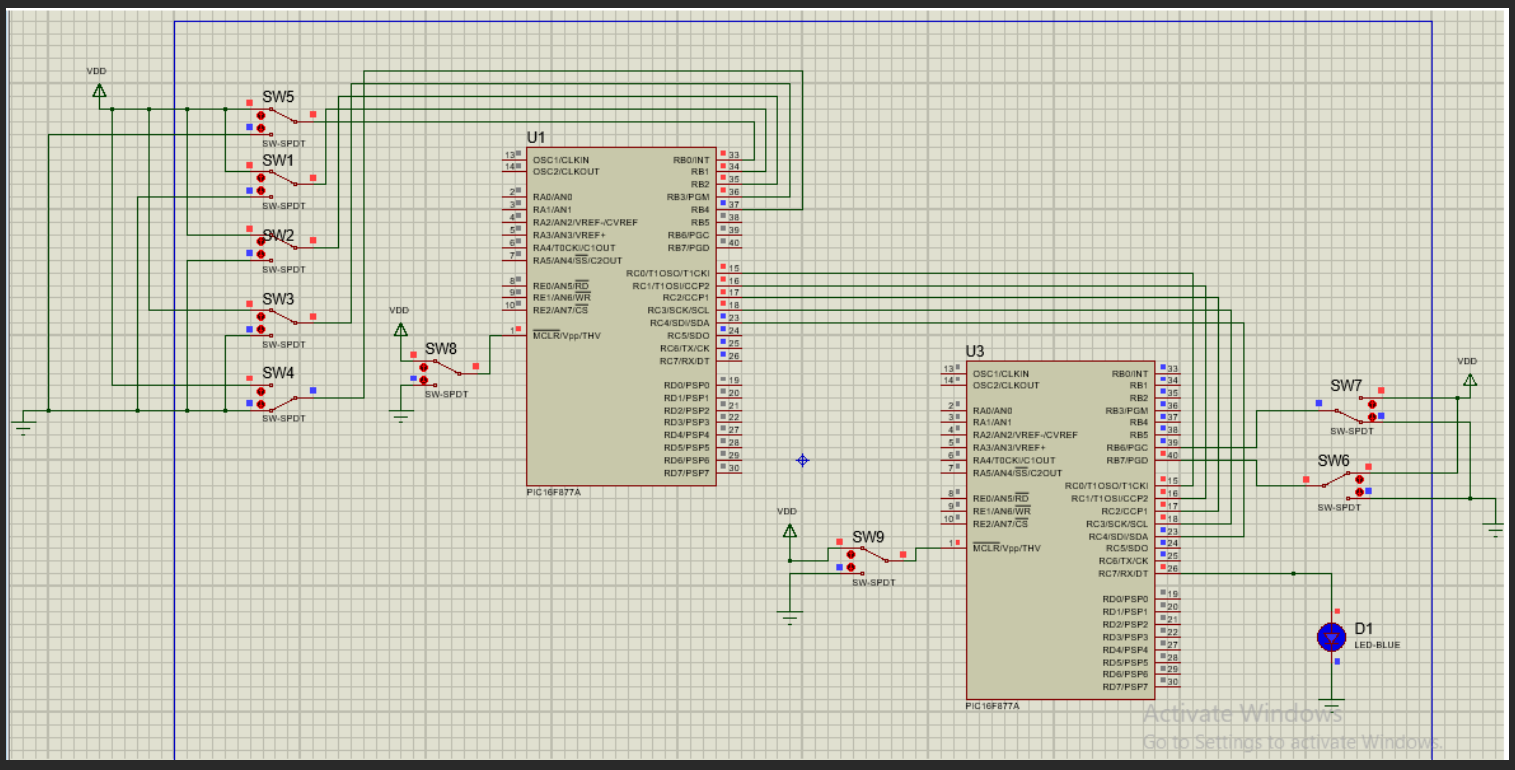
- OFF Mode with temperature 25:



- MIN Mode with temperature 7:



- MID Mode with temperature 15:



- MAX Mode with temperature 6: