DESIGN PROJECT

COMPLEX ENGINEERING PROBLEM REPORT

SUBMITTED BY	RIMSHA ISHTAQ (CS-20039) SYEDA ZOBIA IRSHAD (CS-20043) HABIBA ASIF(CS-20045) VANDANA(CS-20075)
SUBJECT	MICROPROCESSOR-BASED SYSTEM DESIGN COURSE CODE: CS-301
SUBMITTED TO	MS. RAMISH FATIMA

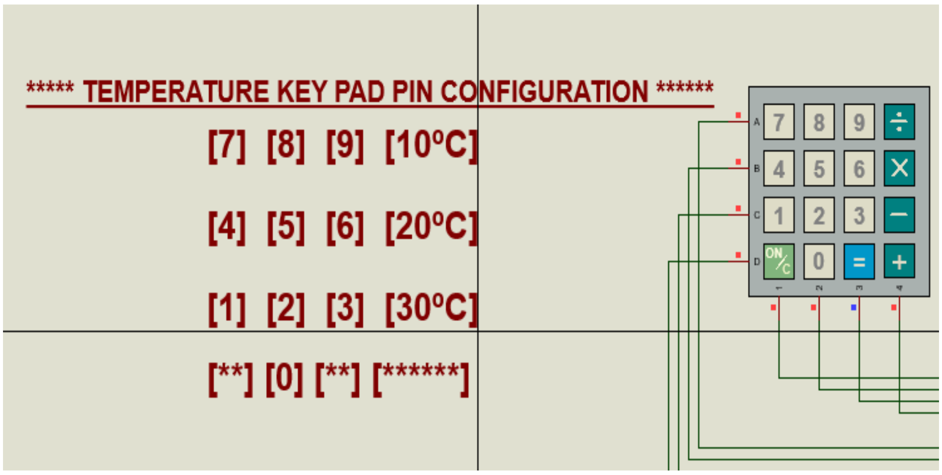
Overall Design Philosophy:

This microprocessor-based application is implemented to control the speed of a fan using PWM technique based on the input temperature through user-specified temperature using Keypad.

This design uses:

- Keypad
- Seven-segment display
- Dc motor
- Motor driver
- Push button

The speed of fan must be in three ranges: fast, medium and slow on some specified temperature so this design is implemented as such that between the range of for temperature input the keypad has been modified as such to take 3 ranges of inputs from the keypad (10-19 low) (20-29 med) (30-39 high/fast) to make the temperature input easier for user. The user must first enter the temperature' tens button i.e., 10°C button and then enter the unit's digit for the range according to given keypad configuration.



***** TEMPERATURE KEY PAD PIN CONFIGURATION ******

[7] [8] [9] [10ºC]

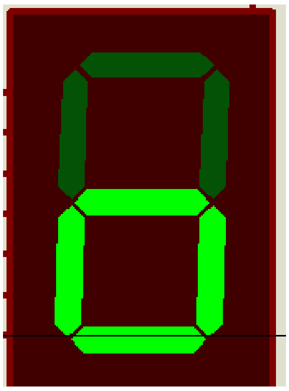
[4] [5] [6] [20ºC]

[1] [2] [3] [30ºC]

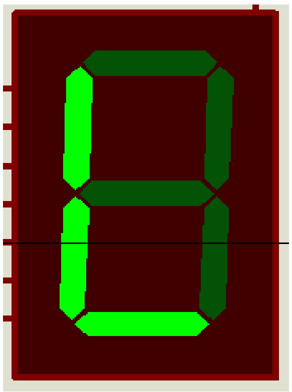
[**] [0] [**] [******]

And the display on the seven-segment will be:

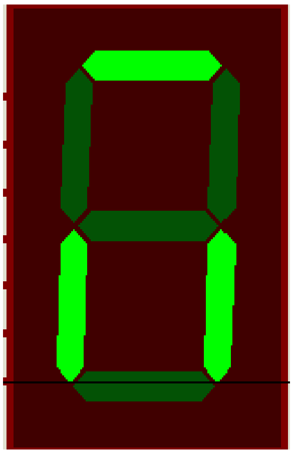
- When motor is off:



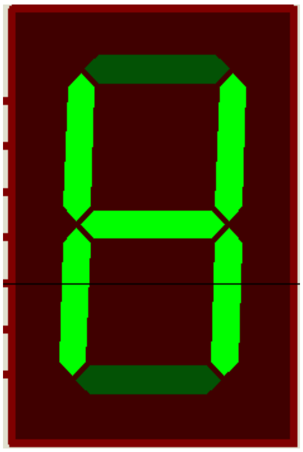
When motor’s speed is low



- When motor’s speed is medium



When motor’s speed is high

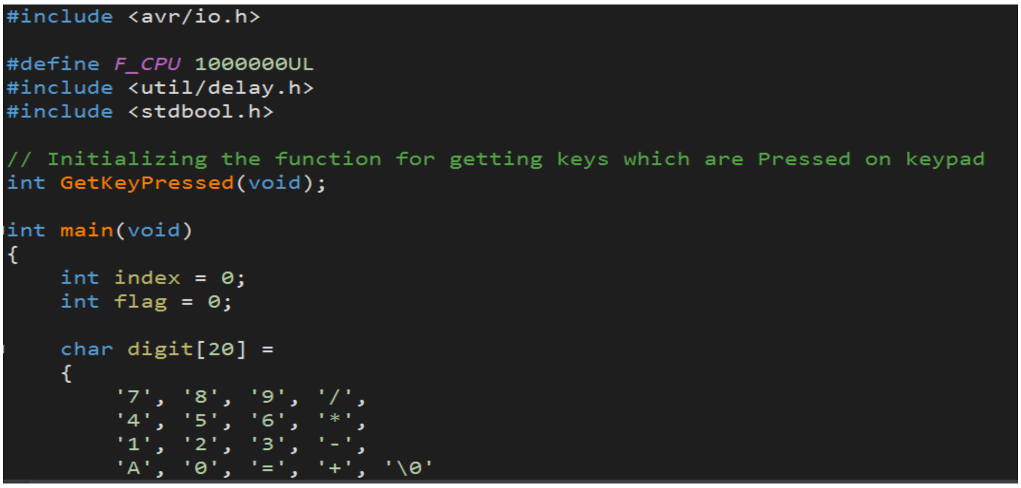


This all is implemented using embedded C language on Atmel studio as:

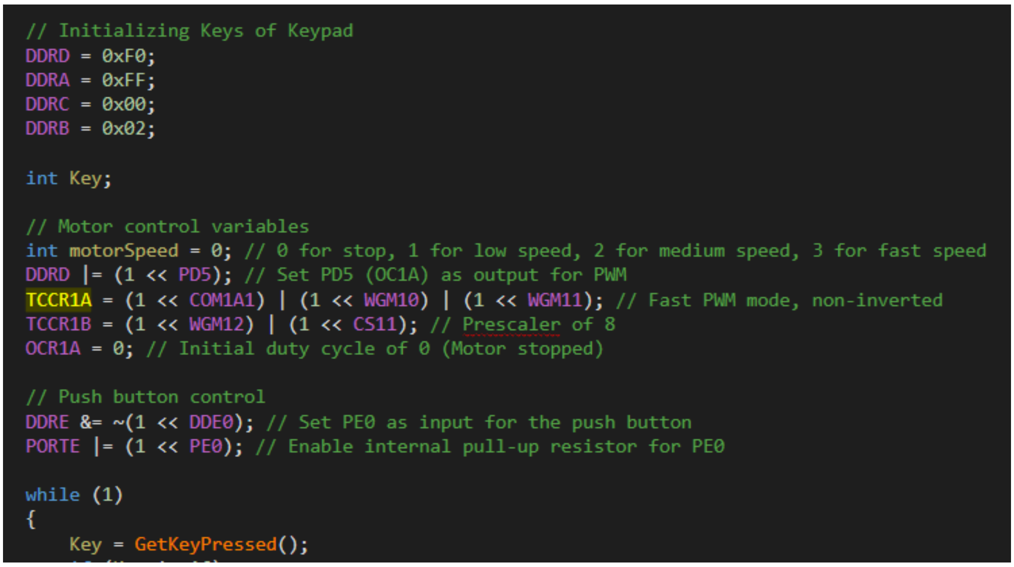
We have used GetKeyPressed function that returns the index value of keypad to main function, if the key is pressed. In main function first we check if the index is returned or not, if it is returned then we check Key pressed with our specified temperature range’s keys and if the pressed key matches with the any value in the range that we have stated then appropriate display will be shown on seven segment and motor will rotate by described speed. In code we have motor speed as:

- 0 for motor off
- 1 for low speed
- 2 for medium speed
- 3 for high/fast speed

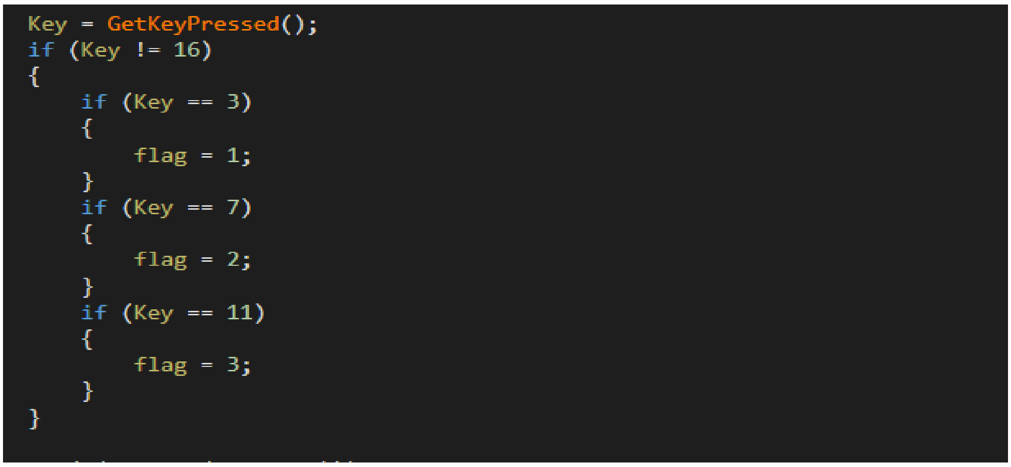
Embedded C-Code:



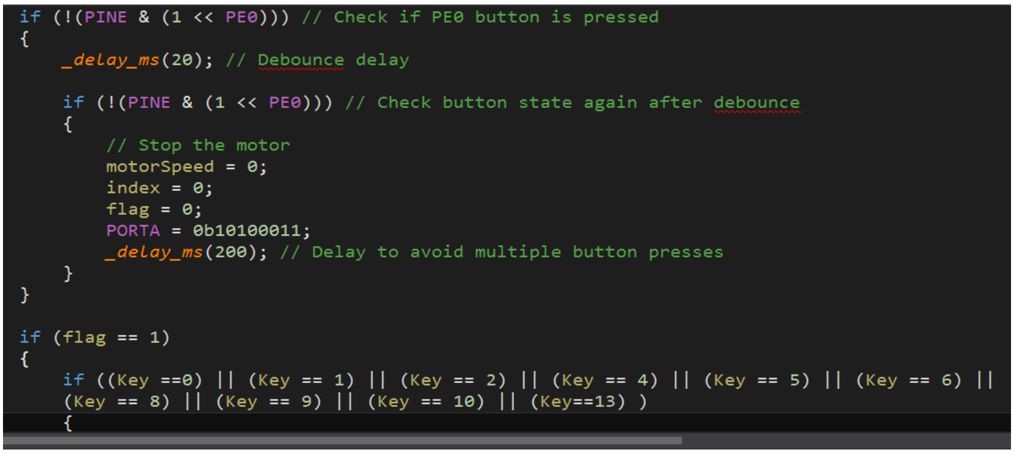
```c
#include <avr/io.h>

#define F_CPU 1000000UL
#include <util/delay.h>
#include <stdbool.h>

// Initializing the function for getting keys which are Pressed on keypad
int GetKeyPressed(void);

int main(void)
{
    int index = 0;
    int flag = 0;

    char digit[20] =
    {
        '7', '8', '9', '/',
        '4', '5', '6', '*',
        '1', '2', '3', '-',
        'A', '0', '=', '+', '\0'
```



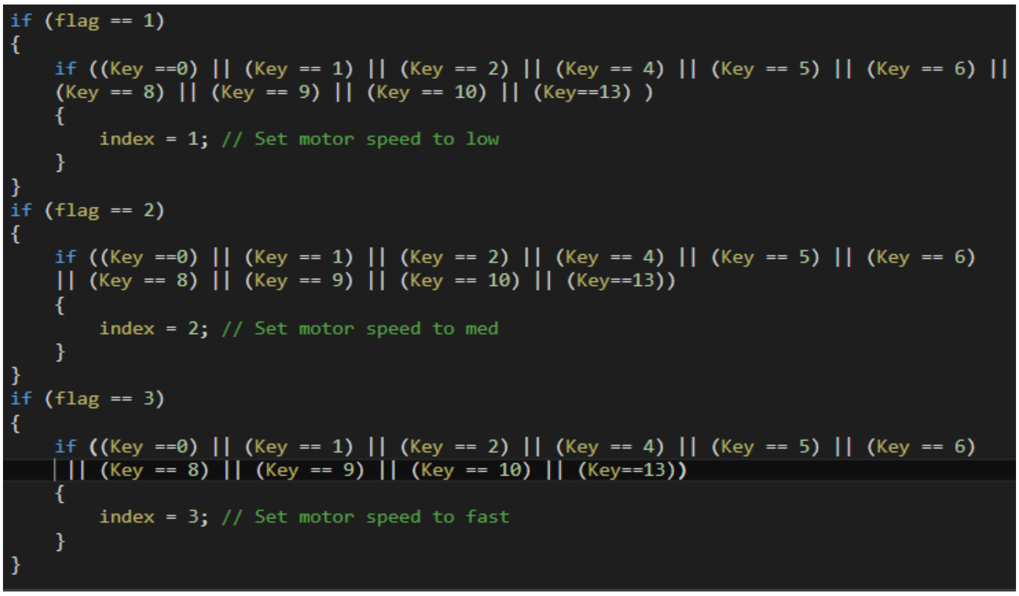
```c
    // Initializing Keys of Keypad
    DDRD = 0xF0;
    DDRA = 0xFF;
    DDRC = 0x00;
    DDRB = 0x02;

    int Key;

    // Motor control variables
    int motorSpeed = 0; // 0 for stop, 1 for low speed, 2 for medium speed, 3 for fast speed
    DDRD |= (1 << PD5); // Set PD5 (OC1A) as output for PWM
    TCCR1A = (1 << COM1A1) | (1 << WGM10) | (1 << WGM11); // Fast PWM mode, non-inverted
    TCCR1B = (1 << WGM12) | (1 << CS11); // Prescaler of 8
    OCR1A = 0; // Initial duty cycle of 0 (Motor stopped)

    // Push button control
    DDRE &= ~(1 << DDE0); // Set PE0 as input for the push button
    PORTE |= (1 << PE0); // Enable internal pull-up resistor for PE0

    while (1)
    {
        Key = GetKeyPressed();
```



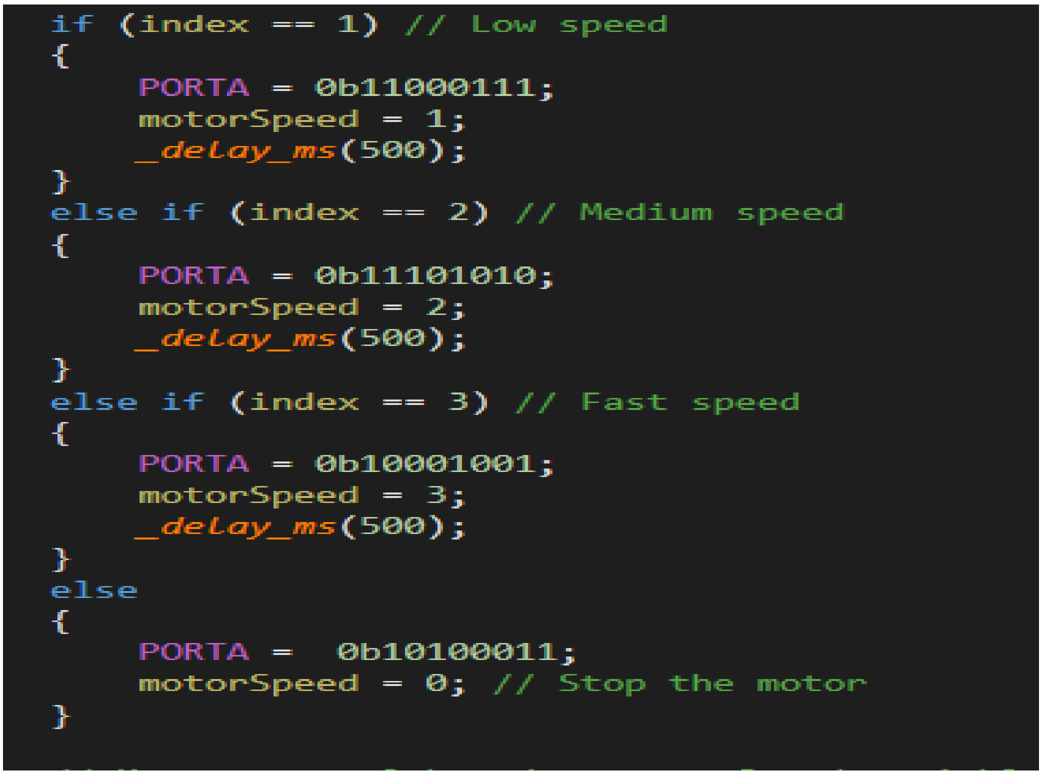
```
Key = GetKeyPressed();
if (Key != 16)
{
    if (Key == 3)
    {
        flag = 1;
    }
    if (Key == 7)
    {
        flag = 2;
    }
    if (Key == 11)
    {
        flag = 3;
    }
}
```



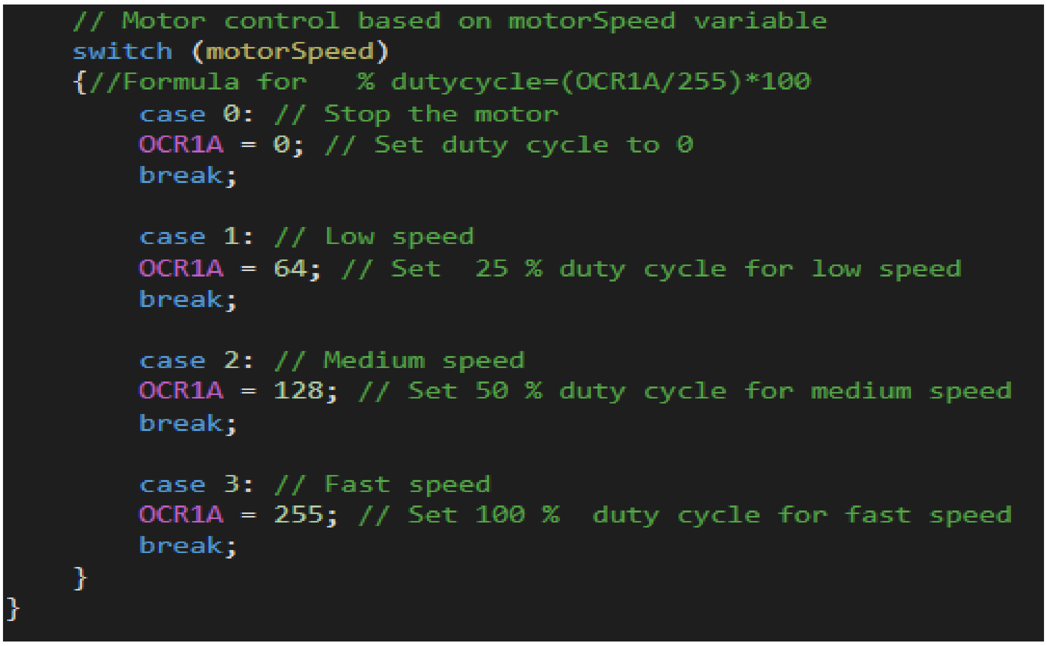
```
if (!(PINE & (1 << PE0))) // Check if PE0 button is pressed
{
    _delay_ms(20); // Debounce delay

    if (!(PINE & (1 << PE0))) // Check button state again after debounce
    {
        // Stop the motor
        motorSpeed = 0;
        index = 0;
        flag = 0;
        PORTA = 0b10100011;
        _delay_ms(200); // Delay to avoid multiple button presses
    }
}

if (flag == 1)
{
    if ((Key ==0) || (Key == 1) || (Key == 2) || (Key == 4) || (Key == 5) || (Key == 6) ||
    (Key == 8) || (Key == 9) || (Key == 10) || (Key==13) )
    {
```



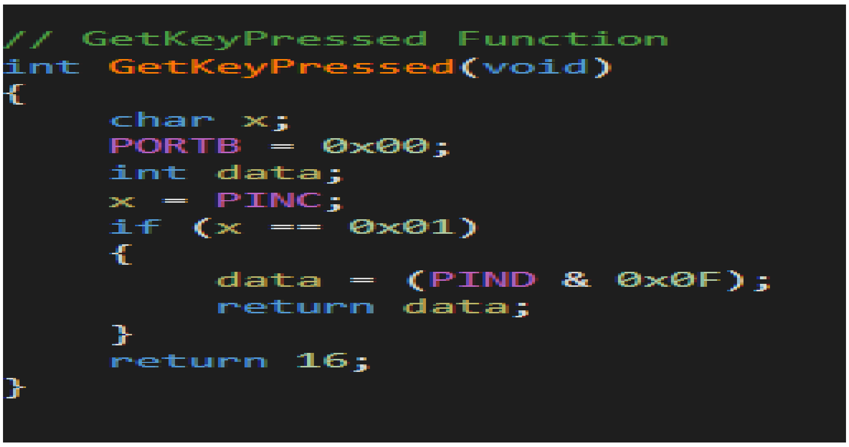
```
if (flag == 1)
{
    if ((Key ==0) || (Key == 1) || (Key == 2) || (Key == 4) || (Key == 5) || (Key == 6) ||
    (Key == 8) || (Key == 9) || (Key == 10) || (Key==13) )
    {
        index = 1; // Set motor speed to low
    }
}
if (flag == 2)
{
    if ((Key ==0) || (Key == 1) || (Key == 2) || (Key == 4) || (Key == 5) || (Key == 6)
    || (Key == 8) || (Key == 9) || (Key == 10) || (Key==13))
    {
        index = 2; // Set motor speed to med
    }
}
if (flag == 3)
{
    if ((Key ==0) || (Key == 1) || (Key == 2) || (Key == 4) || (Key == 5) || (Key == 6)
    || (Key == 8) || (Key == 9) || (Key == 10) || (Key==13))
    {
        index = 3; // Set motor speed to fast
    }
}
```



```
    if (index == 1) // Low speed
    {
        PORTA = 0b11000111;
        motorSpeed = 1;
        _delay_ms(500);
    }
    else if (index == 2) // Medium speed
    {
        PORTA = 0b11101010;
        motorSpeed = 2;
        _delay_ms(500);
    }
    else if (index == 3) // Fast speed
    {
        PORTA = 0b10001001;
        motorSpeed = 3;
        _delay_ms(500);
    }
    else
    {
        PORTA =  0b10100011;
        motorSpeed = 0; // Stop the motor
    }
```



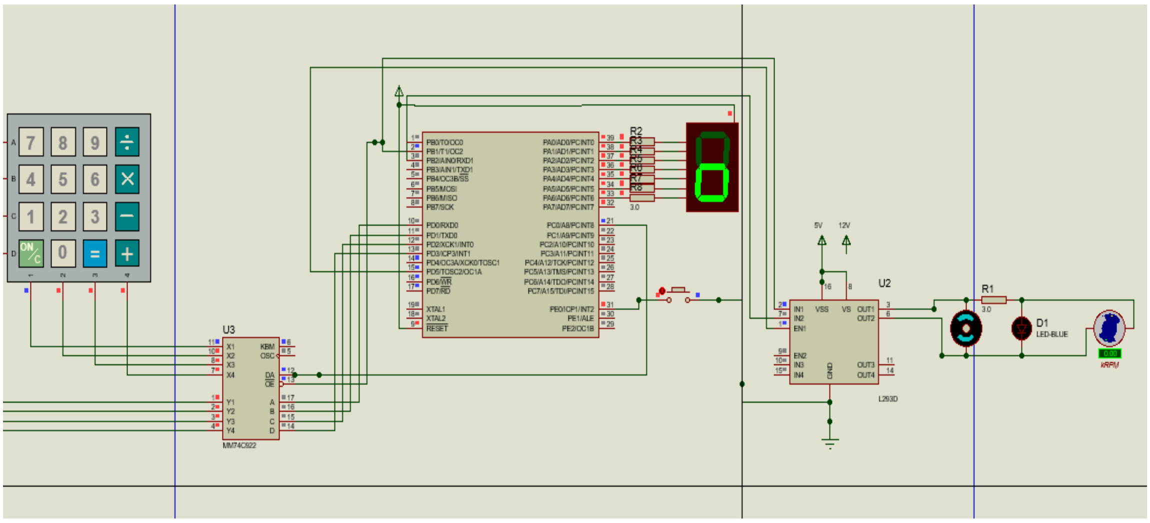
```
    // Motor control based on motorSpeed variable
    switch (motorSpeed)
    {//Formula for   % dutycycle=(OCR1A/255)*100
        case 0: // Stop the motor
        OCR1A = 0; // Set duty cycle to 0
        break;

        case 1: // Low speed
        OCR1A = 64; // Set  25 % duty cycle for low speed
        break;

        case 2: // Medium speed
        OCR1A = 128; // Set 50 % duty cycle for medium speed
        break;

        case 3: // Fast speed
        OCR1A = 255; // Set 100 %  duty cycle for fast speed
        break;
    }
}
```



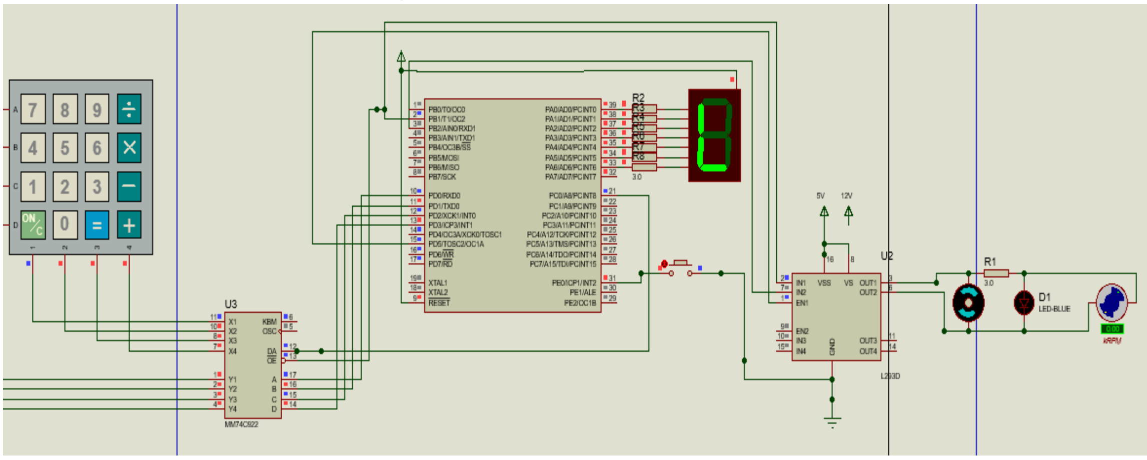
```
// GetKeyPressed Function
int GetKeyPressed(void)
{
        char x;
        PORTB = 0x00;
        int data;
        x = PINC;
        if (x == 0x01)
        {
                data = (PIND & 0x0F);
                return data;
        }
        return 16;
}
```

Schematic Diagram:

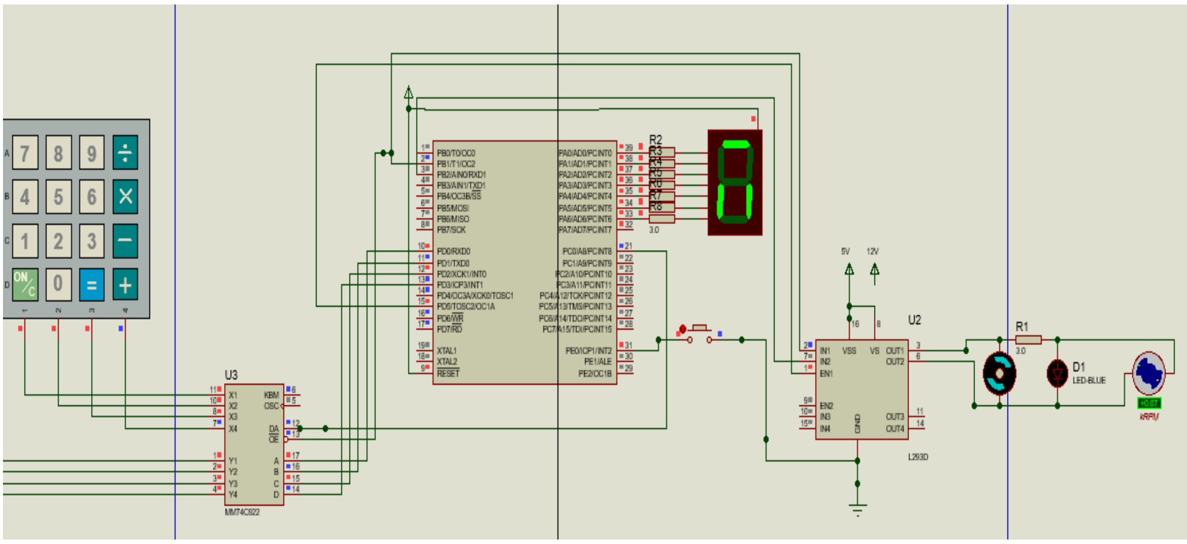
When no temperature is entered:



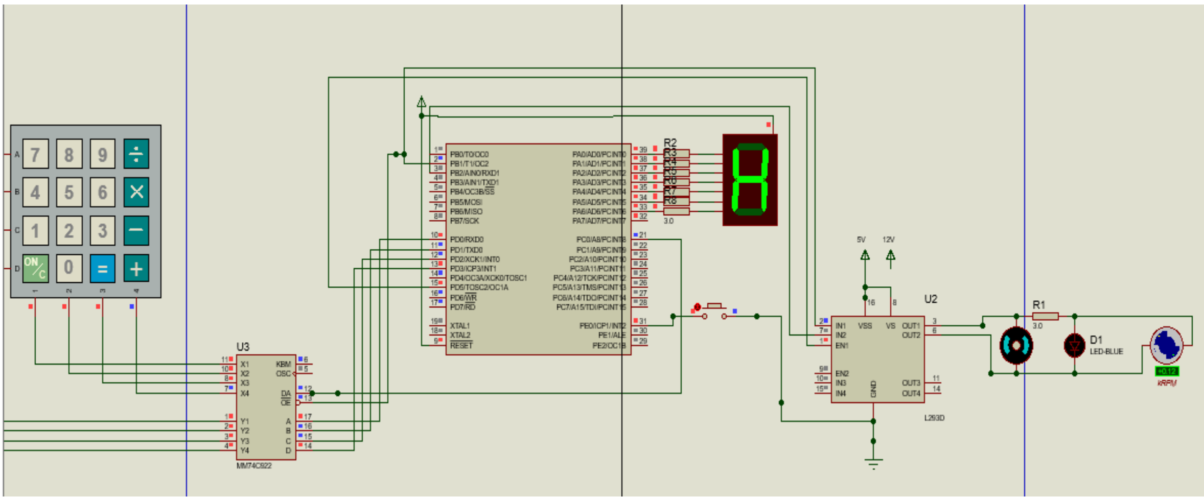
When temperature in the range of low is entered



When temperature in the range of medium is entered



When temperature in the range of high is entered


7
8
9
÷
4
5
6
×
1
2
3
−
ON/C
0
=
+
U3
X1
X2
X3
X4
KBM
OSC
DA
OE
Y1
Y2
Y3
Y4
A
B
C
D
MM74C922
R2
R3
R4
R5
R6
R7
R8
3.0
5V
12V
U2
IN1
IN2
EN1
EN2
IN3
IN4
VSS
VS
GND
OUT1
OUT2
OUT3
OUT4
L293D
R1
3.0
D1
LED-BLUE
+0.12
kRPM