

```
#include <xc.h>
```

```
#include <stdio.h>
```

```
float set_point[48]={
```

```
    25, 32.22, 39.44, 46.66, 53.88, 61.10, 68.32, 75.54, 82.76, 89.98, 97.20, 104.42, 111.64,  
    118.86, 126.08, 133.30, 140.52, 147.74, 154.96, // Pre Heat Time
```

```
    156.62, 158.28, 159.94, 161.60, 163.26, 164.92, 166.58, 168.24, 169.90, 171.56, 173.22,  
    174.88, 176.54, 178.20, 179.86, 181.52, 183.18, 184.84, // Soak Time
```

```
    191.32, 197.80, 204.28, 210.76, 217.24, 223.72, // Reflow Heat
```

```
    217.24, 210.76, 204.28, 197.80, 191.32 // Reflow Cooling
```

```
};
```

```
unsigned char Temp = 0, Time = 0, Htr_PWM_DS = 0, Fan_PWM_DS = 0;
```

```
void timer0_init();
```

```
unsigned int read_MAX6675();
```

```
void delay_us(unsigned char us);
```

```
unsigned char Htr_PWM_Cal(float setpoint, unsigned char currentpoint);
```

```
unsigned char Fan_PWM_Cal(float setpoint, unsigned char currentpoint);
```

```
void Pwm_Gen(unsigned char Duty_Cycle);
```

```
void UART_Init(void);
```

```
void UART_Write(char data);
```

```
void UART_Write_Text(const char *text);
```

```
void UART_Send_Temp(unsigned char time, unsigned int temperature);
```

```
void __interrupt() ISR(void)
```

```

{
    if(TMR0IF)
    {
        Temp = read_MAX6675();
        LATA = ~LATA;
        LATD = Temp;

        if(Time < 43)
        {
            RC0 = 1;
            Htr_PWM_DS = Htr_PWM_Cal(set_point[Time]/3, Temp);
            Pwm_Gen(Htr_PWM_DS);
        }
        else
        {
            RC0 = 0;
            Fan_PWM_DS = Fan_PWM_Cal(set_point[Time]/3, Temp);
            Pwm_Gen(Fan_PWM_DS);
        }

        UART_Send_Temp(Time, Temp); // <-- UART Transmission

        TMR0H = 0xE1;
        TMR0L = 0x67;
        TMR0IF = 0;
    }
}

```

```
        Time++;  
    }  
}  
  
void main(void)  
{  
    TRISA = 0x00;  
    PORTA = 0x00;  
    TRISB = 0x00;  
    LATB = 0x00;  
    TRISC = 0x00;  
    LATC = 0x00;  
    TRISD = 0x00;  
    LATD = 0x00;  
  
    TRISB0 = 1;  
  
    timer0_init();  
    UART_Init();  
  
    while(Time < 48)  
    {  
  
    }
```

```
    Time = 0;
}
```

```
void timer0_init()
```

```
{
    T0CON = 0b00000111;

    TMR0H = 0xE1;
    TMR0L = 0x67;
```

```
    INTCONbits.TMR0IF = 0;
    INTCONbits.TMR0IE = 1;
    INTCONbits.PEIE = 1;
    INTCONbits.GIE = 1;
```

```
    T0CONbits.TMR0ON = 1;
}
```

```
unsigned int read_MAX6675()
```

```
{
    unsigned int Temp_Max = 0;

    LATB2 = 0; //CS Low
    delay_us(10);
```

```

for(int i=0;i<16;i++)
{
    LATB4 = 1; // SCK high

    delay_us(1);

    Temp_Max <<= 1; // Shift data
    if(RB0 == 1)
    {
        Temp_Max |= 1;
    }

    LATB4 = 0; // SCK low

    delay_us(1);
}
LATB2 = 1; // CS high


if(Temp_Max & 0x0004)
{
    return 0xFFFF; //ThermoCouple Error
}


return (Temp_Max >> 5);
}


void delay_us(unsigned char us)
{

```

```

for(int i=0;i<us;i++)
{
    T1CONbits.TMR1ON = 0;

    TMR1H = 0xFF;
    TMR1L = 0xFE;

    TMR1IF = 0;

    T1CON = 0b00110001;

    while(!TMR1IF);
    T1CONbits.TMR1ON = 0;

    TMR1IF = 0;
}
}

unsigned char Htr_PWM_Cal(float setpoint, unsigned char currentpoint)
{
    float Kp = 2.0;
    float error = setpoint - currentpoint;
    float pwm_value = Kp * error;

    return (unsigned char)pwm_value;
}

```

```

unsigned char Fan_PWM_Cal(float setpoint, unsigned char currentpoint)
{
    float Kp = 2.0;

    float error = currentpoint - setpoint;

    float pwm_value = Kp * error;

    return (unsigned char)pwm_value;
}

```

```

void Pwm_Gen(unsigned char Duty_Cycle)
{
    if(Duty_Cycle > 100)
    {
        Duty_Cycle = 100;
    }
    else if (Duty_Cycle < 0)
    {
        Duty_Cycle = 0;
    }

    CCP1CON = 0b00001100;

    PR2 = 249;

    unsigned int pwm_val = (Duty_Cycle * 1023.0) / 100.0;

    CCPR1L = pwm_val >> 2;
}

```

```
CCP1CONbits.DC1B = pwm_val & 0x03;
```

```
T2CON = 0b00000111;
```

```
while (!PIR1bits.TMR2IF);
```

```
PIR1bits.TMR2IF = 0;
```

```
}
```

```
// UART Initialization
```

```
void UART_Init(void)
```

```
{
```

```
    TRISC6 = 0; // TX output
```

```
    TRISC7 = 1; // RX input
```

```
    SPBRG = 51; // For 9600 baud at 8 MHz (adjust if needed)
```

```
    BRGH = 1; // High speed
```

```
    SYNC = 0; // Asynchronous
```

```
    SPEN = 1; // Enable serial port
```

```
    TXEN = 1; // Enable TX
```

```
    CREN = 1; // Enable RX
```

```
}
```

```
// Transmit a single character
```

```
void UART_Write(char data)
```

```
{
```

```
    while(!TXIF);
```



```
TXREG = data;  
}
```

```
// Transmit a string
```

```
void UART_Write_Text(const char *text)
```

```
{  
    while(*text)  
        UART_Write(*text++);  
}
```

```
// Transmit temperature and time data
```

```
void UART_Send_Temp(unsigned char time, unsigned int temperature)
```

```
{  
    char buffer[32];  
    sprintf(buffer, "Time: %d, Temp: %d\n\n", time, temperature);  
    UART_Write_Text(buffer);  
}
```