Faculty of Engineering
BSc. (Hons.) in Engineering in Electronics and Telecommunications

EEE3305-Engineering Research (Embedded Systems)

Project

Automatic Temperature Detecting and Cooling System for Machinery Parts

Project Report

Group Details:

Date of Submission :- **19/01/2024**

Members :- **M. N. K. M. Bandara (21ug0496)**

S. M. S. M. Bandara (21ug0481)

Table of Contents

Abstract

This project uses a comprehensive system built around a PIC 16F877A microcontroller, DT11 sensor, and a variety of peripherals to attempt to address the critical need for effective temperature monitoring and control in industrial machinery. The main objective is to develop a complex system that can manage performance degradation, acquire data in real-time, adapt to cooling conditions, and switch over to a backup system with ease.

The DT11 sensor is used in the field of temperature and humidity sensing because of its accuracy and dependability. The PIC 16F877A microcontroller functions as the central processing unit, coordinating the different parts to make the system responsive and well-functioning. The system's ability to adjust to changing lighting conditions is improved by the addition of an LDR for analog-to-digital signal input.

Analog-to-digital converters, timers, interrupt signals, and PWM are all used in the hardware implementation. By starting the first fan when the temperature hits 31°C, PWM control optimizes thermal management by facilitating the gradual activation of cooling fans. A performance degradation strategy is triggered when the temperature reaches 37°C. This strategy modifies the machine runtime that is available and indicates that a backup system transition is about to occur.

Driven by MOSFETs, the two-tiered cooling system efficiently reduces temperature spikes, protecting machinery from possible damage. Moreover, remote control and monitoring are made possible by the addition of an ESP32 Wi-Fi module. The system exhibits adaptability to dynamic operational scenarios as it can be remotely turned on or off.

The testing phase's results show how well the system responds to temperature changes. Temperature and humidity data are reliably recorded by the DT11 sensor, processed quickly, and shown on an LCD panel. At the predetermined thresholds, the cooling fans operate smoothly, demonstrating the system's real-time adaptability.

The LCD's "Avail. time" performance degradation strategy illustrates how the machine output can be subtly changed by the system in response to temperature readings. The backup machine seamlessly takes over, guaranteeing uninterrupted operation even in demanding thermal environments.

An analog-to-digital signal input from an LDR is used to increase the adaptability of the system and provide information about ambient light conditions that could affect temperature readings. Long-term temperature data storage is made possible by the inclusion of a micro-SD card, which opens up possibilities for thorough analysis and the identification of historical trends.

In terms of the future, the project establishes the framework for improvements and developments. Based on temperature trends, the possibility of using machine learning algorithms for predictive maintenance is investigated. Furthermore, it is intended to enhance the system's functionality in various industrial environments by integrating increasingly sophisticated sensors and communication protocols.

To sum up, this project provides a thorough method for controlling and monitoring temperature in industrial machinery.

Introduction

Background of the project:

Considering how quickly technology is developing today, it is essential that smart systems be implemented into a range of industries. For machinery to operate at peak efficiency, last a long time, and not overheat and cause damage, temperature monitoring and control are essential. By creating a digital thermometer and an intelligent cooling system, the proposed mini project seeks to meet this need.
Machinery frequently operates under strict temperature requirements, especially in industrial settings. Exceeding predetermined thresholds for variations can result in inefficiencies, increased wear and tear, and, in the worst situations, catastrophic failures. Maintaining operational efficiency and avoiding expensive delays requires the integration of both an active cooling mechanism and a real-time temperature monitoring system.

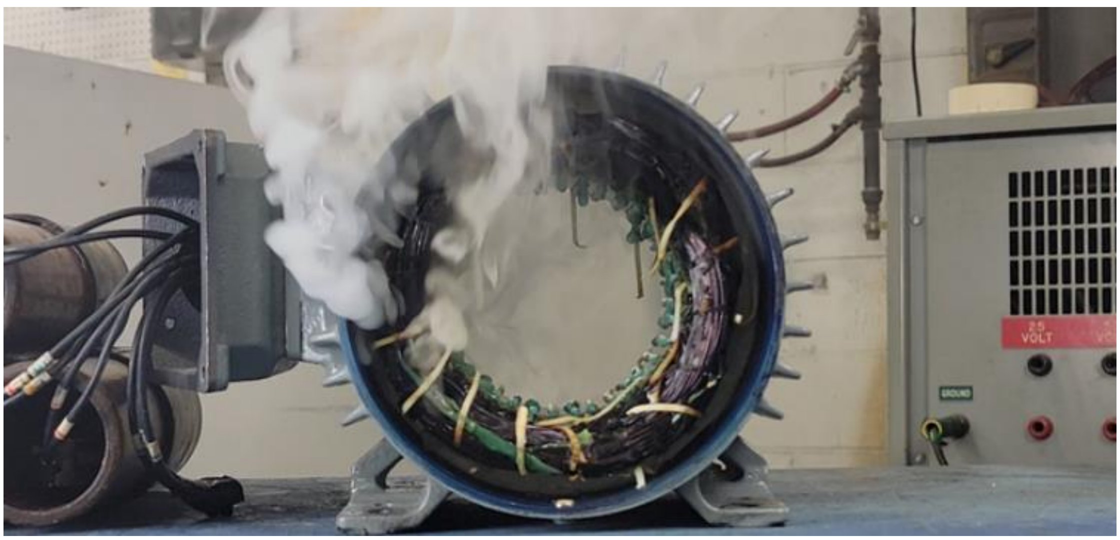

Figure1: overheated machine

Project Description:

In this project, an DT11 temperature and humidity sensor and a PIC16F877A microcontroller will be used to design and implement a digital thermometer. A 16x2 LCD screen will show the temperature readings in real time, offering a user-friendly interface for quick evaluation. Furthermore, an active cooling system will be included, which will start working automatically when the temperature rises above certain thresholds.
The addition of a timer to the project gives it additional functionality. The temperature maximum and minimum over predetermined periods will be recorded and shown. By enabling the analysis of temperature trends, this feature helps engineers and operators optimize machinery performance and perform preventive maintenance.

Objectives

The primary objectives of this project are as follows:

Real-time Temperature Monitoring: Develop a robust system capable of accurately detecting and displaying real-time temperature and humidity data using the DT11 sensor.

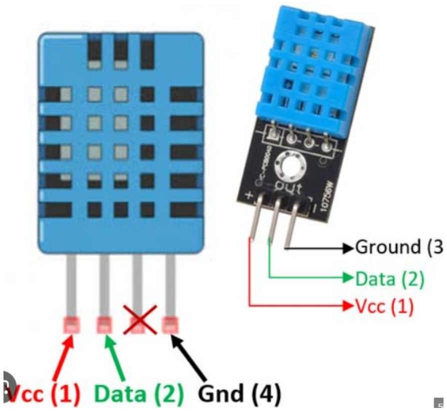

Figure 2: DT11 Temperature and Humidity Sensor

Cooling System Activation: Implement a cooling mechanism that activates when the temperature exceeds predefined thresholds, promoting optimal working conditions for industrial machinery.

Performance Degradation Strategy: Integrate a performance degradation strategy using PWM signals to gradually decrease machine performance when the temperature reaches critical levels.

Backup System Activation: Implement a backup system that automatically activates when the primary machine ceases operation, ensuring continuity and minimizing downtime.

Remote System Control: Utilize an ESP32 Wi-Fi module to enable remote control capabilities, allowing for shutdown and activation of the entire system.

Figure 3: ESP32 Module

Data Logging: Incorporate a micro-SD card for data storage, enabling the recording of temperature data for analysis and future reference.

Methodology

Hardware Implementation:

A PIC 16F877A microcontroller, DT11 and LM35 temperature sensors, an LDR for analog-to-digital signal input, two MOSFET-controlled cooling fans, a micro-SD card for data storage, and an ESP32 Wi-Fi module for remote control are all used in the hardware setup. To support the various functions of the system, the circuit design, which is implemented on a PCB, includes timers, interrupt signals, and analog-to-digital converters.

Temperature Sensing Technologies:

The study of temperature sensing technologies provides the overview of this project. There are many different types of sensors on the market, each with advantages and disadvantages. For our project, the LM35 was chosen because of its exceptional consistency and accuracy. Its simple analog output based on temperature has been widely used in many applications, as shown by literature, which makes it a perfect fit for our real-time temperature monitoring system.

Microcontroller Applications:

Our project is not complete without the PIC 16F877A microcontroller. The review of the literature falls into case studies and current applications that demonstrate PIC microcontrollers' effective use in temperature monitoring systems. PIC microcontrollers are widely used in a variety of engineering projects due to their well-known adaptability, low power consumption, and strong interfacing capabilities.

Digital Thermometer Implementations:

There are many different digital thermometer implementations available on the market that serve a variety of purposes. The survey examines these implementations in detail, paying particular attention to their features and methods. Sensor interfaces, data processing algorithms, and display mechanisms are important components. Comprehending these current solutions facilitates the identification of optimal methodologies and possible improvements for our task.

Past Technologies:

1) Analog Thermometers:

- Traditional analog thermometers, such as mercury or bimetallic strip thermometers, were commonly used in the past.
- Limited precision and sensitivity compared to modern digital solutions.
- Difficulty in real-time monitoring and data recording.

2) Mechanical Temperature Gauges:

- Utilized mechanical components to measure and display temperature.
- Often found in older machinery, these gauges provided a basic indication of temperature changes.
- Limited accuracy and lacked digital data processing capabilities.

The literature review ends with a summary of the main conclusions. Prior solutions frequently draw attention to the difficulties encountered and creative strategies used in digital thermometer projects. The significance of accuracy, real-time monitoring, and user-friendly interfaces are recurring themes. By using these insights, we hope to create a digital thermometer that not only complies with industry standards but also includes improvements based on the flaws noted in previous research.

Present Technologies:

1) Digital Thermometers with NTC/PTC Sensors:

- Implementation of digital thermometers with Negative Temperature Coefficient (NTC) or Positive Temperature Coefficient (PTC) sensors.
- Microcontrollers, such as PIC series, are commonly used for signal processing and interfacing.
- Enhanced accuracy and the ability to display real-time temperature readings.

2) Infrared Thermometers:

- Non-contact temperature measurement using infrared technology.
- Suitable for applications where direct contact with the object is challenging.
- Popular for industrial use but may not be as accurate as direct-contact sensors like the LM35.

3) Wireless Temperature Monitoring Systems:

- Integration of wireless communication protocols for remote temperature monitoring.
- Bluetooth and Wi-Fi-enabled devices allow users to monitor temperatures from a distance.
- Often used in industrial settings for machinery distributed across large areas.

Future Technologies:

1) IoT-Enabled Temperature Monitoring:

- Integration with the Internet of Things (IoT) for enhanced connectivity and data analysis.
- Real-time data transmission to cloud platforms for centralized monitoring and analytics.

2) Advanced Sensor Technologies:

- Development of advanced temperature sensors with improved precision and stability.

- Implementation of sensors based on emerging technologies like graphene or MEMS (Micro-Electro-Mechanical Systems).

3) Machine Learning for Predictive Maintenance:

- Implementation of machine learning algorithms to predict machinery failures based on temperature patterns.

- Integration of predictive maintenance strategies for improved efficiency and reduced downtime.

The review of the literature looks at fresh innovations that could have an impact on temperature monitoring in the future in addition to existing solutions. This covers developments in energy-efficient microcontroller applications, communication protocols, and sensor technologies. In a rapidly changing technological landscape, it is imperative that we understand these emerging trends in order to ensure the continued use and relevance of our digital thermometer.

Sensor Integration:

The project utilized a DT11 sensor for temperature and humidity measurements. The sensor data was processed by a PIC 16F877A microcontroller, utilizing the capabilities of its analog-to-digital converters (ADC) for accurate temperature readings.

Cooling System:

Two cooling fans were installed to regulate temperature. The first fan activated when the temperature reached 31°C, providing immediate cooling. The second fan, initiated at 37°C, operated in conjunction with a gradual reduction in the working performance of the primary machine through PWM signals.

Backup Mechanism:

To ensure continuous operation, a backup mechanism was incorporated. When the primary machine ceased functioning, the second machine seamlessly took over, preventing downtime.

Data Logging:

Temperature data was logged using a micro-SD card for future analysis. This feature facilitates long-term monitoring, trend analysis, and predictive maintenance.

Wi-Fi Module:

An ESP32 Wi-Fi module was integrated to enable remote control of the system. This allowed for system shutdown and initiation of the second cooling fan when necessary.

Programming:

The entire system was programmed using the C language in MPLAB, incorporating timers, interrupt signals, PWM, and ADC functionalities.

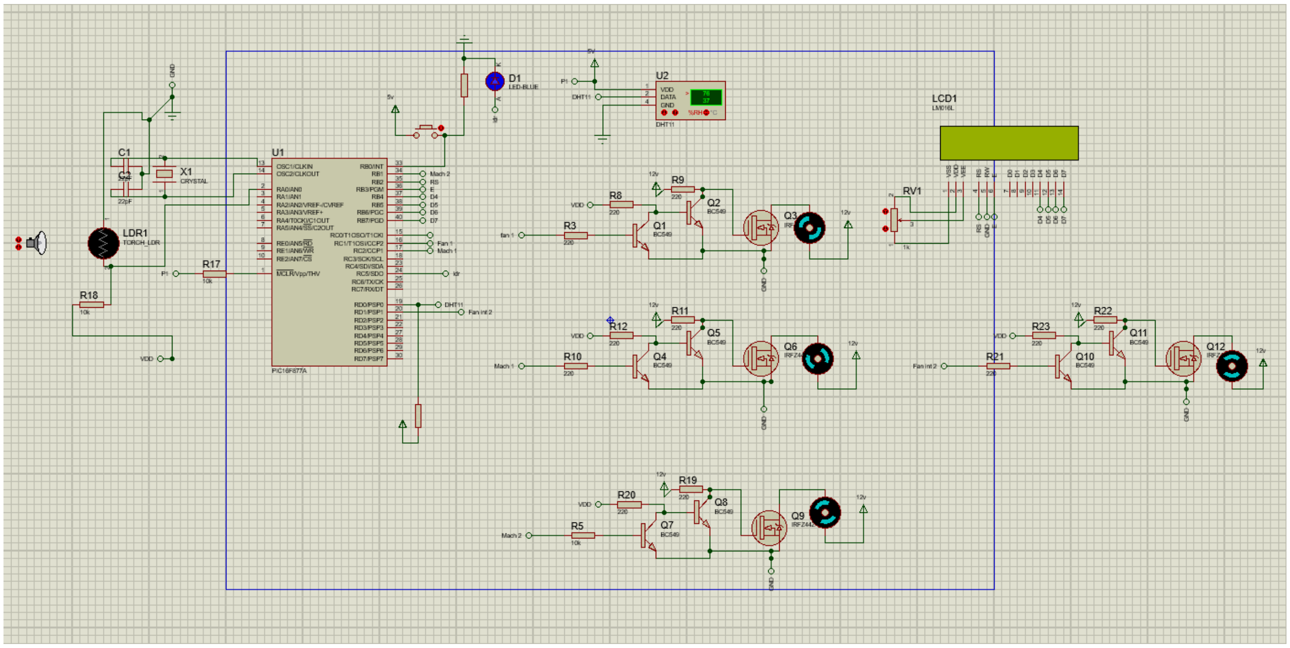

Figure 4: proteus simulation

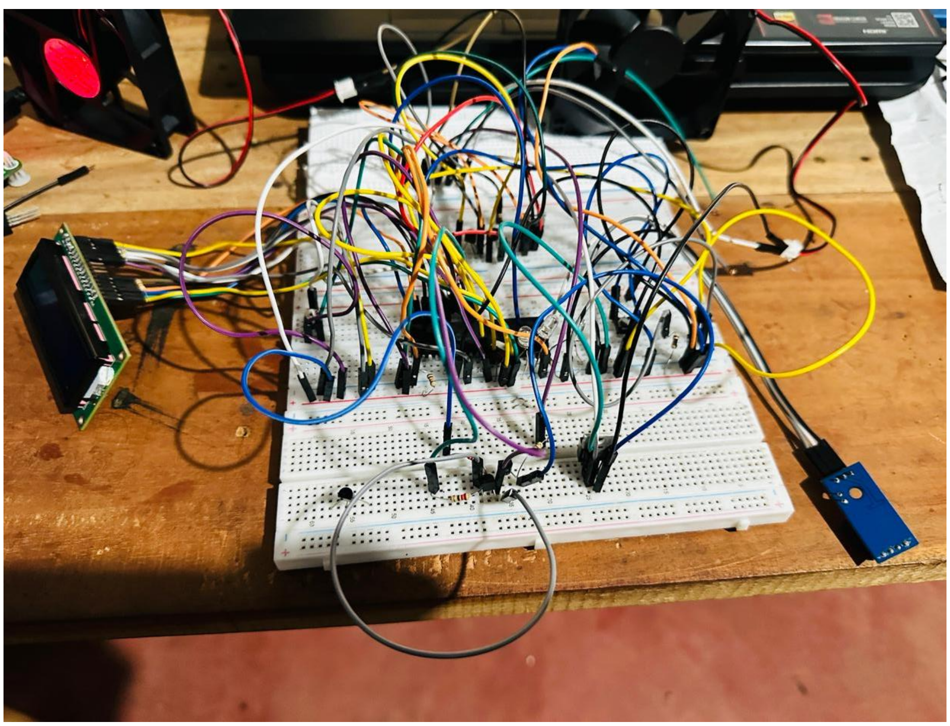

Figure 5: Breadboard circuit

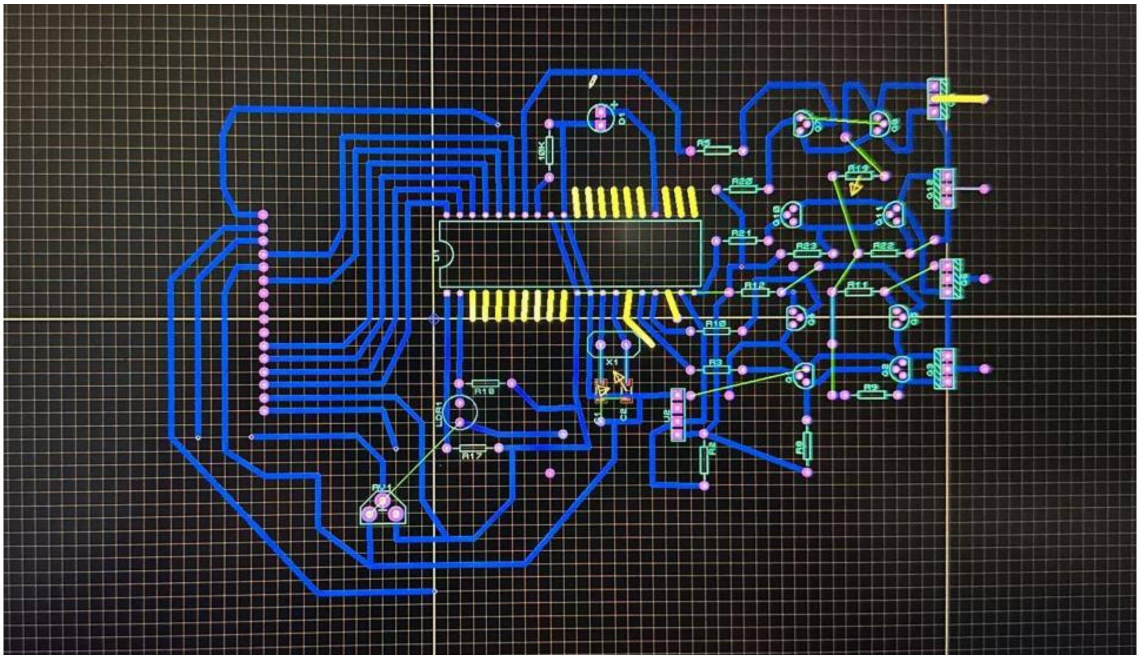

Figure 6: PCB Design

Software Development:

The firmware for the PIC 16F877A microcontroller is programmed to handle temperature data acquisition from the sensors. Timers and interrupt signals are utilized for efficient control of cooling fans and performance degradation. PWM signals regulate the gradual decrease in machine performance based on the sensed temperature. The ESP32 module is programmed to remotely control the system and initiate the backup machine.

Results and Discussion

The outcomes of the temperature control system's implementation demonstrate how well it monitors and regulates the temperature of machinery. The DT11 sensor, which is renowned for its precision, was essential in delivering temperature readings in real time. Using the sensor data, the PIC 16F877A microcontroller integration allowed for quick and accurate decision-making.

Efficiency of the Cooling System:

The two fans that make up the cooling system proved to be effective in keeping the temperature within the intended range. The first fan's activation at 31°C successfully stopped any rapid temperature increase, demonstrating the system's quick response to temperature changes. By starting the second fan at 37°C and gradually reducing the primary machine's output, the temperature drop was managed, preventing unexpected shutdowns, and facilitating a more seamless switch to the backup equipment.

PWM-Controlled Performance Reduction:

The system became more complex when Pulse Width Modulation (PWM) signals were used to regulate the performance of the main machine. The PWM signals dynamically changed the machine's output as the temperature got closer to the critical threshold, enabling a slow decrease in operating performance. This avoided sudden stops and enhanced the machine's energy efficiency by adjusting its operation according to temperature changes.

Safety Measure Reliability:

The system's dependability and failover capacity were shown by the smooth transfer to the backup machinery when the main machine failed. In industrial settings where uninterrupted operation is critical, this feature is essential for preventing downtime and guaranteeing continuous production.

Data Logging and Analysis:

The addition of a micro-SD card for data logging gave post-analysis and trend detection a useful tool. Predictive maintenance can make use of the temperature data that has been stored to identify possible problems before they become serious failures. By adding this feature, the system's capabilities go beyond real-time monitoring and help with the assessment of the long-term health of the machinery.

Conclusion and Future Works

Overall System Performance:

To sum up, the temperature control system that was put in place is effective in meeting its goals of providing real-time monitoring, effective cooling, and smooth machinery transitions. For industrial applications, the marriage of sophisticated control mechanisms, microcontroller intelligence, and sensor accuracy results in a dependable and adaptable system.

Future Improvements:

Despite the system's excellent performance, there is room for improvement in the future. The incorporation of machine learning algorithms to forecast temperature trends and anomalies is one possible area for development. This could make it possible to take preventative maintenance actions and offer insights into the health of the machinery.

More fine-tuning of the PWM control mechanism may result in more subtle machine performance adjustments, enabling optimal operation at different temperatures. A comprehensive monitoring system could benefit from the integration of additional sensors, such as load or vibration sensors, which could offer a more thorough understanding of the health of the machinery.

It is possible to extend the functionality of the Wi-Fi module by adding remote control and monitoring via a web dashboard or a dedicated user interface. This would improve accessibility and convenience by allowing operators to evaluate the state of the system and make changes from one central location.

References:

1. https://smarterhouse.org/cooling-systems/types-cooling-systems
2. https://www.britannica.com/technology/cooling-system
3. https://www.espressif.com/en/products/socs/esp32
4. https://www.microchip.com/en-us/product/pic16f877a
5. https://opentextbc.ca/basichvac/chapter/mechanical-cooling-general-applications/

Appendix

```
#include "config.h"
#include "dht11.h"
#include "lcd.h"
#include <xc.h>
#include "dhtread.h"

int i;
void motor(unsigned int dutyCycle){
   CCPR1L = dutyCycle;
   __delay_ms(10);
}
float temp;
void I2C_Master_Init(const unsigned long c)
{
  SSPCON = 0b00101000;         //SSP Module as Master
  SSPCON2 = 0;
  SSPADD = (_XTAL_FREQ/(4*c))-1; //Setting Clock Speed
  SSPSTAT = 0;
  TRISC3 = 1;                //Setting as input as given in datasheet
  TRISC4 = 1;                //Setting as input as given in datasheet
}
void I2C_Master_Wait()
{
  while ((SSPSTAT & 0x04) || (SSPCON2 & 0x1F)); //Transmit is in progress
}
void I2C_Master_Start()
{
  I2C_Master_Wait();
  SEN = 1;           //Initiate start condition
}
void I2C_Master_RepeatedStart()
{
  I2C_Master_Wait();
  RSEN = 1;          //Initiate repeated start condition
}
void I2C_Master_Stop()
{
  I2C_Master_Wait();
  PEN = 1;          //Initiate stop condition
}
void I2C_Master_Write(unsigned d)
{
  I2C_Master_Wait();
```

```
  SSPBUF = d;        //Write data to SSPBUF
}
unsigned short I2C_Master_Read(unsigned short a)
{
  unsigned short temp;
  I2C_Master_Wait();
  RCEN = 1;
  I2C_Master_Wait();
  temp = SSPBUF;      //Read data from SSPBUF
  I2C_Master_Wait();
  ACKDT = (a)?0:1;    //Acknowledge bit
  ACKEN = 1;          //Acknowledge sequence
  return temp;
}
int main() {
    TRISB=0x01;
    TRISD=0x01;
    TRISC=0x40;
    CCP1CON = 0x0F;// Select the PWM mode.
    T2CKPS0 = 0;
    T2CKPS1 = 1;
    PR2 = 100; // Set the Cycle time to 100 for varying the duty cycle from 0-100
//    CCPR1L = 50;    // By default set the dutyCycle to 50
    T1CON=0x01;
    TMR2ON = 1;     //Start the Timer for PWM generation
    Lcd_Init();
    TRISC6=0;   //Output (TX)
    TRISC7=1;   //Input (RX)
    INTCONbits.GIE=1;
    INTCONbits.PEIE=1;
    INTCONbits.INTE=1;
    OPTION_REGbits.INTEDG=1;
    while(1){
        dht11_init();
        find_response();
        if(Check_bit == 1){
            read_dht();
            if(Sumation == ((RH_byte_1+RH_byte_2+Temp_byte_1+Temp_byte_2) & 0XFF)){
                set_temp_hum();
                Lcd_Clear();
                disp_act_data();
                float temp= ((e/100));
                if(temp>30){RC1=1;}else{RC1=0;}
                if(temp>36 && RB1==0){
                    for(i=0;i<420;i++) {
                        if(temp<30){break;}
```

```
                TMR1H=TMR1L=0;
                while(!TMR1IF);
                TMR1IF=0;
                 Lcd_Set_Cursor(2,1);
                Lcd_Write_String("Avail Time:");
                int z=(10-(i/42));
                Lcd_Write_Char((z/10)+'0');
                Lcd_Write_Char((z%10)+'0');
                if(i<100){
                   motor(100);
                }
                else if(i<200){
                   motor(80);
                }
                else if(i<300){motor(40);}
                else if(i<420){motor(0);}
                RB1=1;
             }

          }
          if(temp<30){motor(100);RB1=0;}
   }
   return;
      }}}

void interrupt ghj() {
   if(INTCONbits.INTF==1){
      RD1=~RD1;
   }INTF=0;}
```