

```
#include <stdio.h>
#include "Adafruit_LiquidCrystal.h"
#include <DHT22.h>
```

```
DHT22 dht22(2);
```

```
Adafruit_LiquidCrystal lcd(0);
```

```
int unitTransfer=1000;
```

```
byte customChar[8] = {
  0b00000,
  0b00100,
  0b00110,
  0b11111,
  0b11111,
  0b00110,
  0b00100,
  0b00000
};
```

```
byte customChar1[8] = {
  0b11111,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
}
```

```
0b00000
};

int stage = -1;
int onOFF = 0;
int time2;
int COL;
int ROW=1;

int NEXT;
int OK;
int BACK;

int noteDuration = 300;

int melody[] = {
  262, 262, 392, 392, 440, 440, 392,
};

int MELODY[] = {
  349, 349, 330, 330, 294, 294, 262,
};

const int door = 5;
const int buzzerPin = 6;
const int pinReset = 7;
const int buttonBACK = 11;
const int buttonOK = 12;
const int buttonNEXT = 13;
```

```
const int heater = 8;
```

```
const int fan = 10;
```

```
const int motor = 9;
```

```
float humidityMeasure = 100.0;
```

```
float temperatureMeasure = 0.0;
```

```
int humidity;
```

```
int temperature;
```

```
int Time=200;
```

```
unsigned long timeRemain;
```

```
unsigned long start;
```

```
unsigned long stop;
```

```
unsigned long sensorSTART;
```

```
unsigned long sensorSTOP;
```

```
int temperature_10=7;
```

```
int temperature_1=0;
```

```
int digitTemp=9;
```

```
int digitTime=5;
```

```
int time_100=1;
```

```
int time_10=0;
```

```
int time_1=0;
```

```
void printLogo(void);
```

```
void readInput(void);
```

```
void stage00(void);
```

```
void stage0(void);
void stage1(void);
void stage2(void);
void stage3(void);
void stage4(void);
void stage5(void);
void stage6(void);
void stage7(void);
void stage8(void);
void stage9(void);
void stage10(void);
void stage11(void);
void stage20(void);
void stopCheck(void);
void doorCheck(void);
void ender(void);
void work(void);
void stopWork(void);
void TemHumTest(void);
void controlTemp(void);

void setup() {

  Serial.begin(9600);
  delay(1000);
  Serial.write(124);
  delay(100);
  Serial.print("p");
  delay(100);
```

```
Serial.begin(38400);

digitalWrite(pinReset, HIGH);
delay(200);
pinMode(pinReset, OUTPUT);
lcd.setBacklight(HIGH);
if(!lcd.begin(20, 4)){
  Serial.print("LCD wrong");
  while(1);
}
else{
  Serial.println("good");
}

pinMode(buzzerPin, OUTPUT);
pinMode(buttonNEXT, INPUT_PULLUP);
pinMode(buttonOK, INPUT_PULLUP);
pinMode(buttonBACK, INPUT_PULLUP);
pinMode(door, INPUT_PULLUP);

pinMode(heater, OUTPUT);
digitalWrite(heater, LOW);
pinMode(fan, OUTPUT);
pinMode(motor, OUTPUT);

lcd.createChar(0, customChar);
lcd.createChar(1, customChar1);
Serial.println("done setup");
}
```

```
void loop() {  
  if (onOFF==0){  
    stage00();  
    stage0();  
    stage1();  
    stage2();  
    stage3();  
    stage5();  
    stage6();  
    stage7();  
    stage8();  
    stage9();  
    stage10();  
    stage20();  
  }  
  else{  
    doorCheck();  
    stopCheck();  
    ender();  
    stage4();  
    stage11();  
    TemHumTest();  
    controlTemp();  
  }  
}
```

```
void printLogo(void) {
```

```
Serial.println("in printlogo");
delay(1000);
lcd.setCursor(0, 0);
for (COL = 0; COL < 20; COL++) {
  lcd.write(0xFF);
  delay(50);
}
lcd.setCursor(19, 1);
lcd.write(0xFF);
delay(50);
lcd.setCursor(19, 2);
lcd.write(0xFF);
delay(50);
for (COL = 19; COL > -1; COL--) {
  lcd.setCursor(COL, 3);
  lcd.write(0xFF);
  delay(50);
}
lcd.setCursor(0, 2);
lcd.write(0xFF);
delay(50);
lcd.setCursor(0, 1);
lcd.write(0xFF);
delay(50);

lcd.setCursor(6, 1);
lcd.print(F("VANT 151"));
lcd.setCursor(7, 2);
lcd.print(F("Team 4"));
```

```
}
```

```
void readInput(void) {
```

```
  do {
```

```
    NEXT = digitalRead(buttonNEXT);
```

```
    OK = digitalRead(buttonOK);
```

```
    BACK = digitalRead(buttonBACK);
```

```
  } while (NEXT == 1 && OK == 1 && BACK == 1);
```

```
  while (digitalRead(buttonNEXT) == 0 || digitalRead(buttonOK) == 0 || digitalRead(buttonBACK) == 0) {
```

```
    delay(10);
```

```
  }
```

```
}
```

```
void stage00(void) {
```

```
  if (stage == -1) {
```

```
    printLogo();
```

```
    readInput();
```

```
    if (NEXT == 0 || OK == 0 || BACK == 0) {
```

```
      lcd.clear();
```

```
      stage = 1;
```

```
    }
```

```
  }
```

```
}
```

```
void stage0(void) {
```

```
  if (stage == 0) {
```

```
    lcd.setCursor(0, 0);
```

```
    lcd.print(F("      Back"));
```



```

lcd.setCursor(0, 1);
lcd.print(F("  Press OK  "));
lcd.setCursor(0, 2);
lcd.print(F("  To turn off  "));
lcd.setCursor(0, 3);
lcd.print(F("          "));
do {
  OK = digitalRead(buttonOK);
  BACK = digitalRead(buttonBACK);
} while (OK == 1 && BACK == 1);

while (digitalRead(buttonOK) == 0 || digitalRead(buttonBACK) == 0) {}
if (OK == 0){
  lcd.clear();
  lcd.setBacklight(LOW);
  readInput();
  lcd.setBacklight(HIGH);
  stage=-1;
}
else if (BACK == 0){
  stage=1;
}
}
}

void stage1(void) {
  if (stage == 1) {
    lcd.setCursor(0, 0);
    lcd.print(F("          "));

```

```
lcd.setCursor(0, 1);
lcd.print(F(" 1. Manu  "));
lcd.setCursor(0, 2);
lcd.print(F(" 2. Auto  "));
lcd.setCursor(0, 3);
lcd.print(F("      "));
lcd.setCursor(5, ROW);
lcd.write((byte)0);
readInput();
if (NEXT == 0) {
  lcd.setCursor(5, ROW);
  lcd.print(" ");
  ROW++;
  if (ROW == 3) {
    ROW = 1;
  }
  lcd.setCursor(5, ROW);
  lcd.write((byte)0);
}
else if (OK == 0) {
  if (ROW == 1) {
    stage = 5;
  }
  else {
    stage = 2;
  }
}
else {
  stage=0;
}
```

```

    }
}
}

void stage2(void) {
    if (stage == 2) {
        lcd.setCursor(0, 0);
        lcd.print(F("Temperature:   "));
        lcd.setCursor(0, 1);
        lcd.print(F("  1. High   "));
        lcd.setCursor(0, 2);
        lcd.print(F("  2. Middle  "));
        lcd.setCursor(0, 3);
        lcd.print(F("  3. Low    "));
        lcd.setCursor(5, ROW);
        lcd.write((byte)0);
        readInput();
        if (NEXT == 0) {
            lcd.setCursor(5, ROW);
            lcd.print(" ");
            ROW++;
            if (ROW == 4) {
                ROW = 1;
            }
            lcd.setCursor(5, ROW);
            lcd.write((byte)0);
        }
        else if (OK == 0) {
            stage++;
        }
    }
}
}
}

```

```
if (ROW == 1) {  
    temperature = 85;  
}  
else if (ROW == 2) {  
    temperature = 70;  
}  
else {  
    temperature = 50;  
}  
}  
else {  
    stage--;  
}  
}  
}
```

```
void stage3(void) {  
    if (stage == 3) {  
        lcd.setCursor(0, 0);  
        lcd.print(F("Dryness:    "));  
        lcd.setCursor(0, 1);  
        lcd.print(F("  1. More    "));  
        lcd.setCursor(0, 2);  
        lcd.print(F("  2. Normal  "));  
        lcd.setCursor(0, 3);  
        lcd.print(F("  3. Less    "));  
        lcd.setCursor(5, ROW);  
        lcd.write((byte)0);  
        readInput();  
    }  
}
```

```
if (NEXT == 0) {
    lcd.setCursor(5, ROW);
    lcd.print(" ");
    ROW++;
    if (ROW == 4) {
        ROW = 1;
    }
    lcd.setCursor(5, ROW);
    lcd.write((byte)0);
}
else if (OK == 0) {
    stage++;
    if (ROW == 1) {
        humidity = 32.5;
    }
    else if (ROW == 2) {
        humidity = 37.5;
    }
    else {
        humidity = 42.5;
    }
    onOFF=!onOFF;
    work();
}
else {
    stage--;
}
}
}
```

```

void stage4(void) {
  if (stage == 4) {
    lcd.setCursor(0, 0);
    lcd.print(F("          "));
    lcd.setCursor(0, 1);
    lcd.print(F("  Drying  "));
    lcd.setCursor(0, 2);
    lcd.print(F("Temperature:  C"));
    lcd.setCursor(0, 3);
    lcd.print(F("Humidity:   %"));
    lcd.setCursor(13,3);
    lcd.print(humidityMeasure);
    lcd.setCursor(13,2);
    lcd.print(temperatureMeasure);
  }
}

```

```

void stage5(void) {
  if (stage == 5) {
    lcd.setCursor(0, 0);
    lcd.print(F("Temperature: 40-80"));
    lcd.setCursor(0, 1);
    lcd.print(F("          "));
    lcd.setCursor(0, 2);
    lcd.print(F("  Degrees Celsius  "));
    lcd.setCursor(0, 3);
    lcd.print(F("          "));
    lcd.setCursor(digitTemp, 1);

```

```
lcd.print(temperature_10);
lcd.setCursor(digitTemp+1, 1);
lcd.print(temperature_1);
lcd.setCursor(digitTemp, 2);
lcd.write((byte)1);
readInput();
if (NEXT == 0) {
    temperature_10++;
    if (temperature_10 == 9) {
        temperature_10 = 4;
    }
}
else if (OK == 0) {
    if (temperature_10 == 8) {
        stage=stage+2;
        temperature=10*temperature_10;
    }
    else {
        lcd.setCursor(digitTemp, 2);
        lcd.print(" ");
        digitTemp++;
        stage++;
    }
}
else {
    stage=1;
}
}
```

```

void stage6(void) {
    if (stage == 6) {
        lcd.setCursor(digitTemp, 1);
        lcd.print(temperature_1);
        lcd.setCursor(digitTemp, 2);
        lcd.write((byte)1);
        readInput();
        if (NEXT == 0) {
            temperature_1++;
            if (temperature_1 == 10) {
                temperature_1 = 0;
            }
        }
        else if (OK == 0) {
            stage++;
            temperature=10*temperature_10+temperature_1;
        }
        else {
            stage--;
            digitTemp--;
            temperature_1=0;
        }
    }
}

```

```

void stage7(void) {
    if (stage == 7) {
        lcd.setCursor(0, 0);

```



```

lcd.print(F("Time:  1-200 min"));
lcd.setCursor(0, 1);
lcd.print(F("          "));
lcd.setCursor(0, 2);
lcd.print(F("  minutes  "));
lcd.setCursor(0, 3);
lcd.print(F("          "));
lcd.setCursor(digitTime, 2);
lcd.print(time_100);
lcd.setCursor(digitTime+1, 2);
lcd.print(time_10);
lcd.setCursor(digitTime+2, 2);
lcd.print(time_1);
lcd.setCursor(digitTime, 3);
lcd.write((byte)1);
readInput();
if (NEXT == 0) {
  time_100++;
  if (time_100 == 3) {
    time_100 = 0;
  }
}
else if (OK == 0) {
  if (time_100 == 2) {
    stage=stage+3;
    Time=100*time_100;
  }
  else {
    if(time_100==0){

```

```
    lcd.setCursor(digitTime, 2);  
    lcd.print(" ");  
}  
lcd.setCursor(digitTime, 3);  
lcd.print(" ");  
digitTime++;  
stage++;  
}  
}  
else {  
    stage=stage-2;  
    digitTemp=9;  
    temperature_1=0;  
}  
}  
}
```

```
void stage8(void) {  
    if (stage == 8) {  
        lcd.setCursor(digitTime, 2);  
        lcd.print(time_10);  
        lcd.setCursor(digitTime, 3);  
        lcd.write((byte)1);  
        readInput();  
        if (NEXT == 0) {  
            time_10++;  
            if (time_10 == 10) {  
                time_10 = 0;  
            }  
        }  
    }  
}
```

```
}  
else if (OK == 0) {  
    if(time_10==0 && time_100==0){  
        lcd.setCursor(digitTime, 2);  
        lcd.print(" ");  
    }  
    lcd.setCursor(digitTime, 3);  
    lcd.print(" ");  
    digitTime++;  
    stage++;  
}  
else {  
    stage--;  
    time_10=0;  
    digitTime=5;  
}  
}  
}
```

```
void stage9(void) {  
    if (stage == 9) {  
        lcd.setCursor(digitTime, 2);  
        lcd.print(time_1);  
        lcd.setCursor(digitTime, 3);  
        lcd.write((byte)1);  
        readInput();  
        if (NEXT == 0) {  
            time_1++;  
            if (time_1 == 10) {
```



```
lcd.print(F(" 3. Less "));
lcd.setCursor(5, ROW);
lcd.write((byte)0);
readInput();
if (NEXT == 0) {
  lcd.setCursor(5, ROW);
  lcd.print(" ");
  ROW++;
  if (ROW == 4) {
    ROW = 1;
  }
  lcd.setCursor(5, ROW);
  lcd.write((byte)0);
}
else if (OK == 0) {
  stage++;
  if (ROW == 1) {
    humidity = 32.5;
  }
  else if (ROW == 2) {
    humidity = 37.5;
  }
  else {
    humidity = 42.5;
  }
  start=millis();
  Serial.println(start);
  onOFF=!onOFF;
  work();
}
```

```
}  
else {  
    stage=stage-3;  
    digitTime=5;  
}  
}  
}
```

```
void stage11(void) {  
    if (stage == 11) {  
        lcd.setCursor(0, 0);  
        lcd.print(F(" Time Remain "));  
        lcd.setCursor(0, 1);  
        lcd.print(F(" minutes "));  
        lcd.setCursor(0, 2);  
        lcd.print(F("Temperature: C"));  
        lcd.setCursor(0, 3);  
        lcd.print(F("Humidity: %"));  
        lcd.setCursor(5, 1);  
        lcd.print(Time-((millis()-start)/unitTranster/60));  
        lcd.setCursor(13,3);  
        lcd.print(humidityMeasure);  
        lcd.setCursor(13,2);  
        lcd.print(temperatureMeasure);  
    }  
}
```

```
void stage20(void) {  
    if (stage == 20) {
```

```

lcd.setCursor(0, 0);
lcd.print(F("          "));
lcd.setCursor(0, 1);
lcd.print(F("  TIME SHOULD  "));
lcd.setCursor(0, 2);
lcd.print(F("  NOT BE ZERO!  "));
lcd.setCursor(0, 3);
lcd.print(F("          "));
for (int i = 0; i < 3; i++) {
  tone(buzzerPin, 200);
  delay(100);
  noTone(buzzerPin);
  delay(100);
}
digitTime=5;
stage=7;
}
}

void stopCheck (void) {
  NEXT = digitalRead(buttonNEXT);
  OK = digitalRead(buttonOK);
  BACK = digitalRead(buttonBACK);
  while (digitalRead(buttonNEXT) == 0 || digitalRead(buttonOK) == 0 || digitalRead(buttonBACK) == 0) {
  }
  if (OK == 0){
    lcd.setCursor(0, 0);
    lcd.print(F("  Stopped  "));
    lcd.setCursor(0, 1);

```

```

lcd.print(F("Press OK to continue"));
lcd.setCursor(0, 2);
lcd.print(F("Press BACK to reset "));
lcd.setCursor(0, 3);
lcd.print(F("          "));
stopWork();
do {
  OK = digitalRead(buttonOK);
  BACK = digitalRead(buttonBACK);
} while (OK == 1 && BACK == 1);

while (digitalRead(buttonOK) == 0 || digitalRead(buttonBACK) == 0) {}
if (OK==0){
  work();
  start=start+(millis()-stop);
}
else if (BACK==0){
  delay(100);
  digitalWrite(pinReset, LOW);
}
}
}

```

```

void doorCheck(void){
  if (digitalRead(door) == 1){
    stopWork();
    lcd.setCursor(0, 0);
    lcd.print(F("  Dangerous!  "));
    lcd.setCursor(0, 1);

```



```

lcd.print(F(" Close the door! "));
lcd.setCursor(0, 2);
lcd.print(F("Press BACK to reset "));
lcd.setCursor(0, 3);
lcd.print(F("          "));
tone(buzzerPin, 200);
while (digitalRead(door)==1){
  if (digitalRead(buttonBACK)==0){
    noTone(buzzerPin);
    delay(100);
    digitalWrite(pinReset, LOW);
  }
  delay(10);
}
noTone(buzzerPin);
work();
start=start+(millis()-stop);
}
}

void ender (void){
  time2=(millis()-start)/60000;
  Serial.print("time2: ");
  Serial.println(time2);
  timeRemain=(time2-Time);
  Serial.print("timeRemain: ");
  Serial.println(timeRemain);
  if (time2 >= 1){
    if (timeRemain <=0 || humidityMeasure<=humidity){

```

```

stopWork();

lcd.setCursor(0, 0);

lcd.print(F("      "));

lcd.setCursor(0, 1);

lcd.print(F("  Finished  "));

lcd.setCursor(0, 2);

lcd.print(F("      "));

lcd.setCursor(0, 3);

lcd.print(F("      "));

for(int xx=3; xx>=0; xx--){
for (int i = 0; i < sizeof(MELODY) / sizeof(MELODY[0]); i++) {
  if (MELODY[i] == 0) {
    delay(noteDuration);
  }
  else {
    tone(buzzerPin, MELODY[i], noteDuration);
    delay(noteDuration);
  }
  noTone(buzzerPin);
  delay(20);
}
for (int i = 0; i < sizeof(melody) / sizeof(melody[0]); i++) {
  if (melody[i] == 0) {
    delay(noteDuration);
  }
  else {
    tone(buzzerPin, melody[i], noteDuration);
    delay(noteDuration);
  }
}

```

```
    }  
    noTone(buzzerPin);  
    delay(20);  
  }  
  
  delay(200);  
  
  digitalWrite(pinReset, LOW);  
}  
}  
}  
  
void work (void){  
  
  for (int fadeValue = 0 ; fadeValue <= 200; fadeValue += 5) {  
    analogWrite(motor, fadeValue);  
    delay(15);  
  }  
  
  for (int fadeValue = 0 ; fadeValue <= 127; fadeValue += 5) {  
    analogWrite(fan, fadeValue);  
    delay(15);  
  }  
  
  digitalWrite(heater, HIGH);  
  
}
```

```
void stopWork (void){

    digitalWrite(heater, LOW);
    delay (100);
    for (int fadeValue = 200; fadeValue >= 0; fadeValue -= 5) {
        analogWrite(motor, fadeValue);
        delay(10);
    }
    stop=millis();

    for (int fadeValue = 127 ; fadeValue >= 0; fadeValue -= 5) {
        analogWrite(fan, fadeValue);
        delay(10);
    }

}

void TemHumTest (void){

    sensorSTOP = millis();

    if (sensorSTOP-sensorSTART > 1000){
        humidityMeasure = dht22.getHumidity();
        temperatureMeasure = dht22.getTemperature();

        if (temperatureMeasure != -1 && humidityMeasure != -1){

            Serial.print("Temperature: ");
            Serial.print(temperatureMeasure);
```

```
Serial.println(" C");
```

```
Serial.print("Humidity: ");
```

```
Serial.print(humidityMeasure);
```

```
Serial.println(" %");
```

```
}
```

```
else{
```

```
Serial.println("Error reading HUM & TEM data");
```

```
}
```

```
sensorSTART = millis();
```

```
}
```

```
}
```

```
void controlTemp(){
```

```
if ((temperatureMeasure - 5) > temperature){
```

```
digitalWrite(heater, LOW);
```

```
}
```

```
else if ((temperatureMeasure + 5) < temperature){
```

```
digitalWrite(heater,HIGH);
```

```
Serial.println("temp con");
```

```
}
```

```
}
```