

# 똑똑한 환풍기

멀티미디어학과

2014111608

라현아

# 똑똑한 환풍기

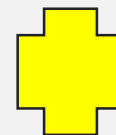
멀티미디어학과  
2014111608  
라현아

## ▣ 연구목적 및 배경

욕실에서 샤워를 하고 난 후 습기와 열기가 제대로 빠져나가지 못해 물이 마르지 않고 수건에서 냄새가 나는 것에 불편함을 느껴 연구를 진행하게 되었다. 기존의 환풍기는 ON/OFF가 수동이라 종종 환풍기를 켜두고 나가 불필요한 전력을 낭비하고 과열의 위험이 있다. 그래서 자동으로 화장실 환경을 인식해 작동하는 환풍기가 없을까 생각해 보다가 똑똑한 환풍기를 생각하게 되었다. 화장실 및 샤워실의 쾌적함을 편리하게 유지하는 것에 연구 목적이 있다.

## ▣ 사용한 부품

아두이노 우노  
온습도 센서 DHT11  
LCD  
RED, GREEN LED  
모터  
팬  
버튼 4개  
9V전원 배터리



## ▣ 연구내용

아두이노 우노에 온습도 센서를 사용하여 현재온도, 습도를 측정하여 LCD에 출력하고 희망 온도와 습도값은 버튼4개를 사용하여 UP/DOWN을 가능하게 하며 그 값 역시 LCD에 출력한다. 현재 온습도값 보다 설정 온습도 값이 낮으면 RED LED가 점등하고 팬의 모터가 구동된다. 반대의 경우에는 팬의 모터가 작동을 멈추고 GREEN LED가 점등된다. 또한 배터리를 연결하여 전원을 공급해 컴퓨터 전원이 없어도 언제 어디서나 작동 가능하며 휴대와 이동이 가능하다.

## ▣ 기대효과

일반 화장실, 샤워실, 목욕탕 등등 환풍이 필요한 모든 곳에서 사용이 가능하고 희망 온도와 습도를 설정할 수 있어 환경에 따른 다른 기준을 부여할 수 있고 그에 따라 모터가 자동으로 ON/OFF 되기 때문에 불필요한 전력소비를 줄일 수 있다. 손쉽게 실내환경을 쾌적하게 유지할 수 있다.

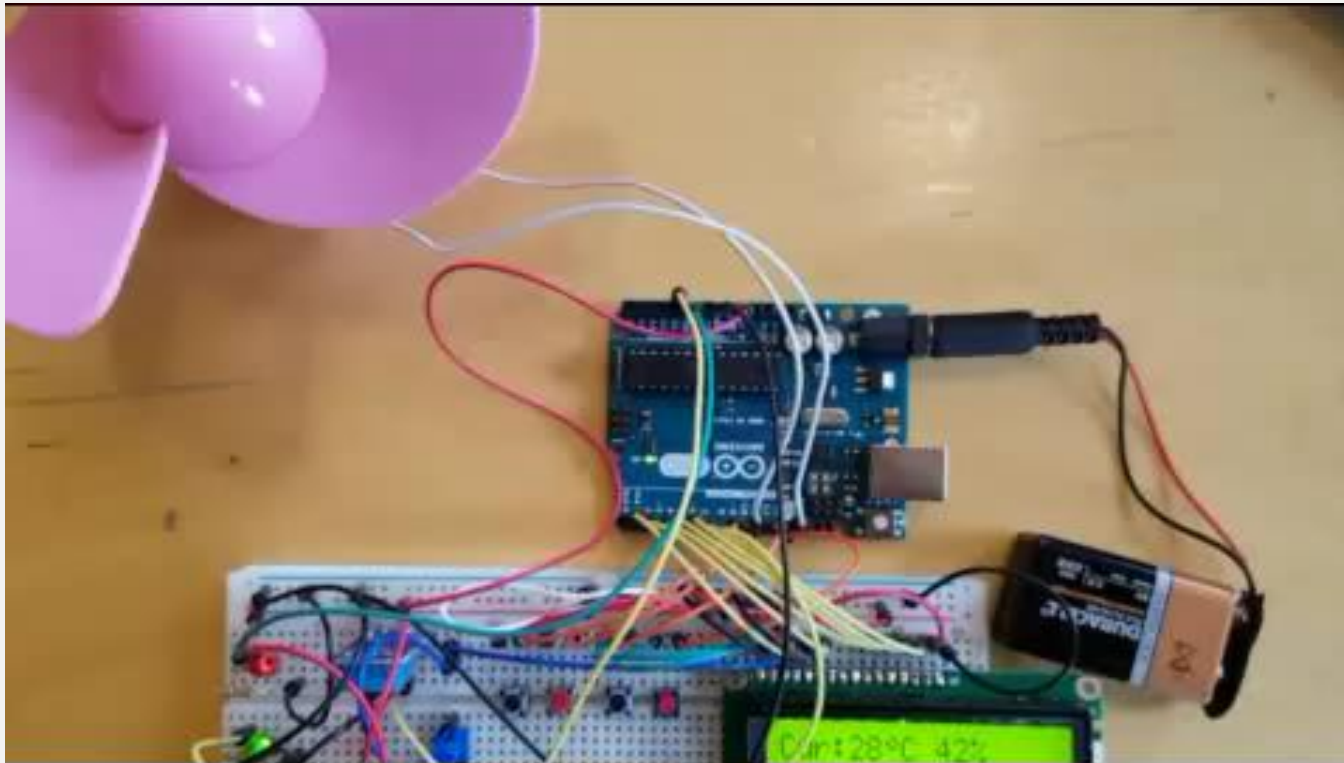
## ▣프로젝트 진행 결과

04.29 연구 주제 고민  
05.09 아이디어 고민  
05.11 최종주제 결정, 기획서작성  
05.16 필요한 부품 파악하기  
05.20 부품주문  
06.06 어떻게 만들지 구상  
06.08 온 습도 센서와 LCD연결  
06.11 버튼과 모터연결  
0.6.18 전원, LED연결  
06.20 전원, 모터 팬 연결  
06.21 전원연결, 최종마무리  
자세한 내용은 연구노트에 작성.

# 똑똑한 환풍기

멀티미디어학과  
2014111608  
라현아

▣시연 영상



## ■시연영상 설명

초기 희망 온도와 습도값을 28℃와 48%로 설정한다. 이 상태는 GREEN LED가 점등하고 모터가 작동하지 않는 상태이다. 모터를 구동해 보기 위해 버튼을 이용하여 희망 온도나 습도를 현재 온도나 습도보다 낮게 설정하여 본다. 현재온도나 습도가 희망 온도나 습도값 이상이 되면 LCD 오른쪽하단의 + 마크가 \*마크로 바뀌고 RED LED가 점등되고 모터가 작동하여 팬이 돌아간다. 아두이노 로의 전원 연결은 9V배터리를 연결하여 이동할 수 있게 한다.

# 똑똑한 환풍기

멀티미디어학과  
2014111608  
라현아

## 코드

```
//2016년 1학기 DIY소프트웨어 프로젝트 "똑똑한 환풍기"
#define TEMP_ADDR 72
//LCD라이브러리 삽입 및 객체 인스턴스 생성
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

//온도기호 정의
byte degree[8] = {
    B00110,
    B01001,
    B01001,
    B00110,
    B00000,
    B00000,
    B00000,
    B00000,
};

//팬 작동중 기호 정의
byte fan_on[8] = {
    B00100,
    B10101,
    B01110,
    B11111,
    B01110,
    B10101,
    B00100,
    B00000,
};

//팬 정지 기호 정의
byte fan_off[8] = {
    B00100,
    B00100,
    B00100,
    B11111,
    B00100,
    B00100,
    B00100,
    B00000,
};

//회로연결 핀 상수 정의
const int DOWN_BUTTON=9;
const int UP_BUTTON=10;
const int DOWN_H_BUTTON=12;
const int UP_H_BUTTON=13;
const int FAN=11;
#define A0 14 //아날로그 핀을 디지털로 사용
#define A1 15 //아날로그 핀을 디지털로 사용
const int RED=14;
const int GREEN=15;

//스위치 디바운싱에 사용할 변수
boolean lastDownTempButton=LOW;
boolean currentDownTempButton=LOW;
boolean lastUpTempButton=LOW;
boolean currentUpTempButton=LOW;

boolean lastDownHumButton=LOW;
boolean currentDownHumButton=LOW;
boolean lastUpHumButton=LOW;
boolean currentUpHumButton=LOW;

int set_temp=29; //희망 온도값 설정
int set_hum=48; //희망 습도값 설정
int DHpin = 8;
byte dat [5];
byte read_data () {
    byte data;
    for (int i = 0; i < 8; i ++ ) {
        if (digitalRead (DHpin) == LOW) {
            while (digitalRead (DHpin) == LOW); // wait for 50us
            delayMicroseconds (30); // determine the duration of the high level
            if (digitalRead (DHpin) == HIGH)
                data |= (1 << (7-i)); // high front and low in the post
            while (digitalRead (DHpin) == HIGH); // data '1 ', wait for the next
        }
    }
    return data;
}

void start_test () {
    digitalWrite (DHpin, LOW); // bus down, send start signal
    delay (30); // delay greater than 18ms, so DHT11 start signal can be received

    digitalWrite (DHpin, HIGH);
    delayMicroseconds (40); // Wait for DHT11 response

    pinMode (DHpin, INPUT);
    while (digitalRead (DHpin) == HIGH);
    delayMicroseconds (80); // DHT11 response, pulled the bus 80us
    if (digitalRead (DHpin) == LOW);
    delayMicroseconds (80); // DHT11 80us after the bus pulled to start signal

    for (int i = 0; i < 4; i ++ ) // receive temperature and humidity data
        dat[i] = read_data ();

    pinMode (DHpin, OUTPUT);
    digitalWrite (DHpin, HIGH); // send data once after releasing the bus
}
```

# 똑똑한 환풍기

멀티미디어학과  
2014111608  
라현아

```
void setup()
{
    pinMode(FAN, OUTPUT);
    pinMode(DHpin, OUTPUT);
    pinMode(RED, OUTPUT); //아날로그 핀을 디지털로 사용
    pinMode(GREEN, OUTPUT); //아날로그 핀을 디지털로 사용

    //LCD를 16행2열로 지정
    lcd.begin(16, 2);
    //사용자 정의문자 등록
    lcd.createChar(0, degree);
    lcd.createChar(1, fan_off);
    lcd.createChar(2, fan_on);

    //LCD에 고정 정보 출력
    lcd.setCursor(0,0);
    lcd.print("Cur:");
    lcd.setCursor(6,0);
    lcd.write((byte)0);
    lcd.setCursor(7,0);
    lcd.print("C");
    lcd.setCursor(11,0);
    lcd.print("%");
    lcd.setCursor(0,1);
    lcd.print("Set:");
    lcd.setCursor(6,1);
    lcd.write((byte)0);
    lcd.setCursor(7,1);
    lcd.print("C");
    lcd.setCursor(11,1);
    lcd.print("%");
    lcd.setCursor(15,1);
    lcd.write(1);
}
```

```
//버튼 두개에서 사용하는 스위치 디바운싱 함수
boolean debounce(boolean last, int pin)
{
    boolean current=digitalRead(pin);
    if(last!=current)
    {
        delay(5);
        current=digitalRead(pin);
    }
    return current;
}
```

```
void loop()
{
    start_test ();
    lcd.setCursor(4,0); //1열 5행으로 커서 이동
    lcd.print(dat [2], DEC); //섭씨온도 LCD에 기록
    lcd.setCursor(9,0); //1열 9행으로 커서 이동
    lcd.print(dat [0], DEC); //습도 LCD에 기록
    delay(1000);

    //버튼 눌림 상태를 디바운싱으로 감지
    currentDownTempButton = debounce(lastDownTempButton, DOWN_T);
    currentUpTempButton = debounce(lastUpTempButton, UP_BUTTON);

    //희망 온도 내림
    if(lastDownTempButton==LOW&&currentDownTempButton==HIGH)
    {
        set_temp--;
    }
    //희망 온도 올림
    else if(lastUpTempButton==LOW&&currentUpTempButton==HIGH)
    {
        set_temp++;
    }
}
```

```
//희망 온도 출력
lcd.setCursor(4,1);
lcd.print(set_temp);
//현재 버튼 상태 업데이트
lastDownTempButton=currentDownTempButton;
lastUpTempButton=currentUpTempButton;
////////////////////////////////////
//버튼 눌림 상태를 디바운싱으로 감지
currentDownHumButton = debounce(lastDownHumButton, DOWN_H_BUTTON);
currentUpHumButton = debounce(lastUpHumButton, UP_H_BUTTON);
//희망 습도 내림
if(lastDownHumButton==LOW&&currentDownHumButton==HIGH)
{
    set_Hum--;
}
//희망 습도 올림
else if(lastUpHumButton==LOW&&currentUpHumButton==HIGH)
{
    set_Hum++;
}
//희망 온도 출력
lcd.setCursor(9,1);
lcd.print(set_Hum);
//현재 버튼 상태 업데이트
lastDownHumButton=currentDownHumButton;
lastUpHumButton=currentUpHumButton;

//습도와 온도가 높으면
if(dat[0]>=set_Hum || dat[2]>=s
{
    //팬 회전시키고, RED LED 점등, LCD화면에 회전표시
    digitalWrite(RED, HIGH);
    digitalWrite(GREEN, LOW);
    digitalWrite(FAN, HIGH);
    lcd.setCursor(15,1);
    lcd.write(2);
}

//습도가 낮으면
else
{
    digitalWrite(FAN, LOW);
    digitalWrite(RED, LOW);
    digitalWrite(GREEN, HIGH);
    lcd.setCursor(15,1);
    lcd.write(1);
}
```



## ■연구노트 (따로제출함)

