
A study on smart shoes styler

스마트 슈즈 스타일러 개발 연구

김예슬 진우열 임재혁 장민혁



한국공학대학교
TECH UNIVERSITY OF KOREA

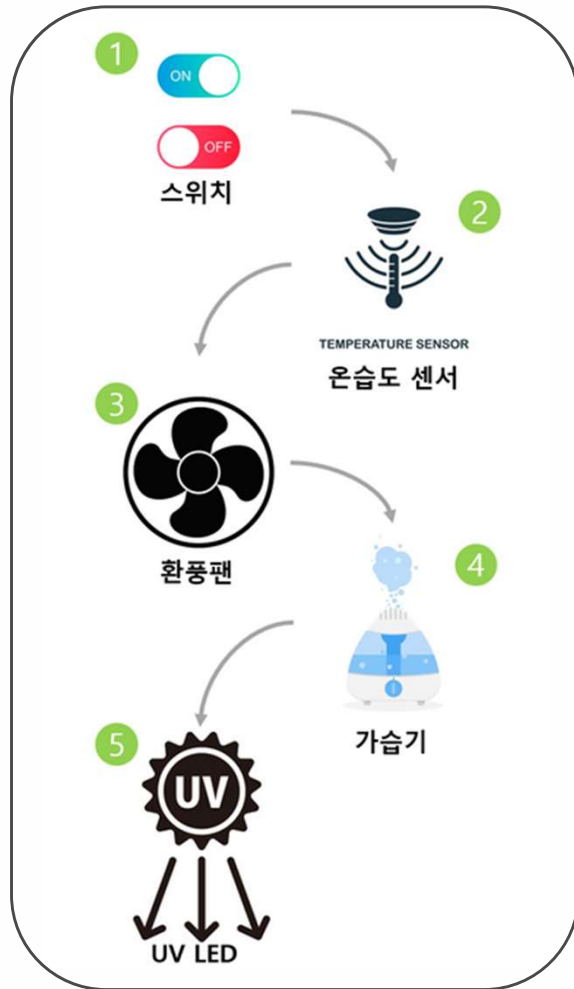
| 요구사항 정의서

구분	기능	설명
S/W	제품 내 적정 온도 조절	쿨링팬 동작으로 인해 조절할 온도를 변경할 수 있다.
	제품 내 적정 습도 조절	쿨링팬 동작으로 인해 조절할 습도를 변경할 수 있다
	내부 온도 조절	쿨링팬을 통해 제품 내의 온도를 높이고, 낮출 수 있다
	내부 습도 조절	쿨링팬을 통해 제품 내의 습도를 높이고, 낮출 수 있다
	작동 상태 확인	PIR센서를 통해 사용자를 감지한다.
	UV등 살균기능 제어	UV LED를 이용하여, 제품 내부의 살균 정도를 조절 할 수 있다.
	LCD를 통한 온습도 표기	온습도 센서를 통해 측정된 내부의 온도, 습도가 LCD에 표기되어 현상태를 알 수 있다

구분	기능	설명
H/W	제품 ON/OFF	토글 스위치를 이용하여 제품을 끄고 켤 수 있다.
	제품 내 온도 측정	온도 센서를 통해 제품 내의 온도를 측정할 수 있다.
	제품 내 온도 내려주기	제품에 달린 쿨링 팬을 사용하여 내부 온도를 낮춰 준다.
	제품 내 습도 측정	습도 센서를 통해 제품 내의 습도를 측정할 수 있다.
	제품 내 습도 내려주기	제품에 달린 쿨링 팬을 사용하여 내부 습도를 낮춰 준다.
	작동상태 LED 표시	PIR센서를 통해 사용자를 감지하여 LED로 기기의 작동 상태를 표시한다.
	신발 향 관리하기	가습기에 탈취제를 추가하여 탈취효과와 좋은 향이 남도록 해준다
	신발 살균하기	UV등을 이용하여 제품 내부의 신발을 살균한다

| 서비스 구성도 - 서비스 시나리오

| 서비스 구성도 - 서비스 시나리오



**[1] 사용자가 신발을 스타일러에
넣은 후에 스위치 조절**

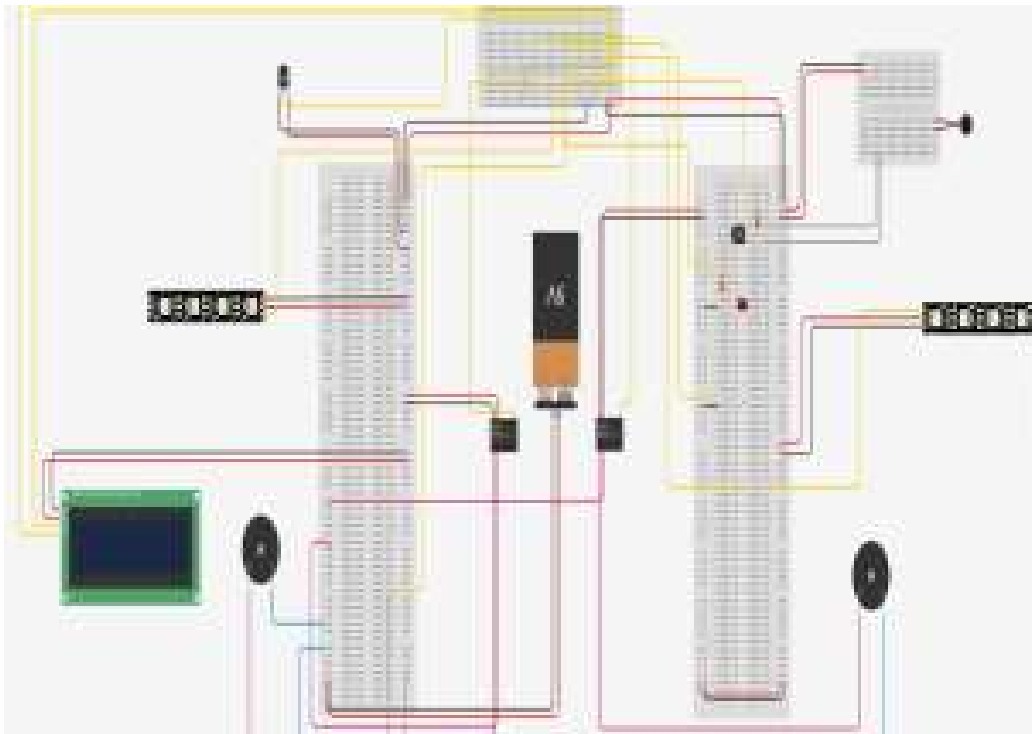
**[2] 온습도 센서가 스타일러 내의
습도를 측정**

**[3] 환풍팬이 설정된 습도까지,
작동하여 습도 조절**

**[4] 환풍팬의 작동이 끝나면, 가습기
모듈이 설정 된 시간만큼 작동**

**[5] 가습기 모듈의 작동이 끝나면,
UV 램프가 작동하여 살균**

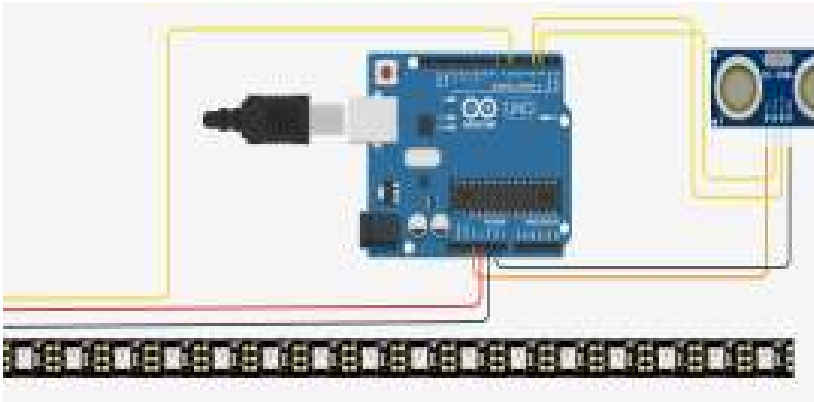
| 하드웨어/센서 구성도



보드-라즈베리파이4

센서 종류	연결 핀	설명	
UVC LED v2.0	GND	라즈베리 파이의 GND에 연결	
	VCC	라즈베리 파이의 5V에 연결	
	SIG	라즈베리 파이의 GPIO11에 연결	
Ultrasonic Humidfier Module	GND	라즈베리 파이의 GND에 연결	
	VCC	라즈베리 파이의 5V에 연결	
	JP1	트랜지스터	GPIO9에 연결
		트랜지스터	
DHT11	VCC	라즈베리 파이의 5V에 연결	
	GND	라즈베리 파이의 GND에 연결	
	DAT	라즈베리 파이의 GPIO22에 연결	
환풍 팬	GND	라즈베리 파이의 GND에 연결	
	VCC	트랜지스터 – GPIO4에 연결	

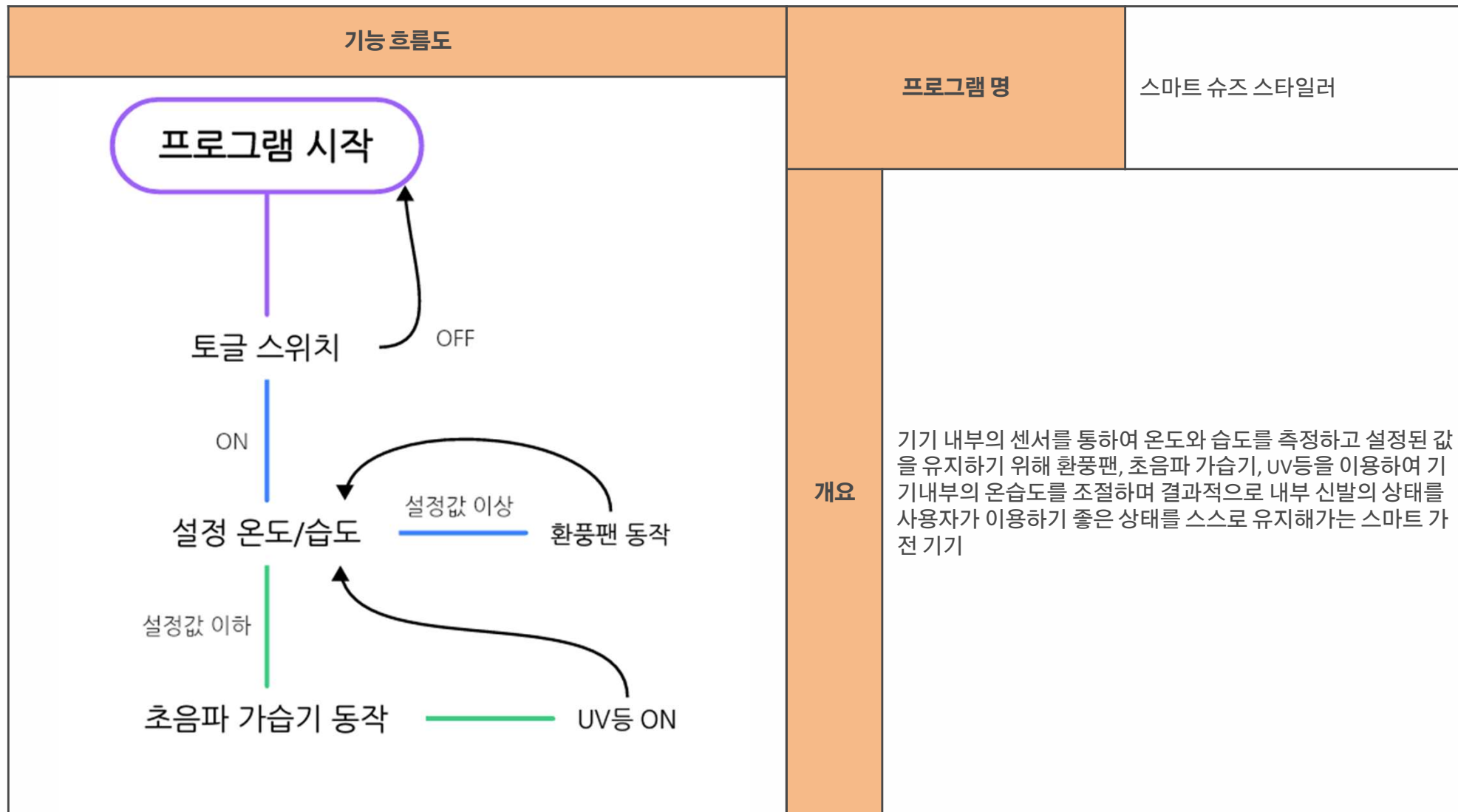
| 하드웨어/센서 구성도

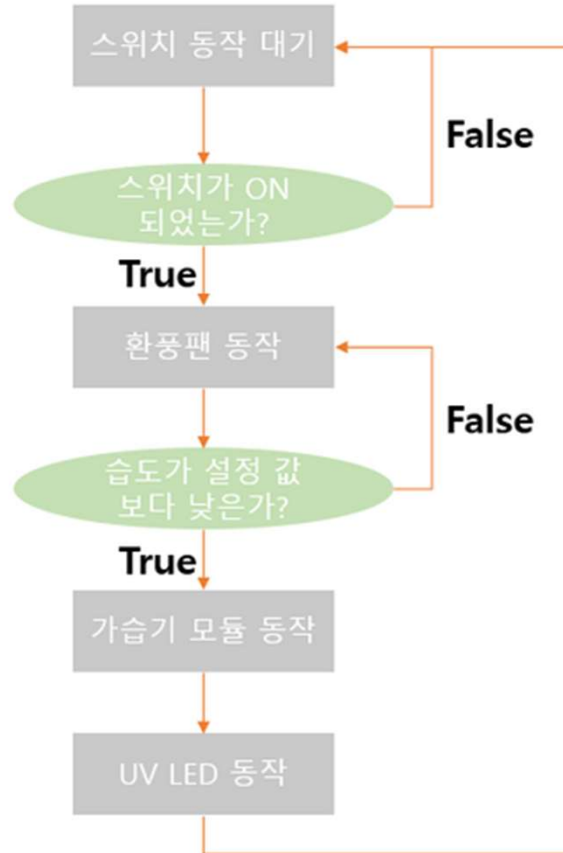


보드-아두이노 우노

센서 종류	연결핀	설명
초음파 센서	GND	아두이노 GND
	VCC	아두이노 5V
	TRIGGER	2번 PIN
	ECHO	3번 PIN
RGB LED	GND	아두이노 GND
	VCC	아두이노 5V
	SIG	6번 PIN
	COUNT	15번 PIN

| 기능 처리도(기능 흐름도)





1. 스위치 OFF 상태에서, 스타일러는 센서 감지로 인한 간접등 기능만을 동작한다
2. 신발을 넣고 스타일러의 스위치를 ON으로 변경하면, 신발 내의 센서가 습도를 측정한다
3. 측정한 습도가 설정 습도 이상일 시, 환풍팬이 동작하여, 제품 내의 습도를 낮춰준다
4. 측정한 습도가 설정 습도보다 이하일 시, 환풍팬 동작을 멈추고, 가습기 모듈을 작동한다
5. 일정 시간 가습기 모듈 동작 후에, UV LED가 동작하여 신발을 살균하여 준다.
6. 일정 시간 UV LED 동작 후에, UV LED가 OFF 되며, 다음 스위치 동작을 대기한다

[스타일러 내 습도 조절 알고리즘]

스타일러 내의 온습도 센서는, 내부의 습도를 계속 측정하여, 환풍팬의 동작을 제어한다.

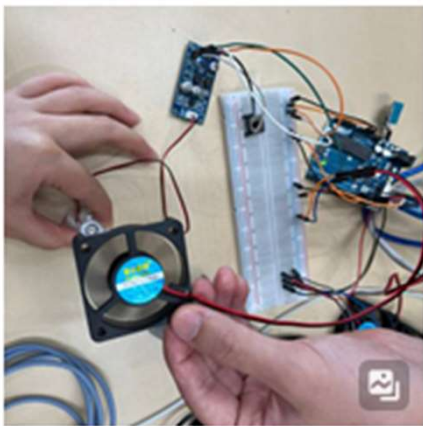
습도 > 85도면 15도 떨어 질때까지

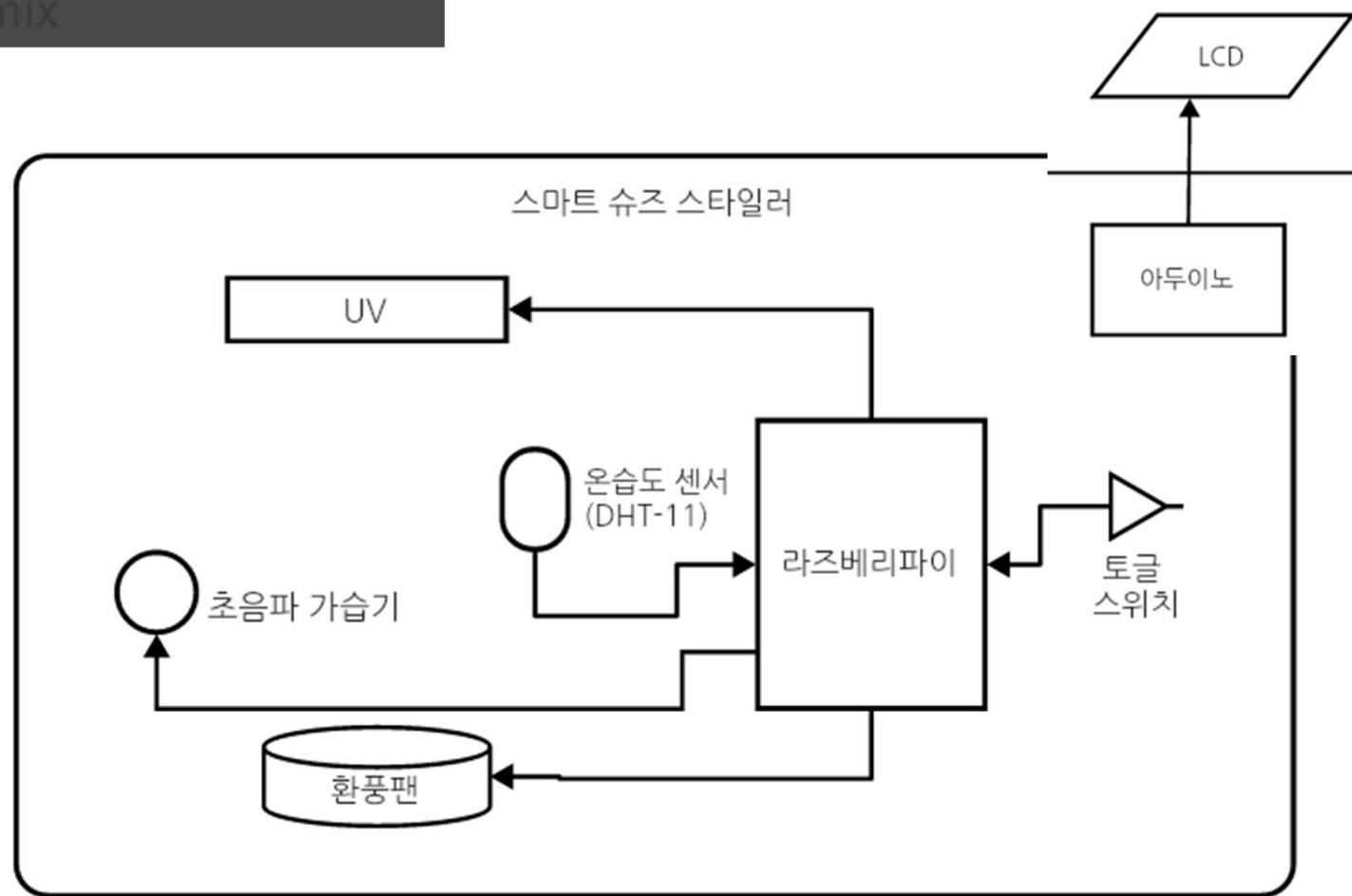
습도 > 80이면 12도 떨어 질때까지

습도 > 75이면 8도 떨어 질때까지

습도 > 70이면 5도 떨어 질때까지

그 이외에는 2도 떨어 질때 까지로 설정 후 다음 모듈로 넘어가도록 설계하였다.





| 프로그램 - 목록

기능 분류	지정 명	기능
기기	FAN	환풍팬의 동작여부를 결정
	GAS	초음파 가습기의 동작여부를 결정
	UV	UV등의 동작여부를 결정
	LED	문 개방시 조명불빛으로 보여줌
센서	온습도 센서	온습도 센서(DHT-11)의 값을 읽음
	초음파 센서	초음파 센서로 문이 열리면 led조명이 작동됨
함수	read_sensor	센서의 값을 불러오고 각 기기들의 동작여부를 판단
	turn_fan_on	환풍팬을 동작시킴
	turn_fan_off	환풍팬을 중지시킴
	control_fan	초음파 가습기의 동작 여부를 status(ON, OFF)로 받아 판단함

```
(i2c_expander, i2c_address, cols=lcd_columns, rows=lcd_rows)
:
ing("TEST HUMIDITY")
perature = Adafruit_DHT.read_retry(sensor, pin)
s not None and temperature is not None:
    P="{0:0.1f}"°C HUM="{1:0.1f}"%.format(temperature, humidity)
    r_pos = (0, 0)
    _string("TEST HUMIDITY")
    r_pos = (1, 0)
    _string('HUM={0:0.1f}%'.format(humidity))

ain')
ty

():
emperature_1 = Adafruit_DHT.read_retry(sensor, pin)
ty_1

:
AN_PIN, GPIO.HIGH)
AN_PIN1, GPIO.HIGH)
```

LCD 모듈 정의

환풍팬 모듈 ON

```
64 def turn_fan_off():
65     GPIO.output(FAN_PIN, GPIO.LOW)
66     GPIO.output(FAN_PIN1, GPIO.LOW)
67     # print("OFF")
68
69 def control_fan(status):
70     if status == 'on':
71         GPIO.output(GAS_pin, GPIO.HIGH)
72         print("환풍기가 켜졌습니다.")
73     elif status == 'off':
74         GPIO.output(GAS_pin, GPIO.LOW)
75         print("환풍기가 꺼졌습니다.")
76     else:
77         print("올바른 상태를 입력하세요.")
78
79 try:
80     GPIO.output(GAS_pin, GPIO.LOW)
81     GPIO.output(UV_pin, GPIO.LOW)
82     GPIO.output(UV_pin1, GPIO.LOW)
83     lcd.clear()
84     lcd.write_string("START")
85     time.sleep(1)
86
87 while True:
88     if GPIO.input(BUTTON_PIN) == False:
89         GPIO.output(LED_PIN, GPIO.HIGH)
90         read_sensor_1()
91         humidity_1 = read_sensor_1()
92         read_sensor()
93         humidity = read_sensor()
94
```

환풍팬 모듈OFF

습도에 따라 환풍팬 제어

```
95 time.sleep(1)
96
97 if humidity_1 >= 85:
98     MODE = 15
99 elif humidity_1 >= 80:
100     MODE = 12
101 elif humidity_1 >= 75:
102     MODE = 8
103 elif humidity_1 >= 70:
104     MODE = 5
105 else:
106     MODE = 2
107
108 while humidity_1 - humidity < MODE:
109     turn_fan_on()
110     print("FAN ON")
111     read_sensor()
112     humidity = read_sensor()
113     time.sleep(1)
114     #print("OFF")
115 if humidity_1 - humidity >= MODE:
116     turn_fan_off()
117     print("FAN OFF")
118     lcd.clear()
119     lcd.cursor_pos = (0, 0)
120     lcd.write_string("HUMIDITER ON")
121     control_fan('on')
122     TIME = 20
123 for sec in range(TIME + 1):
124     lcd.cursor_pos = (0, 0)
125     lcd.write_string("HUMIDITER")
```

온습도센서 수치 값으로 제어

```
126         lcd.cursor_pos = (1, 0)
127         lcd.write_string(f"Remained Time:{TIME}s")
128         #lcd.write_string('HUM={0:0.2f}s'.format(TIME))
129         TIME -= 1
130         time.sleep(1)
131         lcd.clear()
132         #time.sleep(10)
133         control_fan('off')
134         lcd.write_string("HUMIDITER OFF")
135         time.sleep(1)
136         lcd.clear()
137         lcd.write_string("UV LIGHT ON")
138         GPIO.output(UV_pin, GPIO.HIGH)
139         GPIO.output(UV_pin1, GPIO.HIGH)
140         print('UV TURN ON')
141         TIME_UV = 5
142         for sec in range(TIME_UV + 1):
143             lcd.cursor_pos = (0, 0)
144             lcd.write_string("UV LIGHT")
145             lcd.cursor_pos = (1, 0)
146             lcd.write_string(f"Remained Time:{TIME_UV}s")
147             #lcd.write_string('HUM={0:0.2f}s'.format(TIME))
148             TIME_UV -= 1
149             time.sleep(1)
150             lcd.clear()
151         lcd.write_string("UV LIGHT OFF")
152         GPIO.output(UV_pin, GPIO.LOW)
153         GPIO.output(UV_pin1, GPIO.LOW)
154         print('UV TURN OFF')
155         time.sleep(5)
156         lcd.clear()
157         GPIO.output(BUZ_pin, GPIO.HIGH)
158         time.sleep(1)
159         GPIO.output(BUZ_pin, GPIO.LOW)
160         lcd.write_string("Ready for next!")
161         time.sleep(5)
162
163
164     else:
165         GPIO.output(LED_PIN, GPIO.LOW)
166         time.sleep(1)
167
168 except KeyboardInterrupt:
169     GPIO.cleanup()
170     print("END")
```

신발의 냄새 제거 및 UV살균
동작

| 핵심소스코드-아두이노

```
#include <Adafruit_NeoPixel.h>

#define LED_PIN    6
#define LED_COUNT  15 // 10개의 LED를 사용

Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

int triggerPin = 2; // 초음파 센서의 트리거 핀
int echoPin = 3;   // 초음파 센서의 에코 핀

void setup() {
  strip.begin();
  strip.show();
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600); // 시리얼 통신 초기화
}

void loop() {
  // 초음파 센서로부터 거리 측정
  long duration;
  int distance;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
}
```

초음파 센서

```
32 // 거리가 5 이하인 경우 모든 LED를 끄고, 그 이상인 경우 무지개 색상으로 순환
33 if (distance <= 10) {
34   for (int i = 0; i < LED_COUNT; i++) {
35     strip.setPixelColor(i, strip.Color(0, 0, 0)); // 모든 LED 끄기
36   }
37 } else {
38   for (int i = 0; i < LED_COUNT; i++) {
39     // 거리가 5 초과이면 무지개 색상으로 순환
40     strip.setPixelColor(i, rainbowColor(i, millis()));
41   }
42 }
43 strip.show();
44 }
45
46 uint32_t rainbowColor(int led, unsigned long currentTime) {
47   int numLeds = strip.numPixels();
48   int step = 256 / numLeds;
49   int hue = (led * step + (currentTime / 1000)) % 256;
50   return strip.Color(
51     Wheel((hue + 0) & 255),
52     Wheel((hue + 85) & 255),
53     Wheel((hue + 170) & 255)
54   );
55 }
56
57 uint32_t Wheel(byte WheelPos) {
58   if (WheelPos < 85) {
59     return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
60   } else if (WheelPos < 170) {
61     WheelPos -= 85;
62     return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
63   } else {
64     WheelPos -= 170;
65     return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
66   }
67 }
68 }
```

초음파센서가 감지하면 LED 작동

| 참조- 개발 환경 및 설명

구분		항목	적용내역
S/W 개발환경	라즈베리파이 개발	Putty	라즈베리파이 내부에 접속하여 소스코드를 작성하고 작성된 코드를 따라 센서를 통해 정보를 전달받고 기기의 동작을 제어하도록 프로그래밍함
		VNC viewer	
	아두이노 개발	Arduino IDE	아두이노 소스코드를 작성하고 코드를 따라 정보 전달, 기기의 동작 제어 프로그래밍
H/W 구성장비	제어	라즈베리파이	센서의 정보를 전달받고 기기들의 동작을 제어함
	센서	온습도 센서(DHT-11)	스타일러 내부의 온도와 습도를 측정하여 정보를 제공함
		초음파 센서	초음파 센서를 문쪽에 부착하여 스타일러 문을 개방하면 조명이 켜지도록 설정
	기기	UV등	스타일러 내부의 살균을 위한 살균등
		초음파 가습기	스타일러 내부의 습도 유지 및 산뜻한 향을 추가함
		환풍팬	스타일러 내부의 온도 유지 및 환기기능
		LCD	스타일러 내부의 온습도를 표기해줌

| S/W 기능 실사 사진

라즈베리파이

```
Serial.print("습도: ");
Serial.print(humidity);
Serial.print("%, 온도: ");
Serial.print(temperature);
Serial.println("C");

if (temperature > temperatureThreshold) {
    // 온도가 임계값을 초과하면 쿨링 팬을 작동
    digitalWrite(fanPin, HIGH);
    Serial.println("쿨링 팬 작동 중");
} else {
    // 온도가 임계값 이하이면 쿨링 팬을 중지
    digitalWrite(fanPin, LOW);
    Serial.println("쿨링 팬 중지");
```

온/습도 측정하여
일정 수치에 도달하면
환풍팬 동작

```
import RPi.GPIO as GPIO
import time

FAN_PIN = 14

GPIO.setmode(GPIO.BCM)

GPIO.setup(FAN_PIN, GPIO.OUT)

try:
    while True:
        GPIO.output(FAN_PIN, GPIO.HIGH)
        print("ON")
        time.sleep(5)

        GPIO.output(FAN_PIN, GPIO.LOW)
        print("OFF")
        time.sleep(5)

except KeyboardInterrupt:
    GPIO.cleanup()
    print("MAIN OFF")
```

특정 조건에서
가습기 모듈 작동

```
2c_expander, i2c_address, cols=lcd_columns, rows=lcd_rows)

g("TEST HUMIDITY")
rature = Adafruit_DHT.read_retry(sensor, pin)
not None and temperature is not None:
    ={0:0.1f}"C HUM={1:0.1f}%'.format(temperature, humidity))

pos = (0, 0)
tring("TEST HUMIDITY")
pos = (1, 0)
tring('HUM={0:0.1f}%'.format(humidity))

n')

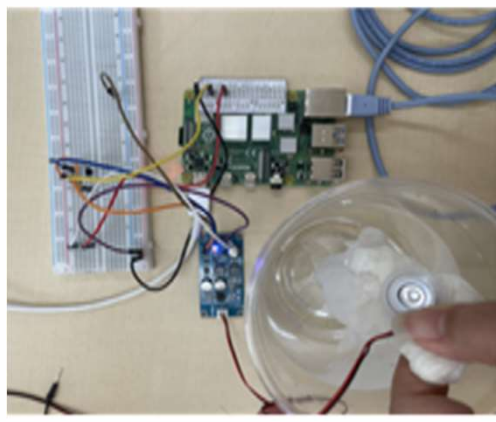
:
perature_1 = Adafruit_DHT.read_retry(sensor, pin)

_1

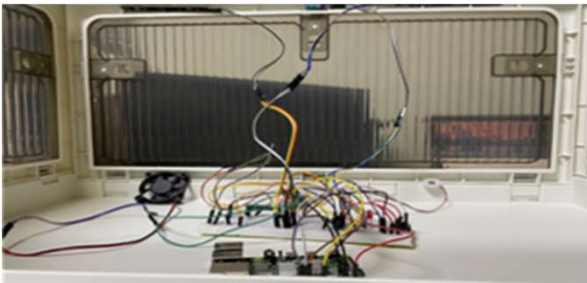
PIN, GPIO.HIGH)
PIN1, GPIO.HIGH)
```

측정된 온/습도를
표기해주는
LCD 모듈

| H/W 기능 실사사진

구분		항목	적용내역	
H/W 구성장비	센서	온습도 센서(DHT-11) & 환풍팬		
	기기	UV등 (좌)		
		초음파 가습기 (우)		
		LCD		

스타일러 부품 배치



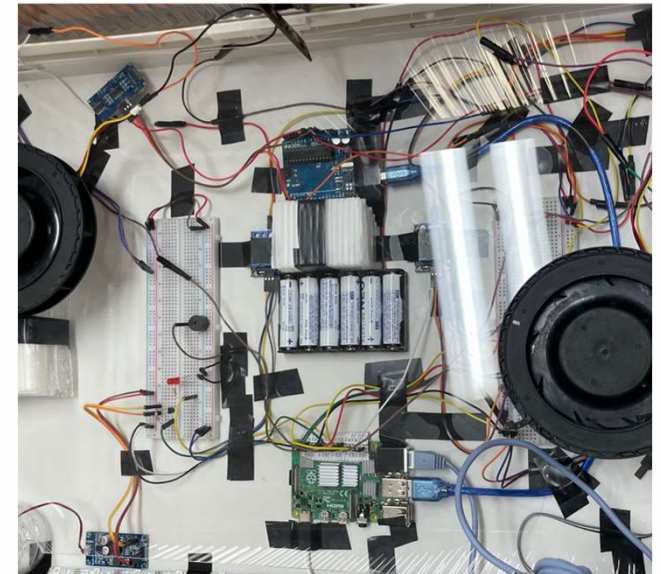
스타일러 내부에 신발을
넣고
스위치가 켜진다.

작동확인



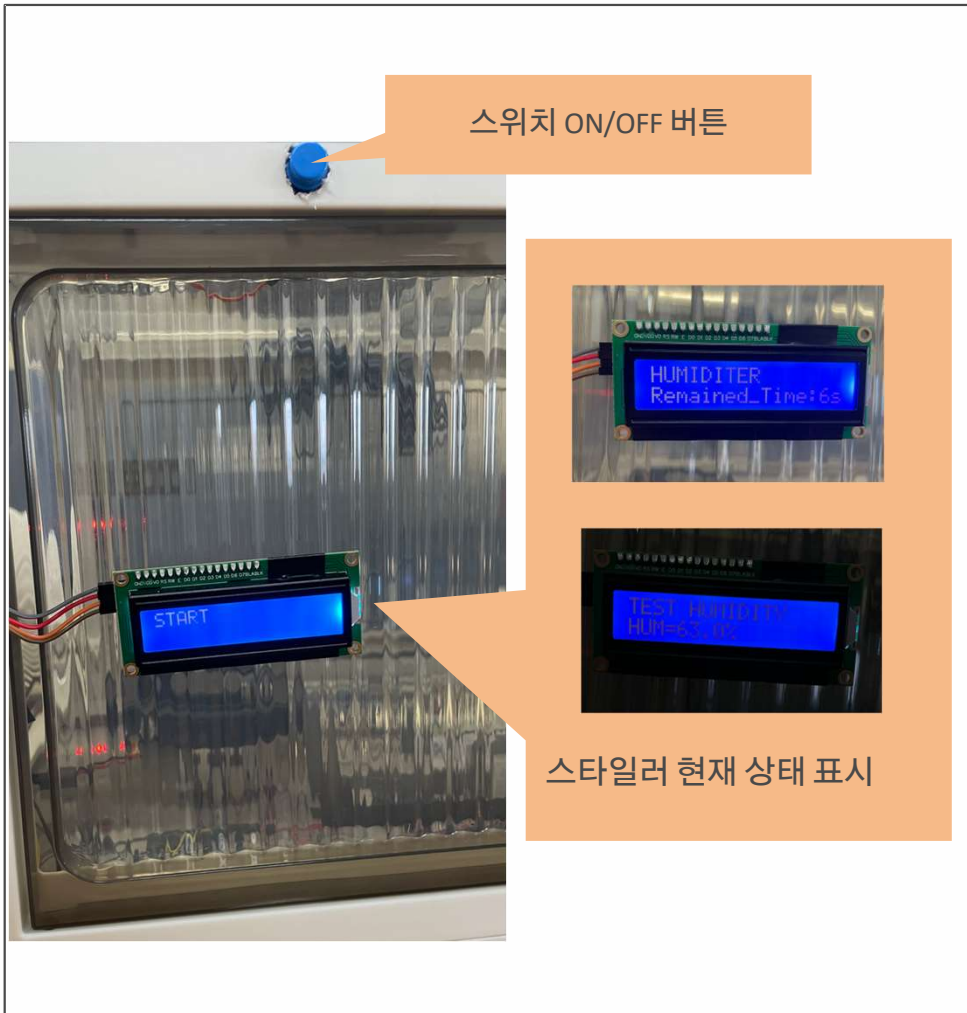
신발의 상태를
온/습도 센서로 확인하고
설정 값과 비교한다.

모듈병합 확인

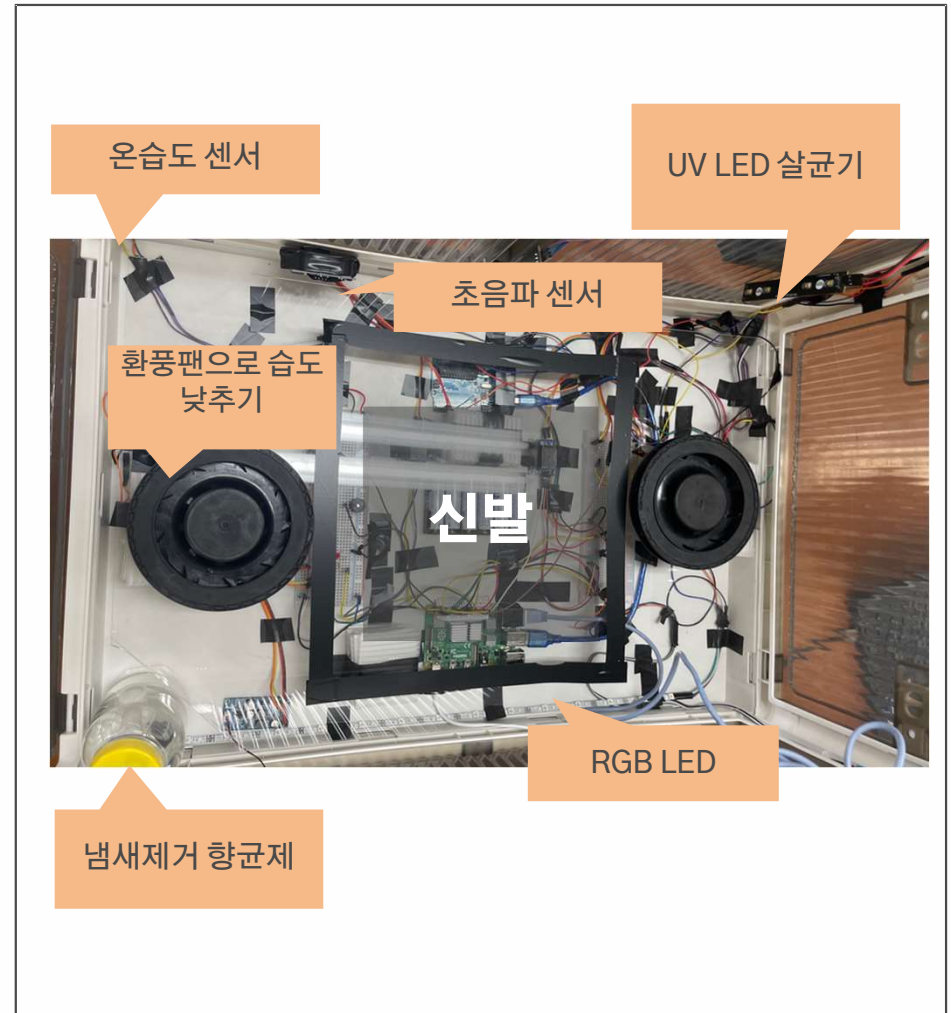


설정된 온습도를
유지하기 위해
조건에 따라 기기들을
동작시킨다.

스타일러 외부



스타일러 내부



| 시뮬레이션 영상



링크 : <https://www.youtube.com/watch?v=LEXf7wRyFPo>

Thank you