

# 스마트 홈 케어 시스템

## (Smart Home Care System)

본 보고서를 Project 프로젝트의 최종보고서로 제출합니다.

학과(전공): 컴퓨터소프트웨어공학

담당 교수: 김성우 교수님

팀 명: 5조

제 출 일: 2024년 6월 12일

팀 장 : 김민곤 20193174

팀원1 : 남진우 20193189

팀원2 : 오승헌 20193183

팀원3 : 김소희 20203164

동 의 대 학 교

컴퓨터소프트웨어공학과 학과장 귀하

## 〈 요약 문 〉

|                     |  |       |    |
|---------------------|--|-------|----|
| 프로젝트 목표<br>(300자내외) | 본 프로젝트는 라즈베리파이와 아두이노 키트를 활용하여 스마트 홈 케어 시스템을 개발하는 것을 목표로 한다. 이 시스템은 온습도 센서와 모터를 이용해 실내 환경을 자동으로 조절하고, 리모컨을 통해 조명 및 출입구를 제어할 수 있으며, 카메라를 통한 홈 CCTV 기능으로 집안의 안전을 강화하는 등 여러 가지 기능을 제공한다. 또한, 웹 서버를 구축하여 언제 어디서나 집안의 상태를 모니터링하고 제어할 수 있는 편리성을 제공한다. 이를 통해 사용자가 더욱 편리하고 안전한 생활 환경을 누릴 수 있도록 한다.  |       |    |
| 내용<br>(500자내외)      | <p>본 프로젝트는 사물인터넷 기술을 기반으로 한 스마트 홈 케어 시스템을 개발하는 것을 목적으로 한다. 이 시스템은 라즈베리파이를 중심으로 아두이노와 다양한 센서, 기기를 결합하여 실내 환경을 자동으로 조절하고, 사용자의 생활 편의성을 높이는 다양한 기능을 제공한다. 주요 기능으로는 온습도 조절, 조명 및 출입구 제어, 홈 CCTV 운영 등이 있으며, 이 모든 기능은 웹 서버를 통해 원격으로 제어 및 모니터링할 수 있다.</p> <p>본 시스템은 사용자가 언제 어디서나 집안의 상태를 확인하고 제어할 수 있도록 설계되어, 보다 안전하고 편리한 생활 환경을 제공한다. 프로젝트는 소규모 시뮬레이션을 통해 구현단계에서부터 실제 환경 적용 가능성을 탐색하며, 향후 확장 방안도 고려할 것이다. 이 과정에서 기술적 문제 해결 능력뿐만 아니라, IoT 기기의 작동 원리 및 통합 방법에 대한 이해도를 높일 수 있다.</p> |       |    |
| 기대효과<br>(200자내외)    | <p><b>에너지 절약을 통한 환경 보호</b></p> <p>본 프로젝트의 결과물이 대다수의 가정에 보급될 경우 원격 가전 제어를 통해 전력 낭비를 막는 것으로 에너지 절약을 할 수 있고, 이로 인한 환경 보전까지 기대해 볼 수 있다.</p> <p><b>치안 유지</b></p> <p>보편적인 보안 강화를 통해 무단 주거 침입, 강도 및 절도 등의 범죄를 미리 예방하고 범죄율을 낮출 수 있다. 이로 인한 치안 유지의 효과를 기대해 볼 수 있다.</p> <p><b>임베디드 시스템 발전</b></p> <p>본 프로젝트의 결과물이 널리 보급되어 프로젝트 결과물을 많은 사람들이 보고 임베디드 시스템이라는 분야에 관심을 가지게 하여 임베디드 시스템의 발전까지 기대해 볼 수 있다.</p>   |       |    |
| Keywords            | 사물 인터넷   | 홈 케어  | 보안 |
|                     | 모니터링   | 원격 제어 |    |

# 목 차

|                       |    |
|-----------------------|----|
| <b>I. 서 론</b>         | 4  |
| 1. 프로젝트 대상의 소개 및 설명   | 4  |
| 2. 프로젝트의 배경           | 4  |
| 3. 프로젝트의 필요성          | 5  |
| <b>II. 프로젝트 수행 내용</b> | 5  |
| 1. 프로젝트 목표            | 5  |
| 1.1 공학적 목표            | 5  |
| 1.2 비공학적 목표           | 5  |
| 2. 설계 구성 요소           | 6  |
| 3. 현실적 제한 조건          | 8  |
| 2.1 sub-section 제목    | 8  |
| 2.2 sub-section 제목    | 9  |
| 2.3 sub-section 제목    | 10 |
| <b>III. 결 론</b>       | 11 |
| <b>IV. 참고문헌</b>       | 12 |
| <b>[부록]</b>           | 13 |

## 1. 서론

### 1.1. 프로젝트 대상의 소개 및 설명

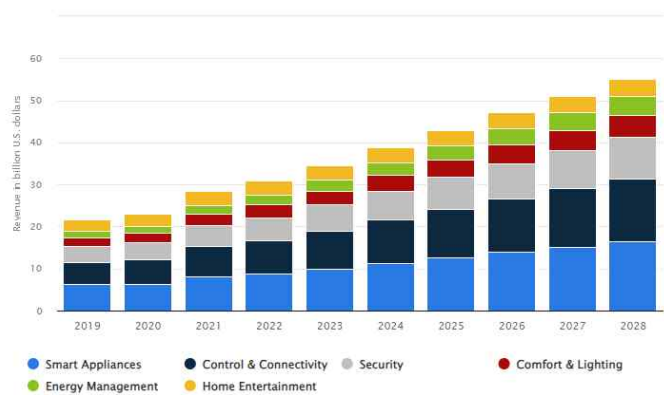
이 프로젝트는 스마트 홈 케어 시스템의 설계를 목표로 한다. 스마트 홈 케어 시스템은 다양한 센서와 기기를 통해 주거 환경을 보다 안전하고 편리하게 만드는 시스템이다. 이 시스템은 리모컨을 통한 조명 및 기기 제어, 자동 온습도 조절, 움직임 감지 센서 및 카메라를 이용한 홈 CCTV 기능 등을 제공한다. 이러한 기능들은 웹 서버와 클라우드 컴퓨팅을 이용하여 실시간으로 모니터링 및 제어가 가능하다. 이 프로젝트는 모든 사용자가 안전하고 편리한 생활을 할 수 있도록 설계되었다.

### 1.2. 프로젝트의 배경

무선 네트워크, 사물 인터넷, 인공지능, 클라우드 컴퓨팅 등의 과학 기술이 발전하면서 사용자가 네트워크나 컴퓨터를 의식하지 않고 장소에 상관없이 자유롭게 정보를 활용할 수 있는 환경, 즉 유비쿼터스 컴퓨팅 환경 실현을 향해 점점 다가가고 있다. 그 중에서도 고령화 사회로의 진입과 1인 가구의 증가로 스마트 헬스케어, 스마트홈이 화두가 되고 있다.



[그림 1] 스마트홈 연평균 성장률, ("따로 놓던 '스마트홈' 하나로...삼성-LG의 '동침'", 조선경제)



[그림 2] 미국 스마트홈 분야별 시장 규모, (Statista)

더불어 코로나 19의 영향으로 집에서 더 많은 시간을 보내게 되면서, 인테리어뿐만 아니라 사물인터넷, 가정형 인공지능 등의 홈 자동화 사업, 즉 스마트홈 시장에 대한 관심도와 투자 및 수요가 전 세계적으로 증가하고 있다.

국내의 미래창조과학부는 2017년에 홈-가전, 에너지, 헬스케어, 자동차, 제조 등 기존 산업과 IoT의 융합을 촉진하기 위해 총 989억원을 투자할 계획을 밝혔으며, 미국의 시장조사기관인 스탯이스타의 분석에 따르면 2023년 미국의 스마트홈 시장 규모는 약 346억 달러로 2028년에는 약 550억 달러에 이를 것으로 전망하고 있다.

또한 국내의 삼성과 LG, 외국의 아마존, 애플, 구글 등 전 세계적인 기업들이 협업을 통해 스마트홈 통신 표준규격, 즉 매터 구축 및 상용화를 추진하여 스마트홈 시장 확대에 나서고 있다.

### 1.3. 프로젝트의 필요성

스마트 홈 케어 시스템은 여러가지 센서와 기기들을 이용하여 리모컨을 통한 조명 및 기기 제어, 자동 온습도 조절, 움직임감지 센서 또는 카메라를 이용한 홈 CCTV 기능으로 주거지의 안전을 강화하는 등의 여러가지 기능을 웹 서버, 클라우드 컴퓨팅 등을 이용하여 실시간 모니터링 및 제어를 할 수 있는 편리성과 편의성, 쾌적한 주거 환경을 제공할 수 있다.

일반 사용자의 쾌적하고 편의성 높은 주거 환경 조성 뿐만 아니라 노인과 신체 장애인 등 사회적 소수자에게도 편의성과 안전한 주거 공간을 제공할 수 있을 것으로 예상된다. 2025년 초고령사회로의 진입이 예상됨에 따라 노인 인구의 안전과 건강관리에 대한 관심이 높아지고 있다. 이에 스마트 홈 케어 시스템으로 독거노인의 건강 모니터링, 응급상황 감지 및 대응 등을 통해 노인의 안전한 생활을 지원할 수 있다.

이 프로젝트를 통하여 일반 사용자의 편의와 안전을 제공할 수 있고, 더 나아가 사회적 소수자도 포용할 수 있는 포용적 공간, 포용적 도시 설계에도 도움이 될 것이다.

## 2. 프로젝트 수행 내용

### 2.1. 프로젝트 목표

#### 2.1.1. 공학적 목표

| 기능                 | 설명   | 사용 기술                |
|--------------------|--|----------------------|
| 홈 캠                | 웹 카메라를 통해 집안 내부를 실시간으로 확인할 수 있어야 한다.                               | OpenCV               |
| 화재 감지              | 화재가 감지되면 경고음을 발생시켜야 한다.  | BLE장치 통신,            |
| 조명 및 개폐문 제어        | 웹 서버의 요청에 따라 조명 및 개폐문을 제어하고, 화재가 감지되면 화재 확산 방지를 위해 개폐문이 모두 닫혀야 한다. | BLE장치 통신, ServoESP32 |
| 온습도 측정             | 집안 내부의 온습도를 실시간으로 확인할 수 있어야 한다.                                    | BLE장치 통신,            |
| 웹 관리 시스템           | 웹을 통해 집안 내부의 영상과 온습도를 실시간으로 확인할 수 있고, 원격으로 조명 및 개폐문을 제어할 수 있어야 한다. | Flask, HTML,         |
| 침입자 감지 및 화재 알림 시스템 | 화재가 감지되거나 외출 모드 적용 후 카메라에 움직임이 감지되면 서버에 등록된 이메일에 해당 알림이 전송되어야 한다.  |                      |

#### 2.1.2. 비공학적 목표

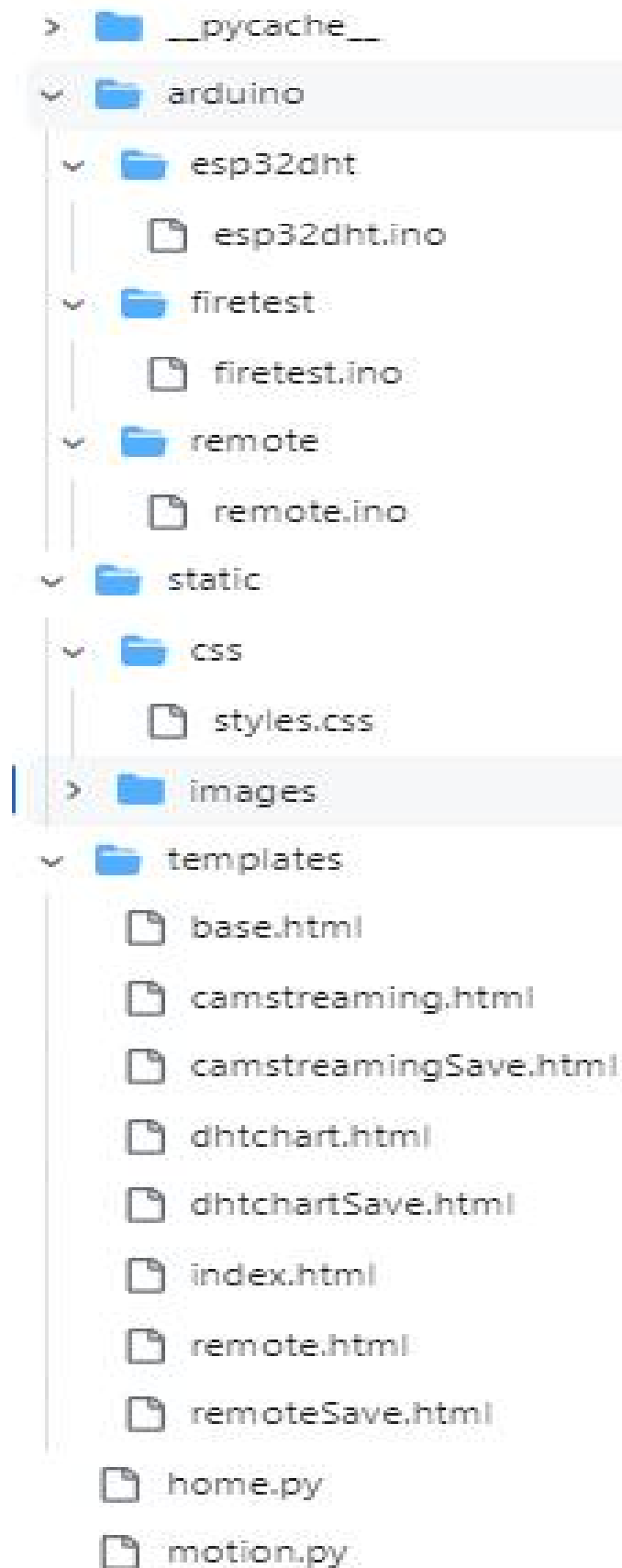
- 사용자 편의성 제고: 일반 사용자는 물론 노인이나 신체 장애인 등 사회적 소수자도 쉽게 사용할수 있도록 사용자 인터페이스(UI)를 단순화하고 접근성을 높인다.
- 안전하고 쾌적한 주거 환경 조성: 자동화된 환경 제어와 실시간 모니터링을 통해 사용자에게 안전하고 쾌적한 생활 환경을 제공한다.
- 사회적 포용성 증진: 스마트 홈 케어 시스템을 통해 노인 및 신체 장애인 등 사회적 소수자의 생활 편의를 증진시키며, 독거노인의 경우 건강 모니터링 및 응급상황 대응을 통해 안전한 생활을 지원한다.

## 2.2. 프로젝트 스펙

| 기능          | 하드웨어                  | 소프트웨어 및 라이브러리                                 |
|-------------|-----------------------|---|
| 홈 캠         | 파이 카메라                | OpenCV, smtplib                               |
| 화재 감지       | 불꽃 감지 센서              | smtplib                                       |
| 조명 및 개폐문 제어 | LED, Servo모터          | GPIO출력, ESP32Servo                            |
| 온습도 측정      | DHT22                 | DHT sensor Library                            |
| 웹 관리 시스템    | -                     | Flask, HTML                                   |
| 기타          | ESP32, Raspberry Pi 4 | Wifi, WifiServer, esp32, Raspberry Pi OS 5.10 |

## 2.3. 기능별 설계 및 구현 내용

### - 프로젝트 구조





### 2.3.1. 홈 캠 기능 구현 내용

#### 2.3.1.1. 카메라 프레임 생성

##### 소스 코드

```
def generate_frames():
    """ 스트리밍을 위한 프레임 생성 함수 """
    while True:
        with lock:
            success, frame = cam.read()
        if not success:
            break
        else:
            _, buffer = cv2.imencode('.jpg', frame)
            frame = buffer.tobytes()
            yield (b'--frame\r\n'
                  b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```

##### 부가 설명

홈 캠 스트리밍에 필요한 카메라 프레임 생성 코드

### 2.3.1.2. 움직임 감지 모듈

#### 소스 코드

```
1  # motion.py
2  import cv2
3
4  ▼ def getGrayCamImg(cam):
5      """카메라로부터 그레이스케일 이미지를 캡처하는 함수"""
6      ret, img = cam.read()
7      if not ret:
8          print("Failed to grab frame.")
9          return None
10     gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
11     return gray
12
13  ▼ def diffImage(i):
14      """세 이미지 간의 차이를 계산하는 함수"""
15      diff0 = cv2.absdiff(i[0], i[1])
16      diff1 = cv2.absdiff(i[1], i[2])
17      return cv2.bitwise_and(diff0, diff1)
18
19  ▼ def updateCameraImage(cam, i):
20      """카메라 이미지를 업데이트하는 함수"""
21      i[0] = i[1]
22      i[1] = i[2]
23      i[2] = getGrayCamImg(cam)
```

#### 부가 설명

시간에 따른 이미지의 차이를 계산하는 코드

### 2.3.1.3. 방법 기능

#### 소스 코드

```
86  def surveillance():
87      """ 방법 기능을 수행하는 함수 """
88      thresh = 50 # 움직임 감지 임계값
89      global surveillance_mode
90
91      if not cam.isOpened():
92          print("cam isn't opened")
93          return
94
95      i = [None, None, None]
96      flag = False
97
98      # 초기 이미지 3프레임 캡처
99      for n in range(3):
100          with lock:
101              i[n] = getGrayCamImg(cam)
102
103      checkFlag = 0
104
105      while True:
106          with lock:
107              diff = diffImage(i)
108              thrimg = cv2.threshold(diff, thresh, 1, cv2.THRESH_BINARY)[1]
109              count = cv2.countNonZero(thrimg)
110              time.sleep(1)
111              if not surveillance_mode:
112                  print('Surveillance mode OFF')
113                  continue
114              print("count :", count)
115              print("checkFlag :", checkFlag)
116              # 침입자 감지 시
117              if count > 1:
118                  checkFlag += 1
119                  if checkFlag >= 3 and not flag:
120                      sendMail(i[2])
121                      flag = True
122                      print("invader is coming!!!")
123              elif count == 0 and flag:
124                  flag = False
125                  checkFlag = 0
126              # 다음 이미지 처리
127              with lock:
128                  updateCameraImage(cam, i)
129              key = cv2.waitKey(10)
130              if key == 27: # ESC 키를 누르면 종료
131                  break
```

#### 부가 설명

움직임을 감지할 경우 메일을 전송하는 코드

### 2.3.1.4. 메일 전송 기능

#### 소스 코드

```
52  def sendMail(image=None):
53      """ 감지된 이미지를 이메일로 전송하는 함수 """
54      to = [userid]
55      msg = MIME multipart()
56      if image is not None:
57          imageByte = cv2.imencode(".jpeg", image)[1].tobytes()
58          imageMime = MIMEImage(imageByte)
59          msg['Subject'] = "Invader is Coming!!"
60          msg.attach(imageMime)
61      else:
62          msg['Subject'] = "Fire!!"
63          body = "Fire!!"
64          msg.attach(MIMEText(body, 'plain'))
65
66
67      msg['From'] = 'Me'
68      msg['To'] = to[0]
69
70      # SMTP 서버 연결 및 로그인
71      server = smtplib.SMTP(smtp_server, port)
72      server.ehlo_or_helo_if_needed()
73      ret, m = server.starttls()
74      server.ehlo_or_helo_if_needed()
75      ret, m = server.login(userid, passwd)
76
77      if ret != 235:
78          print("login fail")
79          return
80
81      # 이메일 전송
82      server.sendmail('rlaalsrhs59@gmail.com', to, msg.as_string())
83      server.quit()
```

#### 부가 설명

메일 전송 기능을 구현한 코드

## 2.3.2. 화재감지 기능 구현 내용

### 2.3.2.1. WiFi 설정

| 소스 코드   |
|---|
| <pre>const char* ssid = "mingon"; const char* password = "123456780";</pre> |
| 부가 설명   |
| ESP32가 연결할 WiFi 네트워크의 SSID와 비밀번호를 설정  |

### 2.3.2.2. 핀 설정

| 소스 코드  |
|--|
| <pre>int flame = 36; // ESP32의 아날로그 핀 int LED = 13; // ESP32의 디지털 핀 int count = 0; WiFiServer server(8080);</pre>      |
| 부가 설명  |
| flame은 불꽃 감지 센서가 연결된 아날로그 핀<br>LED는 부저가 연결된 디지털 핀<br>count는 불꽃이 연속으로 감지된 횟수를 저장하는 변수<br>server는 포트 8080에서 동작하는 웹 서버 객체 |

### 2.3.2.3. Setup() 함수

| 소스 코드   |
|---|
| <pre>void setup() {   Serial.begin(230400);   pinMode(flame, INPUT);   pinMode(LED, OUTPUT);   Serial.print("Connecting to ");   Serial.println(ssid);   WiFi.begin(ssid, password);   while (WiFi.status() != WL_CONNECTED) {     delay(500);     Serial.print(".");   }   Serial.println("WiFi connected");   server.begin();   Serial.println(WiFi.localIP()); }</pre> |
| 부가 설명   |
| 시리얼 통신을 초기화하고, 핀 모드를 설정<br>WiFi에 연결하고 연결되면 IP 주소를 출력<br>웹 서버 시작   |

### 2.3.2.4. loop() 함수

#### 소스 코드

```
void loop() {
  int val = analogRead(flame);
  Serial.print("flame_sensor : ");
  Serial.println(val);
  if (val > 4000) count ++;
  else count = 0;
  if (count > 4) {
    digitalWrite(LED, HIGH);
    Serial.println("FIRE!!!!");
  } else {
    digitalWrite(LED, LOW);
    Serial.println("NO FIRE");
  }
  WiFiClient client = server.available();
  if (client) {
    Serial.println("New client");
    String currentLine = "";
    String webString = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n') {
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:application/json");
            client.println();
            client.print(webString);
            client.println();
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
        if (currentLine.endsWith("GET /fireevents")) {
          if (count > 4) {
            webString = "{\"W\"statusW\": W\"FIREW\"}";
          } else {
            webString = "{\"W\"statusW\": W\"NO FIREW\"}";
          }
        }
      }
    }
    delay(1);
    client.stop();
    Serial.println("client disconnected");
  }
  delay(400);
}
```

#### 부가 설명

불꽃 감지 센서 값을 읽고 시리얼 모니터에 출력한다  
센서 값이 4000 이상일 때 count를 증가시키고, 그렇지 않으면 count를 0으로 초기화  
count가 4보다 크면 부저를 켜고, "FIRE!!!!", 그렇지 않으면 부저를 끄고 "NO FIRE"  
클라이언트가 웹 서버에 접속하면 현재 상태를 JSON 형식으로 반환  
/fireevents요청이 들어오면 불꽃이 감지되었는지 여부에 따라 "FIRE" 또는 "NO FIRE" 상태를 JSON  
형식으로 반환

### 2.3.3. 조명 및 개폐문 제어 기능 구현 내용

#### 2.3.3.1. 헤더파일 추가 및 전역 변수 생성

| 소스 코드   |
|---|
| <pre>#include &lt;WiFi.h&gt; #include &lt;WiFiServer.h&gt; #include &lt;ESP32Servo.h&gt; // 서보 모터 라이브러리 추가 Servo servo; // 서보 모터 객체 생성 Servo window; // 서보 모터 객체 생성</pre> |
| 부가 설명   |
| <p>Esp32보드와 와이파이 연결을 위한 헤더파일 Wifi.h, WiFiServer.h 와 esp32와 연결된 서보 모터를 제어하기 위한 ESP32Servo.h 헤더파일 추가<br/>서보 모터 제어를 위한 Servo 객체 생성</p>                                       |

※ 중복되는 코드는 설명 제외

#### 2.3.3.2. LED 제어 핀 설정 및 서보 모터 제어 핀 설정

| 소스 코드   |
|---|
| <pre>pinMode(LIGHT1_PIN, OUTPUT); digitalWrite(LIGHT1_PIN, LOW); // 기본적으로 LED를 끄  pinMode(LIGHT2_PIN, OUTPUT); digitalWrite(LIGHT2_PIN, LOW); // 기본적으로 LED를 끄  // 서보 모터 핀 설정 servo.attach(DOORPIN); window.attach(WINDOWPIN);</pre> |
| 부가 설명   |
| <p>LED와 서보 모터를 제어하기 위한 핀 값을 지정</p>  |

### 2.3.3.3. LED 제어 핀 설정 및 서보 모터 제어 핀 설정

#### 소스 코드

```
// LED 제어 핀 설정
if (currentLine.indexOf("light1_on") >= 0) {
    digitalWrite(LIGHT1_PIN, HIGH); // LED 켜기
    client.println("0");
} else if (currentLine.indexOf("light1_off") >= 0) {
    digitalWrite(LIGHT1_PIN, LOW); // LED 끄기
    client.println("1");

} else if (currentLine.indexOf("light2_on") >= 0) {
    digitalWrite(LIGHT2_PIN, HIGH); // LED 켜기
    client.println("0");
} else if (currentLine.indexOf("light2_off") >= 0) {
    digitalWrite(LIGHT2_PIN, LOW); // LED 끄기
    client.println("1");

} else if (currentLine.indexOf("open_door") >= 0) {
    // 서보 모터를 90도 회전
    servo.write(90);
    client.println("0");
} else if (currentLine.indexOf("close_door") >= 0) {
    // 서보 모터를 90도 회전
    servo.write(0);
    client.println("1");
} else if (currentLine.indexOf("open_window") >= 0) {
    // 서보 모터를 90도 회전
    window.write(90);
    client.println("0");
} else if (currentLine.indexOf("close_window") >= 0) {
    // 서보 모터를 90도 회전
    window.write(0);
    client.println("1");
} else {
    client.print("{\"status\": \"unknown command\"}");
}
```

#### 부가 설명

요청 에 대한 정보 currentLine에 각 기능에 해당하는 문자열이 포함되어 있으면 LED와 서보 모터를 제어함

※ 중복되는 코드는 설명 제외



### 2.3.4. 온습도 측정 기능 구현

#### 2.3.4.1. 헤더파일 추가 및 전역 변수 생성

| 소스 코드  |   |
|--|---|
| <pre>#include &lt;WiFi.h&gt; #include &lt;WiFiServer.h&gt; #include &lt;WiFiClient.h&gt; #include &lt;DHT.h&gt; #define DHTTYPE DHT22 #define DHTPIN 4</pre>                       | <pre>const char* ssid = "mingon"; const char* password = "123456780"; // Create an instance of the server WiFiServer server(8080); // Create an instance of DHT sensor DHT dht(DHTPIN, DHTTYPE); float temp, humi; String webString=""; unsigned long previousMillis = 0; const long interval = 1000; void gettemphumi();</pre> |
| 부가 설명  |   |
| <p>Wifi 관련 라이브러리와 온습도 센서 사용을 위한 DHT 라이브러리를 사용한다.</p> <p>온습도 센서 측정 데이터를 입력받는 PIN을 지정한다.</p> <p>Wifi ID와 Password를 입력하여 ESP보드를 Wifi에 연결한다.</p> <p>온도, 습도, 측정 시간 주기를 위한 변수를 지정한다.</p> |   |

#### 2.3.4.2. 측정한 온도 및 습도 데이터 전송

| 소스 코드   |  |
|---|--|
| <pre>client.print(webString); client.println(); break; } else {     currentLine = ""; } } else if (c!='\r') {     currentLine += c; } }  if (currentLine.endsWith("GET /dhtevents")) {     gettemphumi();     webString="{\"temperature\": \"" +String(temp) + "\", \"humidity\": \"" + String(humi)+ "\" }"; } } } delay(1);  client.stop(); Serial.println(webString); Serial.println("client disconnected");</pre> |  |
| 부가 설명   |  |
| <p>gettemphumi() 함수를 통해 온습도를 측정하고 /dhtevents, 서버를 통해 JSON형태로 데이터를 전송한다.</p>   |  |

### 2.3.4.3. 온도 및 습도 측정 함수

#### 소스 코드

```
void gettemphumi() {  
    unsigned long currentMillis = millis();  
    if (currentMillis - previousMillis >= interval) {  
        previousMillis = currentMillis;  
        humi = dht.readHumidity();  
        temp = dht.readTemperature(false);  
        if (isnan(humi) || isnan(temp)) {  
            Serial.println("Failed to read dht sensor.");  
            return;  
        }  
        Serial.print("Humidity: ");  
        Serial.print(humi);  
        Serial.print("% Temperature: ");  
        Serial.print(temp);  
        Serial.println("'c");  
    }  
}
```

#### 부가 설명

gettemphumi() 함수를 정의한다.

dht.read\_() 함수를 통해 온습도를 측정하여 변수에 저장하고, 변수에 저장된 값이 없을 경우 오류 메시지를 출력한다.

## 2.4. 프로젝트 작품의 구성요소

| 설계 구성요소 | 내용   |
|---------|--|
| 합성      | <ul style="list-style-type: none"> <li>- 고령화 사회와 1인 가구 증가에 대응하며, 사회적 약자를 위한 안전하고 편리한 주거 환경을 조성하는 스마트 홈 케어 시스템을 설계한다.</li> <li>- 비용 효율성 : 저비용으로 설치 및 유지보수를 할 수 있어야 한다.</li> <li>- 통합성 : 다양한 기기와 센서를 통합하여 제어할 수 있어야 한다.</li> <li>- 안전성, 편의성, 비용, 통합성 등 다양한 기준을 설정하여 목표 달성을 평가한다.</li> <li>- 일정은 각 단계의 중요성과 의존성을 고려하여 수립되며, 팀원들의 역할 분담은 각자의 전문성과 역량에 기반하여 계획된다.</li> </ul>  |
| 분석      | <ul style="list-style-type: none"> <li>- 프로그램의 핵심 구성 요소로는 침입자 감지, 온습도 측정, 화재 감지, 조명 제어 등이 있다.</li> <li>- 침입자 감지는 카메라로 움직임을 감지하여 침입자를 감지하는 역할을 수행한다.</li> <li>- 온습도 측정은 온습도 센서를 이용하여 온습도 센서를 일정 주기마다 측정하고 그 결과들을 보여준다.</li> <li>- 화재 감지는 불꽃 감지 센서로 불꽃을 감지하여 화재를 감지하고 메일을 보낸다.</li> <li>- 조명 제어는 웹 서버를 이용하여 공간 내에 존재하는 조명들을 제어할 수 있다.</li> <li>- 각 센서와 기기를 통합 플랫폼과 연결하여 센서 설치 및 연동을 수행한다.</li> <li>- 사용자 친화적인 웹 인터페이스를 설계한다.</li> <li>- 센서와 플랫폼 간의 데이터 전송이 수월하게 되도록 하고, 사용자 인터페이스의 직관성 및 사용 편의성을 위한 방안을 고려한다.</li> </ul> |
| 제작      | <ul style="list-style-type: none"> <li>- 파이 카메라, 온습도 센서, 조명, 불꽃 감지 센서 등 제어기기와 센서를 설치하고 설정한다.</li> <li>- 서버 설정 및 데이터 처리 시스템을 구현한다.</li> <li>- 웹 인터페이스를 구현한다.</li> </ul>   |
| 시험      | <ul style="list-style-type: none"> <li>- 움직임 감지, 온습도 측정, 조명 제어, 화재 감지 등 각 요소의 동작을 개별적으로 확인한다.</li> <li>- 각 요소들이 통합된 전체 시스템의 동작을 확인하고 부작용이 없는지 검토한다.</li> <li>- 다양한 상황에서의 동작을 시뮬레이션하여 프로그램의 견고성을 검증한다.</li> </ul>   |

|    |  |
|----|--|
|    | <ul style="list-style-type: none"> <li>- 동작 중 발생한 문제를 신속하게 파악하고 디버깅하여 수정한다.</li> </ul>   |
| 평가 | <ul style="list-style-type: none"> <li>- 정의된 요구 사항이 충족되었는지 평가하고 부족한 부분을 확인한다.</li> <li>- 정량적인 지표를 사용하여 시스템의 구현 정도를 평가한다.</li> <li>- 부족한 부분에 대한 원인 분석과 개선 방향을 문서화한다.</li> <li>- 발견된 결함이나 개선 사항을 정리하고 이를 수정하여 보완한다.</li> </ul> |
| 기타 | <ul style="list-style-type: none"> <li>- 웹 프레임워크, 라즈베리파이, 프로그래밍 언어 등 다양한 도구를 효과적으로 활용한다.</li> <li>- 추가적인 설계 주제에 따라 필요한 구성요소들을 효과적으로 도입하여 시스템을 보완한다.</li> </ul>   |

## 2.5. 설계 작품의 제한요소

| 설계 제한요소 | 내용   |
|---------|--|
| 원가      | <ul style="list-style-type: none"> <li>- 제작 및 유지에 필요한 비용을 최소화하여 프로그램을 경제적으로 제작해야 한다.</li> <li>- 각 구성 요소의 가격과 대체 가능성을 고려하여 예산 내에서 효과적인 프로그램을 개발해야 한다.</li> <li>- 프로그램이 특정 사용자를 대상으로 하기에, 그에 맞게 대량 생산이 가능하도록 설계되어야 한다.</li> </ul>  |
| 안정성     | <ul style="list-style-type: none"> <li>- 프로그램은 초기에 정의된 설계 목표를 충족하여 다양한 사용자에게 안정적인 보정을 제공해야 한다.</li> <li>- 외부 환경의 변화나 예상치 못한 입력에 효과적으로 대응하여 안정성을 유지해야 한다.</li> </ul>  |
| 신뢰성     | <ul style="list-style-type: none"> <li>- 프로그램은 최소한의 오류로 정확한 기능을 제공해야 한다.</li> <li>- 프로그램은 빠르게 작동하여 사용자에게 실시간으로 정확한 결과를 제공해야 한다.</li> <li>- 동일한 기능에 대해 항상 동일한 결과를 보장해야 한다.</li> <li>- 사용자의 역량에 영향받지 않고 쉽게 조작되어야 한다.</li> <li>- 관련된 산업 표준을 준수하여 신뢰성을 강화해야 한다.</li> </ul> |
| 미학      | <ul style="list-style-type: none"> <li>- 사용자 경험을 개선하고 직관적이며 사용하기 편리한 인터페이스를 제공해야 한다.</li> <li>- 사용자에게 명확한 조작 안내를 제공하여 오류를 최소화하고 조작자를 지원해야 한다.</li> <li>- 시각적으로 매력적이며 사용자 친화적인 디자인을 갖춰 사용자에게 호감을 줘야 한다.</li> </ul>  |
| 내구성     | <ul style="list-style-type: none"> <li>- 장시간 동안 안정적으로 동작하여 사용자에게 신뢰성을 제공해야 한다.</li> <li>- 다양한 환경에서 안정적으로 동작하여 외부 요인에 영향을 받지 않아야 한다.</li> </ul>   |
| 기타      | <ul style="list-style-type: none"> <li>- 사회적으로 수용 가능하고 색각 이상자의 권익을 보호하는 윤리적인 가치를 준수해야 한다.</li> <li>- 추가적인 설계 주제에 따른 특별한 제한 사항을 고려하여 색각 보정 프로그램을 개발해야 한다.</li> </ul>  |

## 2.6. 프로젝트 추진일정 및 방법

| 설계과제 추진전략 및 방법 |                         |      |   |   |   |   |   |   |   |   |           |  |
|----------------|-------------------------|------|---|---|---|---|---|---|---|---|-----------|--|
| No             | 과제 내용                   | 추진일정 |   |   |   |   |   |   |   |   | 기간<br>(주) | 추진 방법  |
|                |                         | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |           |  |
| 1              | 대략적 주제 선정               |      |   |   |   |   |   |   |   |   | 3         | 각자의 아이디어 공유 및 아이디어 회의를 통하여 추가적으로 필요한 기능 회의 및 추가                |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
| 2              | 관련정보 수집, 자료 조사 및 계획서 작성 |      |   |   |   |   |   |   |   |   | 2         | 인터넷 및 신문 자료 조사, PowerPoint와 hwp 활용하여 계획서 및 발표자료 작성             |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
| 3              | 프로젝트 계획서 제출 및 제안 발표     |      |   |   |   |   |   |   |   |   | 1         | 교수님께 추가 피드백 받기   |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
| 4              | 각 기능 하드웨어 연결 및 모듈 테스트   |      |   |   |   |   |   |   |   |   | 3         | Esp32 BLE 장치 센서 및 하드웨어 연결, 라즈베리파이에 파이 카메라 연결 및 모듈 테스트 소스 코드 개발 |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
| 5              | 모형 제작 및 테스트             |      |   |   |   |   |   |   |   |   | 2         | 모형 제작 및 기능 통합 테스트  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
| 6              | 프로젝트 보고서 작성             |      |   |   |   |   |   |   |   |   | 1         | PowerPoint 및 hwp 활용하여 보고서 및 발표자료 작성                            |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
| 7              | 프로젝트 기말 발표              |      |   |   |   |   |   |   |   |   | 1         | 수업시간 발표  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |
|                |                         |      |   |   |   |   |   |   |   |   |           |  |

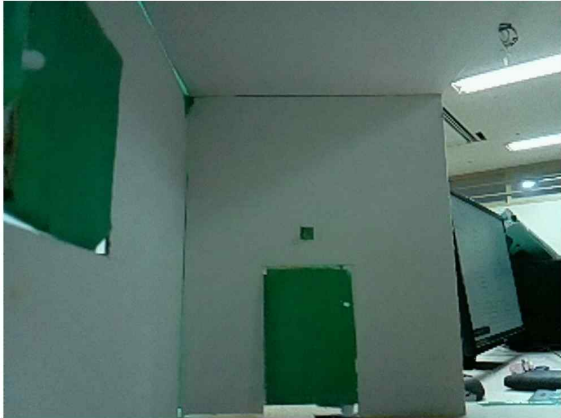
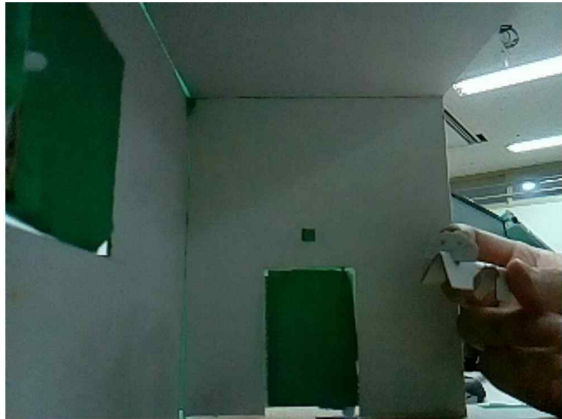
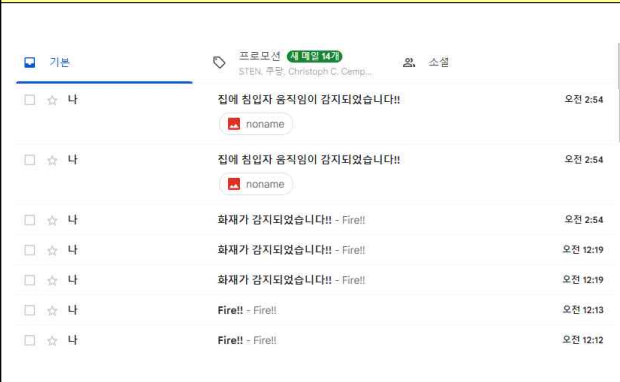


## 2.7. 설계 작품의 제한요소

| 성명 / 학번   |          | 수행분야  | 역할 |
|---|----------|---|----|
|    | 김민곤      | <ul style="list-style-type: none"> <li>카메라 모니터링 기능 개발</li> <li>침입자 감지 기능 개발</li> <li>이메일 전송 기능 개발</li> <li>전체 회로 연동</li> <li>모형 제작</li> <li>발표 자료 작성</li> </ul> | 팀장 |
|   | 20193172 |   |    |
|    | 남진우      | <ul style="list-style-type: none"> <li>조명 및 개폐문 제어 기능 개발</li> <li>모형 제작</li> <li>보고서 작성</li> <li>발표 자료 작성</li> </ul>  | 팀원 |
|   | 20193189 |   |    |
|   | 오승현      | <ul style="list-style-type: none"> <li>화재 감지 및 알림 기능 개발</li> <li>모형 제작</li> <li>보고서 작성</li> </ul>   | 팀원 |
|   | 20193183 |   |    |
|  | 김소희      | <ul style="list-style-type: none"> <li>온습도 모니터링 기능 개발</li> <li>보고서 작성</li> <li>발표 자료 작성</li> </ul>  | 팀원 |
|   | 20203164 |   |    |

### 3. 결론

#### 3.1. 프로젝트 결과 (결과 사진 등등)

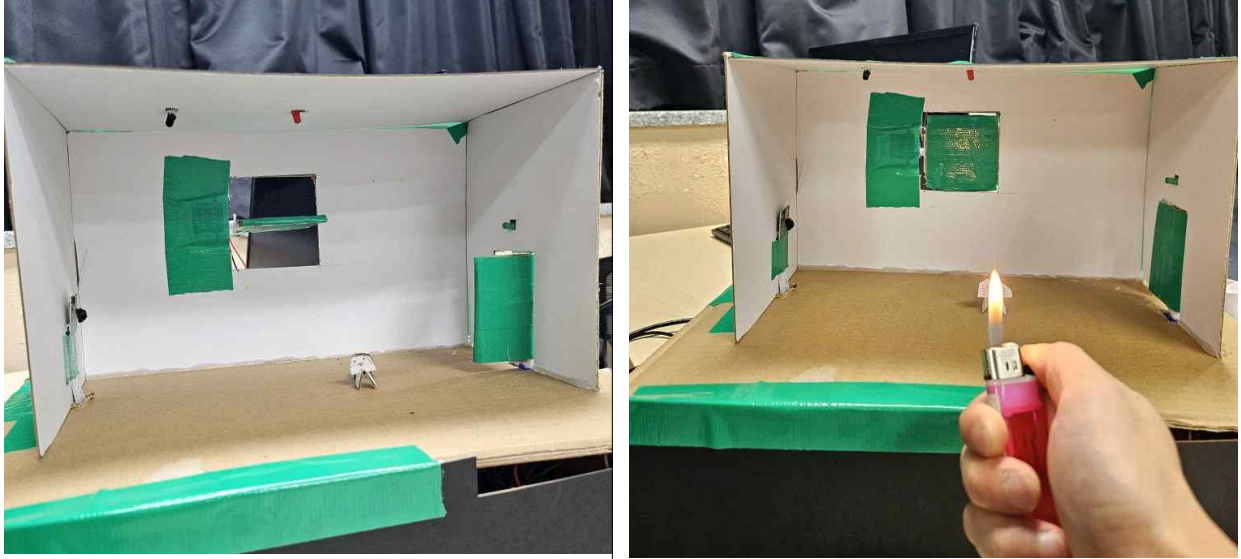
##### 3.1.1. 홈 캠 침입자 감지

| 홈 캠 결과  |  |
|---|--|
| <div><h2>Home Cam</h2><p>date: 2024-06-12 3:15:45</p><div>Surveillance mode ON</div><div>Surveillance mode OFF</div></div> | <div><h2>Home Cam</h2><p>date: 2024-06-12 3:15:34</p><div>Surveillance mode ON</div><div>Surveillance mode OFF</div></div>   |
| 이메일 전송  |  |
|    | <div><p>집에 침입자 움직임이 감지되었습니다!! <span>받은편지함 ✕</span></p><div><p>rlaalsrhs59@gmail.com<br/>나에게 ▼</p></div><p>첨부파일 1개 • Gmail에서 스캔함 ⓘ</p></div> |
| 부가 설명   |  |
| <p>라즈베리파이에 연결된 파이 카메라를 통해 확보된 영상을 웹 페이지에서 확인할 수 있고, 감시 기능을 키고 나서 영상에 움직임이 포착되면 서버에 등록된 이메일로 해당 사진과 함께 침입자 감지 알림 메일이 전송되는 것을 확인할 수 있다.</p>   |  |

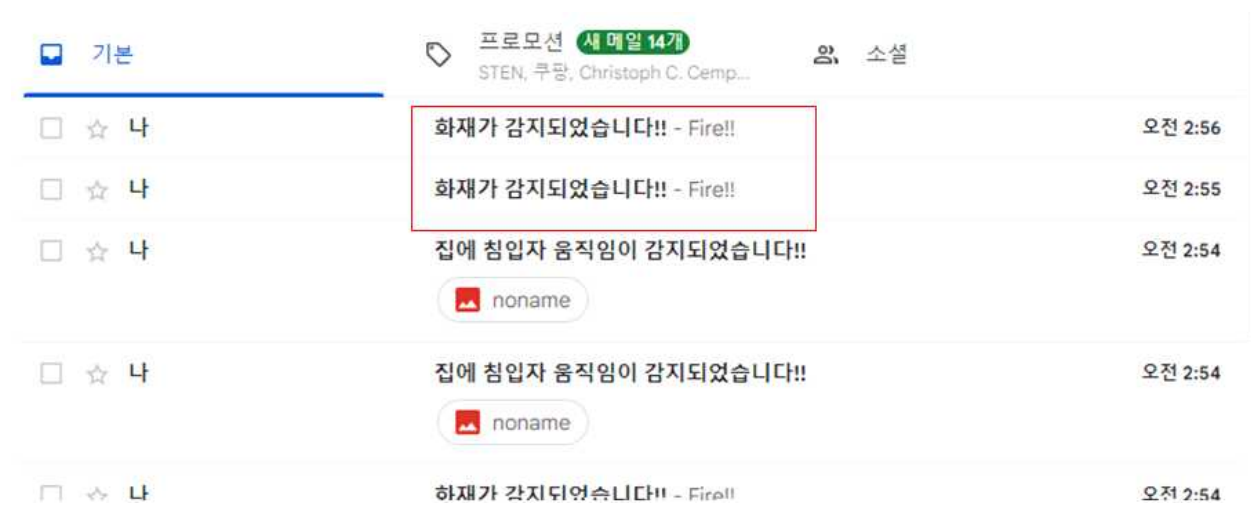


### 3.1.2. 화재 감지

#### 화재 감지 결과 사진








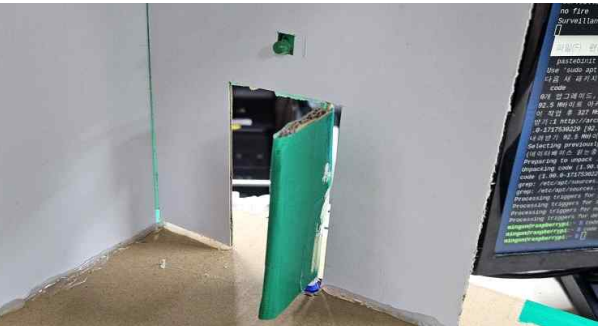
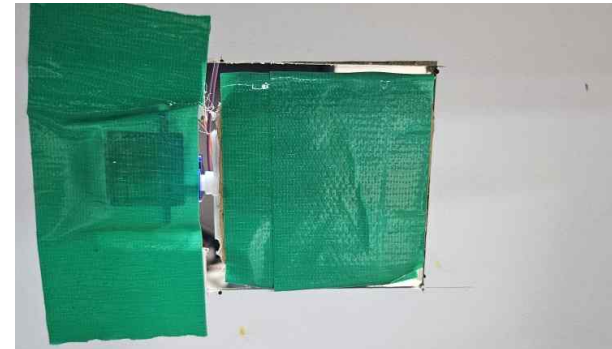
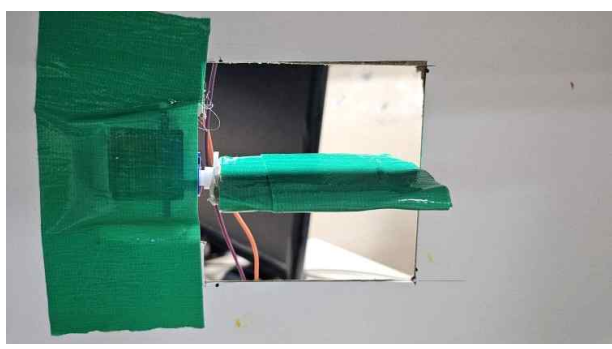
#### 이메일 전송



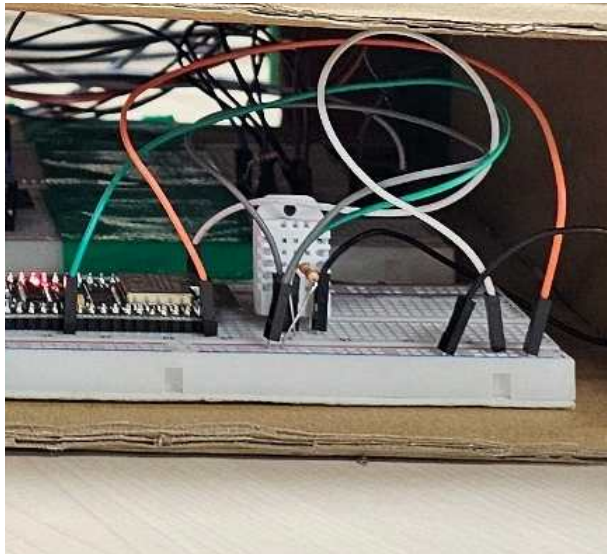
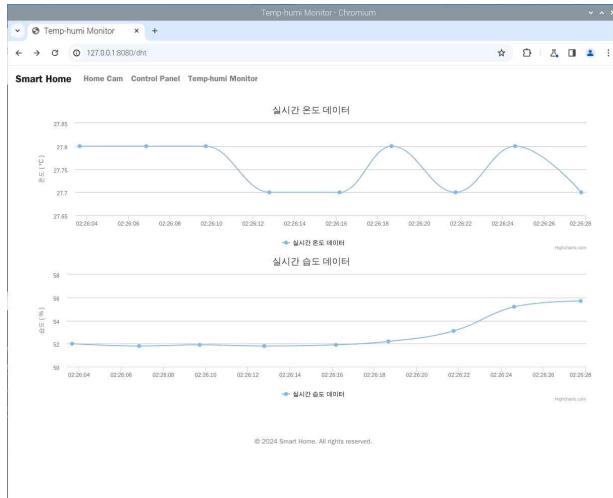
#### 부가 설명

esp32보드에 연결된 불꽃 감지 센서에 불꽃을 가져다 대면 불꽃에서 방출되는 적외선 파장을 감지하여 서버에 신호를 보내면 화재의 확산을 방지하기 위해 모든 개폐문이 닫히고, 서버에 등록된 이메일로 화재 알림 메일이 전송되는 것을 확인할 수 있다.

### 3.1.3. 조명 및 개폐문 제어

| 조명 제어   |  |
|---|--|
|                  |    |
|                 |   |
| 개폐문 제어  |  |
|                |  |
|                |  |
| 부가 설명   |  |
| <p>esp32보드에 연결된 하드웨어를 제어하기 위해 웹 사이트의 조명 및 개폐문 제어 버튼을 클릭하면 해당하는 조명과 개폐문이 열리고 닫히는 모습을 확인할 수 있다.</p> |  |

### 3.1.4. 화재 감지

| 온습도 감지 및 실시간 확인  |  |
|--|--|
| 온습도 센서   | 실시간 온습도 데이터 확인   |
|   |  |
| 부가 설명  |  |
| <p>esp32보드에 연결된 온습도 센서에서 측정된 온도와 습도에 대한 데이터를 실시간으로 서버에 전송하고 전송 받은 데이터를 기반으로 웹 사이트를 작성하여, 홈 케어 웹 사이트에서 온습도에 대한 실시간 데이터를 확인할 수 있다.</p> |  |

3.1.5. 웹 관리 시스템

| 웹 관리 시스템  |              |
|---|--------------|
| 메인 페이지  | 홈 카메라 모니터링   |
|   |              |
| 하드웨어 제어   | 온습도 실시간 모니터링 |
|   |              |
| 부가 설명   |              |
| <p>Flask라이브러리와 부트스트랩을 활용하여 홈 케어 시스템의 전반적인 기능들을 지원하는 웹 사이트를 확인할 수 있다. 메인페이지에서는 각 기능들을 제어, 모니터링할 수 있는 전용페이지로 연동 시켜주며 각 페이지에서는 하드웨어 제어, 홈 모니터링, 온습도 모니터링 기능들을 이용 할 수 있다.</p> |              |

### 3.2. 설계 목표의 중요도 및 달성도

| 목표                 | 중요도(%) | 달성도(%) | 수행내용   |
|--------------------|--------|--------|--|
| 홈 캠                | 15     | 100    | OpenCV 라이브러리를 활용하여 실시간으로 영상을 확인하는 기능 구현  |
| 화재 감지              | 15     | 100    | ESP32에 연결된 불꽃 감지 센서를 활용해 불꽃의 적외선이 3초이상 감지되면 비프음을 발생시키는 기능 구현                   |
| 조명 및 개폐문 제어        | 15     | 100    | 화재가 감지되면 개폐문을 모두 닫는 기능과 웹 관리 시스템의 요청에 따라 조명과 개폐문을 제어하는 기능 구현                   |
| 온습도 측정             | 15     | 100    | ESP32에 연결된 DHT22 온습도 센서를 활용하여 집안 내부의 온습도를 측정하고 일정 시간 간격으로 서버에 측정 값을 전송하는 기능 구현 |
| 웹 관리 시스템           | 20     | 100    | Flask 라이브러리를 활용하여 홈 캠 영상과 온습도 측정 그래프를 실시간으로 확인하고 조명 및 개폐문을 원격으로 제어하는 기능 구현     |
| 침입자 감지 및 화재 알림 시스템 | 20     | 100    | 화재가 감지되거나 외출 모드 적용 후 카메라에 움직임이 감지되면 서버에 등록된 이메일에 해당 알림이 전송되는 기능 구현             |
| 합계                 | 100    |        |  |

## [참고문헌]

- [1] 조경희, "헬스케어를 위한 상황인식기반 스마트홈 시스템", 조선대학교대학원 박사학위논문, 2009.
- [2] 이하철, "스마트홈 서비스 구현 및 진화 전망", 한국통신학회 추계종합학술발표회,p784-785, 2023.
- [3] 김진선, "액티브 시니어를 위한 스마트홈 헬스케어 서비스 적용에 관한 연구, 부경대학교대학원 석사학위논문, 2024.