

캡스톤 디자인 B

동아줄

(청각 장애인용 화재 경보 시스템)



세종대학교

기술 보고서

전자정보통신공학과

7조

조원 : 16010924 윤주혁

17010790 전호균

17010734 박태운

요 약(summary)

개발 환경	<ul style="list-style-type: none">○ 아두이노○ 앱인벤터
개요	<ul style="list-style-type: none">○ 일산화탄소 감지센서(MQ-7), 온습도 센서(DHT-11), 불꽃감지센서(HS-FLAME MODULE)를 사용하여 화재를 감지하고, 진동모터 모듈(DM159), LED, LED스트립(WS2812B)를 통해 화재 발생 상황을 인지시킨다.○ 블루투스 모듈(HC-06)을 사용해 아두이노로 수집한 데이터를 어플리케이션에 연동하여 사용자에게 최적의 탈출경로를 제시할 수 있도록 설계하였다.

목 차

[서론] I. 시스템 개발 환경 및 구성도

[본론] II. 각 구성별 기능 및 동작 원리

II-1. 화재감지 알고리즘

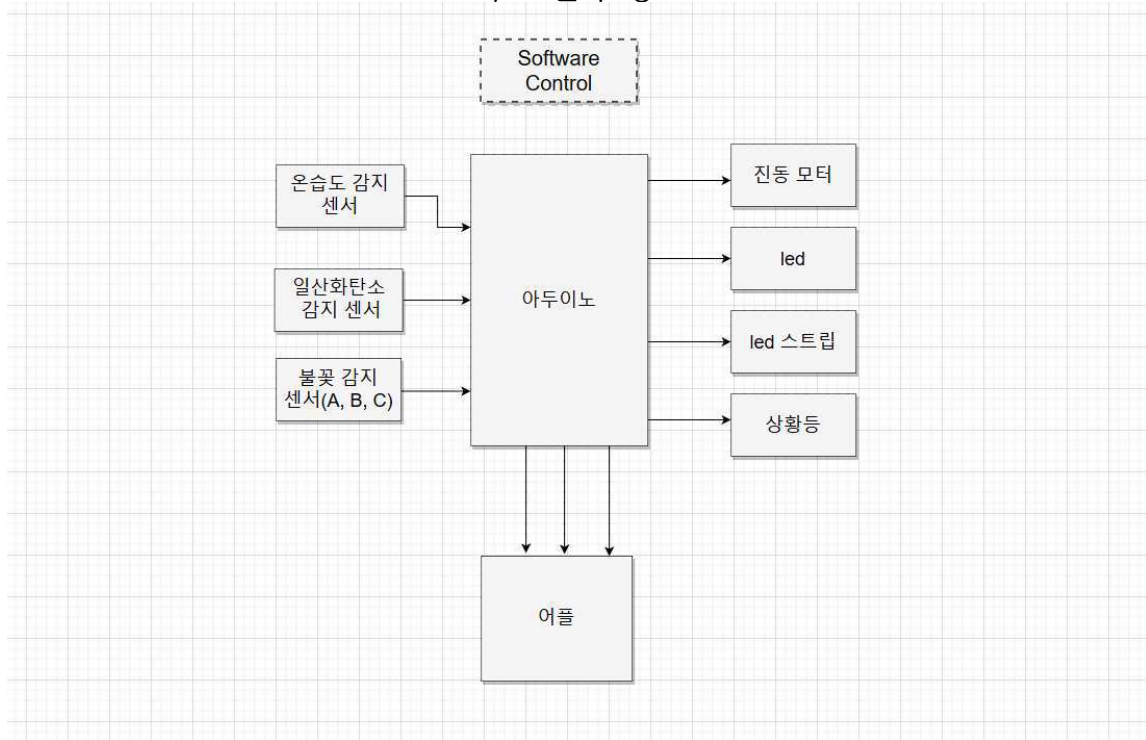
II-2. 화재발생 경고 알고리즘

II-3. 어플리케이션

[결론] III. 결론

I. 시스템 개발 환경 및 구성도

<시스템구성도>



온습도 센서와 일산화탄소 감지 센서, 불꽃 감지 센서를 사용하여 화재를 감지하고, 진동모터와 led, led스트립을 작동하여 화재 발생 상황을 인지시킨다. 또한 블루투스 모듈을 통해 아두이노에서 수집한 데이터를 스마트폰에 보내 앱인벤터 프로그램을 사용하여 만든 어플리케이션에서 현재 상황을 직관적으로 인식할 수 있도록 설계하였다. 어플리케이션에서 온·습도, 가스농도, 화재발생 여부를 확인할 수 있으며, 화재가 발생하였을 때 스마트폰에서 현재 gps 정보를 읽어 건물의 위치와 구조 요청을 소방서에 문자로 전송하고, 스마트폰 화면에 피난경로도를 띄워 대피를 원활하게 하고자 한다.

II. 각 구성별 기능 및 동작 원리

<아두이노 배치도>



II-1. 화재감지 알고리즘

방이 3개 있는 아파트 내부에 화재가 발생하였을 경우를 가정하여 설계하였다.

화재감지 알고리즘에는 일산화탄소 센서와 불꽃감지 센서를 사용하였는데, 거실 중앙에 일산화탄소 센서를 배치하고 각각의 방에 1~3번 불꽃감지 센서를 설치해 화재가 발생한 위치를 대략적으로 추측할 수 있고, 이에 따라 상황 1~6으로 나누어 유기적으로 대처할 수 있도록 하였다.

- 상황 1) 일산화탄소 센서, 1번 불꽃감지 센서 작동하는 경우 -> 1번 구역 화재발생
- 상황 2) 일산화탄소 센서, 2번 불꽃감지 센서 작동하는 경우 -> 2번 구역 화재발생
- 상황 3) 일산화탄소 센서, 3번 불꽃감지 센서 작동하는 경우 -> 3번 구역 화재발생
- 상황 4) 불꽃감지 센서는 작동하지 않고, 일산화탄소 센서만 작동하는 경우 -> 경고
- 상황 5) 일산화탄소 센서는 작동하지 않고, 불꽃감지 센서만 작동하는 경우 -> 경고
- 상황 6) 일산화탄소 센서와 불꽃감지 센서 모두 작동하지 않는 경우 -> 안전

<아두이노 코드>

```
1  #include <SoftwareSerial.h>      // 블루투스 통신을 위한 라이브러리
2  #include <Adafruit_NeoPixel.h>    // led스트립 사용을 위한 라이브러리
3  #include "DHT.h"                  // 온습도센서 사용을 위한 라이브러리
4
5  #define DHTPIN 5                  // 온습도센서 5번핀
6  #define DHTTYPE DHT11            // 온습도센서 종류
7  #define flame_1 2                // 1번 불꽃감지 2번핀
8  #define flame_2 3                // 2번 불꽃감지 3번핀
9  #define flame_3 4                // 3번 불꽃감지 4번핀
10 #define vibrator 6               // 진동모터 6번핀
11 #define BTtx 7                   // 블루투스 TX 7번핀
12 #define BTrx 8                   // 블루투스 RX 8번핀
13
```

#include : 아두이노에서 온습도센서, 블루투스 모듈, led스트립을 사용하기 위한 라이브러리 추가 함수

#define : 보드에서 사용된 핀의 번호 지정

```
22 SoftwareSerial BT(BTtx, BTrx); //Rx, Tx
23 DHT dht(DHTPIN, DHTTYPE);      //온습도센서 핀번호와 종류를 설정
24
25 int flame_state_1 = 1; // 불꽃감지 센서 on-off 확인 변수
26 int flame_state_2 = 1; // 불꽃감지 센서 on-off 확인 변수
27 int flame_state_3 = 1; // 불꽃감지 센서 on-off 확인 변수
28 int Gaspin = A0;          // 일산화탄소 감지센서 핀 번호 A0
29 int situation = 0;        // 현재 상황을 7가지로 나누어 대치
```

블루투스 모듈, 온습도 센서 핀번호 설정

불꽃 감지에 대한 변수 'flame_state' 선언

일산화탄소 농도에 대한 아날로그 변수 'Gaspin' 선언

일산화탄소 농도와 불꽃감지센서 감지 여부에 따라 상황 1~6으로 분류하여 변수 'situation' 선언

```
42 void setup() {
43   pinMode(Gaspin, INPUT); // 가스 감지 센서를 입력으로 설정
44   pinMode(LED_1, OUTPUT); // 1번 소화기 위치 led 출력으로 설정
45   pinMode(LED_2, OUTPUT); // 2번 소화기 위치 led 출력으로 설정
46   pinMode(MOOD_R, OUTPUT); // 무드등 빨간색 출력
47   pinMode(MOOD_G, OUTPUT); // 무드등 초록색 출력
48   pinMode(MOOD_Y, OUTPUT); // 무드등 노란색 출력
49   pinMode(flame_1, INPUT); // 1번 불꽃 감지 센서를 입력으로 설정
50   pinMode(flame_2, INPUT); // 2번 불꽃 감지 센서를 입력으로 설정
51   pinMode(flame_3, INPUT); // 3번 불꽃 감지 센서를 입력으로 설정
52   pinMode(vibrator, OUTPUT); // 진동모터 출력으로 설정
53   Serial.begin(9600);        // 시리얼 모니터 통신 시작
54   BT.begin(9600);           // 블루투스 통신 시작
55   dht.begin();               // 온습도센서
```

setup 단계

불꽃감지센서, 일산화탄소 센서 input 설정

LED, 진동모터 output 설정

블루투스, 온습도센서 시작

```
68 void loop() {
69   flame_state_1 = digitalRead(flame_1); // 1번 불꽃 감지 센서값 저장
70   flame_state_2 = digitalRead(flame_2); // 2번 불꽃 감지 센서값 저장
71   flame_state_3 = digitalRead(flame_3); // 3번 불꽃 감지 센서값 저장
72   float temp = dht.readTemperature(); // temp 변수에 온도값 소수점 단위 저장
73   float humi = dht.readHumidity();    // humi 변수에 습도값 소수점 단위 저장
74   int conc = analogRead(Gaspin);      // 일산화탄소 센서값 정수형 변수 conc에 저장
```

불꽃감지센서, 온습도, 일산화탄소 센서의 data를 각각의 변수에 저장

```

76 | if (conc > 200){ // 일산화탄소 농도 200ppm 이상일 때
77 |   if (flame_state_1 == 0 && flame_state_2 == 1 && flame_state_3 == 1){ // 1번 위치에 불이 났을 경우 & 2,3번 화재 감지 x
78 |     situation = 1; // 현재 상황 : 1번 위치 화재 발생
79 |     Serial.println("Fire_1"); // 화재 발생
80 |     analogWrite(vibrator,200); // 진동모터 on
81 |     digitalWrite(MOOD_R, HIGH); // 무드등 빨간색 on
82 |     digitalWrite(MOOD_G, LOW); // 무드등 초록색 off
83 |     digitalWrite(MOOD_Y, LOW); // 무드등 노란색 off
84 |     digitalWrite(LED_1, HIGH); // 소화기 위치 알람등 on
85 |     digitalWrite(LED_2, HIGH); // 소화기 위치 알람등 on

```

상황 1)

일산화탄소 농도 200ppm 이상, 1번 불꽃감지센서 동작할 경우 -> 1번 구역 화재발생
진동모터 on, 무드등 빨간색 on, 소화기 위치 알람등 on

```

111 | else if (flame_state_1 == 1 && flame_state_2 == 0 && flame_state_3 == 1){ // 2번 위치에 불이 났을 경우 & 1,3번 화재 감지 x
112 |   situation = 2; // 현재 상황 : 2번 위치 화재 발생
113 |   Serial.println("Fire_2"); // 화재 발생
114 |   analogWrite(6,200); // 진동모터 on
115 |   digitalWrite(MOOD_R, HIGH); // 무드등 빨간색 on
116 |   digitalWrite(MOOD_G, LOW); // 무드등 초록색 off
117 |   digitalWrite(MOOD_Y, LOW); // 무드등 노란색 off
118 |   digitalWrite(LED_1, HIGH); // 소화기 위치 알람등 on
119 |   digitalWrite(LED_2, HIGH); // 소화기 위치 알람등 on

```

상황 2)

일산화탄소 농도 200ppm 이상, 2번 불꽃감지센서 동작할 경우 -> 2번 구역 화재발생
진동모터 on, 무드등 빨간색 on, 소화기 위치 알람등 on

```

147 | else if (flame_state_1 == 1 && flame_state_2 == 1 && flame_state_3 == 0){ // 3번 위치에 불이 났을 경우 & 1,2번 위치 화재 감지 x
148 |   situation = 3; // 현재 상황 : 3번 위치 화재 발생
149 |   Serial.println("Fire_3"); // 화재 발생
150 |   analogWrite(6,200); // 진동모터 on
151 |   digitalWrite(MOOD_R, HIGH); // 무드등 빨간색 on
152 |   digitalWrite(MOOD_G, LOW); // 무드등 초록색 off
153 |   digitalWrite(MOOD_Y, LOW); // 무드등 노란색 off
154 |   digitalWrite(LED_1, HIGH); // 소화기 위치 알람등 on
155 |   digitalWrite(LED_2, HIGH); // 소화기 위치 알람등 on

```

상황 3)

일산화탄소 농도 200ppm 이상, 3번 불꽃감지센서 동작할 경우 -> 3번 구역 화재발생
진동모터 on, 무드등 빨간색 on, 소화기 위치 알람등 on

```

183 | else if(flame_state_1 == 1 && flame_state_2 == 1 && flame_state_3 == 1){ // 1,2,3 위치 모두 불이 안났을 경우(가스만 감지되었을 경우)
184 |   situation = 4; // 현재 상황 : 가스 감지(경고)
185 |   Serial.println("Gas Detection"); // 가스 감지
186 |   analogWrite(6,0); // 진동모터 off
187 |   digitalWrite(MOOD_R, LOW); // 무드등 빨간색 off
188 |   digitalWrite(MOOD_G, LOW); // 무드등 초록색 off
189 |   digitalWrite(MOOD_Y, HIGH); // 무드등 노란색 on
190 |   digitalWrite(LED_1, LOW); // 소화기 위치 알람등 off
191 |   digitalWrite(LED_2, LOW); // 소화기 위치 알람등 off

```

상황 4)

불꽃감지센서 모두 동작하지 않고, 일산화탄소 농도 200ppm 이상일 경우 -> 경고
진동모터 off, 무드등 노란색 on, 소화기 위치 알람등 off

```

209 | else {                                     // 일산화탄소 농도 200ppm 이하일 때
210 |     if (flame_state_1 == 0 || flame_state_2 == 0 || flame_state_3 == 0){ // 셋 중 하나라도 불꽃이 감지되면
211 |         situation = 5;                     // 현재 상황 : 불꽃 감지(경고)
212 |         Serial.println("Warning");         // 불꽃 발생 위험
213 |         analogWrite(6,0);                 // 진동모터 on
214 |         digitalWrite(MOOD_R, LOW);        // 무드등 빨간색 off
215 |         digitalWrite(MOOD_G, LOW);        // 무드등 초록색 off
216 |         digitalWrite(MOOD_Y, HIGH);       // 무드등 노란색 on
217 |         digitalWrite(LED_1, HIGH);        // 소화기 위치 알람등 on
218 |         digitalWrite(LED_2, HIGH);        // 소화기 위치 알람등 on

```

상황 5)

일산화탄소 농도 200ppm 이하일 때, 불꽃감지센서 중 어느 하나라도 동작할 경우 -> 경고
진동모터 off, 무드등 노란색 on, 소화기 위치 알람등 on

```

235 | else{
236 |     Serial.println("SAFE");               // 불꽃 감지 x
237 |     analogWrite(6,0);                     // 진동모터 off
238 |     digitalWrite(MOOD_R, LOW);            // 무드등 빨간색 off
239 |     digitalWrite(MOOD_G, HIGH);          // 무드등 초록색 on
240 |     digitalWrite(MOOD_Y, LOW);           // 무드등 노란색 off
241 |     digitalWrite(LED_1, LOW);            // 소화기 위치 알람등 off
242 |     digitalWrite(LED_2, LOW);            // 소화기 위치 알람등 off

```

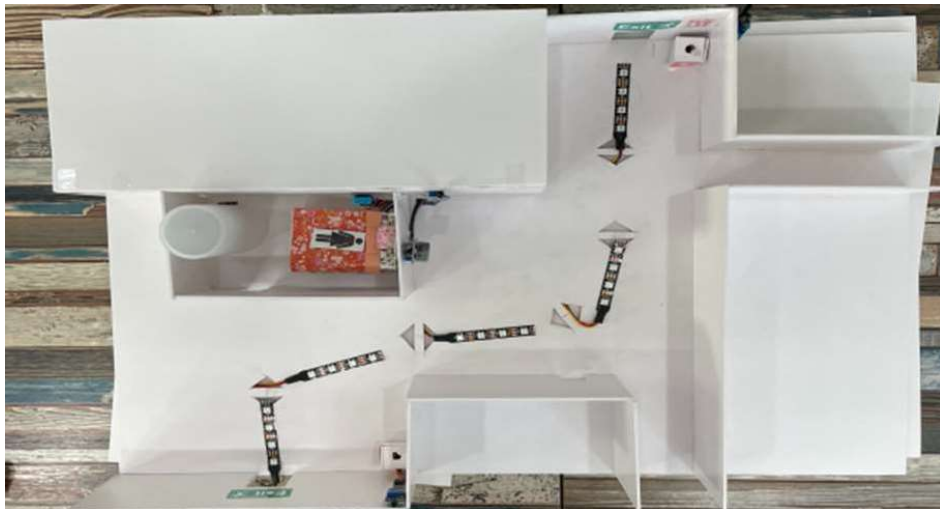
상황 6)

불꽃감지센서, 일산화탄소 센서 모두 동작하지 않을 경우 -> 안전
진동모터 off, 무드등 초록색 on, 소화기 위치 알람등 off

II -2. 화재 발생 경고 알고리즘

화재 발생 경고 알고리즘에는 진동모터와 led, led스트립을 사용하였다. 화재가 발생하였을 때, 소리를 듣지 못하는 청각장애인을 위해 진동모터를 사용하여 위험 상황을 인지시키고, 건물전체가 정전되어 앞이 잘 보이지 않는 상황을 대비하여 led를 통해 소화기의 위치를 알려준다. 또한 화재가 발생한 구역에 따라 서로 다른 최적의 탈출 경로를 제시하는 것을 목표로 하였다. 바닥에 설치된 led스트립은 탈출 방향으로 순서대로 켜지도록 하여 사용자에게 최적의 대피경로를 안내한다.

<아두이노 배치도>



왼쪽아래 led스트립부터 오른쪽 위 순서로 1~5번이다.


```

14 #define PIN_1 13      // 1번 LED 스트립 13번핀
15 #define PIN_2 12      // 2번 LED 스트립 12번핀
16 #define PIN_3 11      // 3번 LED 스트립 11번핀
17 #define PIN_4 10      // 4번 LED 스트립 10번핀
18 #define PIN_5 9       // 5번 LED 스트립 9번핀
19
20 #define NUM_LEDS 4     // led 스트립의 led소자 개수 4개

```

led소자의 개수가 4개인 led스트립을 5개를 사용하기 위해 각각에 9~13번의 핀 번호를 부여하였다.

```

36 Adafruit_NeoPixel strip_1 = Adafruit_NeoPixel(NUM_LEDS, PIN_1, NEO_GRB + NEO_KHZ800); // LED 스트립 구동 코드(led소자수,핀번호,네오피셀 종류)
37 Adafruit_NeoPixel strip_2 = Adafruit_NeoPixel(NUM_LEDS, PIN_2, NEO_GRB + NEO_KHZ800); // LED 스트립 구동 코드(led소자수,핀번호,네오피셀 종류)
38 Adafruit_NeoPixel strip_3 = Adafruit_NeoPixel(NUM_LEDS, PIN_3, NEO_GRB + NEO_KHZ800); // LED 스트립 구동 코드(led소자수,핀번호,네오피셀 종류)
39 Adafruit_NeoPixel strip_4 = Adafruit_NeoPixel(NUM_LEDS, PIN_4, NEO_GRB + NEO_KHZ800); // LED 스트립 구동 코드(led소자수,핀번호,네오피셀 종류)
40 Adafruit_NeoPixel strip_5 = Adafruit_NeoPixel(NUM_LEDS, PIN_5, NEO_GRB + NEO_KHZ800); // LED 스트립 구동 코드(led소자수,핀번호,네오피셀 종류)

```

Adafruit_NeoPixel strip : 5개의 led스트립 각각에 led소자 수, 핀 번호, 네오피셀 종류를 지정해준다.

```

56 strip_1.begin();           // 1~5번 led스트립 구동시작 및 초기화
57 strip_1.show();
58 strip_2.begin();
59 strip_2.show();
60 strip_3.begin();
61 strip_3.show();
62 strip_4.begin();
63 strip_4.show();
64 strip_5.begin();
65 strip_5.show();

```

1~5번 led스트립 구동시작

```

strip.setPixelColor(0, 255, 0, 0); // Neopixel 색상 설정 ( 첫번째 소자위치 , 색상설정(Red) , 0 , 0 )
strip.setPixelColor(1, 0, 255, 0); // ( 두번째 소자위치 , 0 , 색상설정(Green) , 0 )
strip.setPixelColor(2, 0, 0, 255); // ( 세번째 소자위치 , 0 , 0 , 색상설정(Blue) )
strip.setPixelColor(3, 255, 255, 255); // ( 네번째 소자위치 , (Red) , (Green) , (Blue) ) 3가지 색을 다 킨다면 White가 커
집
strip.show(); // LED가 켜지는 동작을 하게 합니다

```

setPixelColor : led스트립 소자 각각의 색을 지정한다. (소자 번호, R, G, B)
 예를 들어 2번째 소자에 초록색 빛을 켜려면 “strip.setPixelColor(1,0,255,0);” 라고 선언한다.



< 1번 구역 화재 발생 >

< 2번 구역 화재 발생 >

< 3번 구역 화재 발생 >

```

76  if (conc > 200){                                     // 일산화탄소 농도 200ppm 이상일 때
77      if (flame_state_1 == 0 && flame_state_2 == 1 && flame_state_3 == 1){ // 1번 위치에 불이 났을 경우 & 2,3번 화재 감지 x
78          situation = 1;                               // 현재 상황 : 1번 위치 화재 발생
79          Serial.println("Fire_1");                   // 화재 발생
80          analogWrite(vibrator,200);                   // 진동모터 on
81          digitalWrite(MOOD_R, HIGH);                  // 무드등 빨간색 on
82          digitalWrite(MOOD_G, LOW);                   // 무드등 초록색 off
83          digitalWrite(MOOD_Y, LOW);                   // 무드등 노란색 off
84          digitalWrite(LED_1, HIGH);                   // 소화기 위치 알람등 on
85          digitalWrite(LED_2, HIGH);                   // 소화기 위치 알람등 on
86
87          for(int i=0;i<4;i++){                         // 탈출경로를 알리기 위해 led 스트립 소자 순서대로 켜지도록 함
88              strip_1.setPixelColor(i, 0, 0, 0);      // 1번 스트립(led소자 순서,red,green,blue)
89              strip_2.setPixelColor(i, 0, 0, 0);      // 2번 스트립(led소자 순서,red,green,blue)
90              strip_3.setPixelColor(i, 0, 255, 0);    // 3번 스트립(led소자 순서,red,green,blue)
91              strip_4.setPixelColor(i, 0, 255, 0);    // 4번 스트립(led소자 순서,red,green,blue)
92              strip_5.setPixelColor(i, 0, 255, 0);    // 5번 스트립(led소자 순서,red,green,blue)
93              strip_1.show();
94              strip_2.show();
95              strip_3.show();
96              strip_4.show();
97              strip_5.show();
98          }
99          delay(1000);
100
101          for(int i=0;i<4;i++){
102              strip_3.setPixelColor(i, 0, 0, 0);
103              strip_4.setPixelColor(i, 0, 0, 0);
104              strip_5.setPixelColor(i, 0, 0, 0);
105              strip_3.show();
106              strip_4.show();
107              strip_5.show();
108          }

```

상황 1)

1번 구역에서 화재가 발생하면 아래 출입구가 아닌 위쪽 출입구로 대피하는 것이 안전하다. 따라서 1,2번 led스트립은 끄고 3,4,5번 led스트립을 위쪽 출입구 방향으로 순서대로 켜지는 것을 반복하여 대피경로를 안내한다.

```

111     else if (flame_state_1 == 1 && flame_state_2 == 0 && flame_state_3 == 1){        // 2번 위치에 불이 났을 경우 & 1,3번 화재 감지 x
112         situation = 2;                    // 현재 상황 : 2번 위치 화재 발생
113         Serial.println("Fire_2");        // 화재 발생
114         analogWrite(6,200);              // 진동모터 on
115         digitalWrite(MOOD_R, HIGH);      // 무드등 빨간색 on
116         digitalWrite(MOOD_G, LOW);       // 무드등 초록색 off
117         digitalWrite(MOOD_Y, LOW);       // 무드등 노란색 off
118         digitalWrite(LED_1, HIGH);       // 소화기 위치 알람등 on
119         digitalWrite(LED_2, HIGH);       // 소화기 위치 알람등 on
120
121         for(int i=0;i<4;i++){              // 탈출경로를 알리기 위해 led 스트립 소자 순서대로 켜지도록 함
122             strip_1.setPixelColor(4-i, 0, 255, 0);
123             strip_2.setPixelColor(4-i, 0, 255, 0);
124             strip_3.setPixelColor(i, 0, 0, 0);
125             strip_4.setPixelColor(i, 0, 255, 0);
126             strip_5.setPixelColor(i, 0, 255, 0);
127             strip_1.show();
128             strip_2.show();
129             strip_3.show();
130             strip_4.show();
131             strip_5.show();
132         }
133         delay(1000);
134
135         for(int i=0;i<4;i++){
136             strip_1.setPixelColor(i, 0, 0, 0);
137             strip_2.setPixelColor(i, 0, 0, 0);
138             strip_4.setPixelColor(i, 0, 0, 0);
139             strip_5.setPixelColor(i, 0, 0, 0);
140             strip_1.show();
141             strip_2.show();
142             strip_4.show();
143             strip_5.show();

```

상황 2)

2번 구역에서 화재가 발생하면 3번 스트립은 동작을 멈추고, 1,2번 스트립은 탈출경로를 아래쪽으로 4,5번 스트립은 탈출경로를 위쪽으로 안내한다.

```

147     else if (flame_state_1 == 1 && flame_state_2 == 1 && flame_state_3 == 0){        // 3번 위치에 불이 났을 경우 & 1,2번 위치 화재 감지 x
148         situation = 3;                    // 현재 상황 : 3번 위치 화재 발생
149         Serial.println("Fire_3");        // 화재 발생
150         analogWrite(6,200);              // 진동모터 on
151         digitalWrite(MOOD_R, HIGH);      // 무드등 빨간색 on
152         digitalWrite(MOOD_G, LOW);       // 무드등 초록색 off
153         digitalWrite(MOOD_Y, LOW);       // 무드등 노란색 off
154         digitalWrite(LED_1, HIGH);       // 소화기 위치 알람등 on
155         digitalWrite(LED_2, HIGH);       // 소화기 위치 알람등 on
156
157         for(int i=0;i<4;i++){              // 탈출경로를 알리기 위해 led 스트립 소자 순서대로 켜지도록 함
158             strip_1.setPixelColor(4-i, 0, 255, 0);
159             strip_2.setPixelColor(4-i, 0, 255, 0);
160             strip_3.setPixelColor(4-i, 0, 255, 0);
161             strip_4.setPixelColor(4-i, 0, 255, 0);
162             strip_5.setPixelColor(i, 0, 0, 0);
163             strip_1.show();
164             strip_2.show();
165             strip_3.show();
166             strip_4.show();
167             strip_5.show();
168         }
169         delay(1000);
170
171         for(int i=0;i<4;i++){
172             strip_1.setPixelColor(i, 0, 0, 0);
173             strip_2.setPixelColor(i, 0, 0, 0);
174             strip_3.setPixelColor(i, 0, 0, 0);
175             strip_4.setPixelColor(i, 0, 0, 0);
176             strip_1.show();
177             strip_2.show();
178             strip_3.show();
179             strip_4.show();

```

상황 3)

3번 구역에서 화재가 발생하면 5번 스트립은 동작을 멈추고, 1,2,3,4번 스트립은 탈출경로를 아래쪽으로 안내한다.

```

183     else if(flame_state_1 == 1 && flame_state_2 == 1 && flame_state_3 == 1){ // 1,2,3 위치 모두 불이 안났을 경우(가스만 감지되었을 경우)
184         situation = 4; // 현재 상황 : 가스 감지(경고)
185         Serial.println("Gas Detection"); // 가스 감지
186         analogWrite(6,0); // 진동모터 off
187         digitalWrite(MOOD_R, LOW); // 무드등 빨간색 off
188         digitalWrite(MOOD_G, LOW); // 무드등 초록색 off
189         digitalWrite(MOOD_Y, HIGH); // 무드등 노란색 on
190         digitalWrite(LED_1, LOW); // 소화기 위치 알람등 off
191         digitalWrite(LED_2, LOW); // 소화기 위치 알람등 off
192         delay(10);
193
194         for(int i=0;i<4;i++){
195             strip_1.setPixelColor(i, 0, 0, 0);
196             strip_2.setPixelColor(i, 0, 0, 0);
197             strip_3.setPixelColor(i, 0, 0, 0);
198             strip_4.setPixelColor(i, 0, 0, 0);
199             strip_5.setPixelColor(i, 0, 0, 0);
200             strip_1.show();
201             strip_2.show();
202             strip_3.show();
203             strip_4.show();
204             strip_5.show();
205         }
206     }

```

상황4)

불꽃감지 센서는 작동하지 않고, 일산화탄소 센서만 작동하는 경우 무드등의 색깔을 노란색으로 출력하여 사용자에게 경고를 알린다.

```

209     else { // 일산화탄소 농도 200ppm 이하일 때
210         if (flame_state_1 == 0 || flame_state_2 == 0 || flame_state_3 == 0){ // 셋 중 하나라도 불꽃이 감지되면
211             situation = 5; // 현재 상황 : 불꽃 감지(경고)
212             Serial.println("Warning"); // 불꽃 발생 위험
213             analogWrite(6,0); // 진동모터 off
214             digitalWrite(MOOD_R, LOW); // 무드등 빨간색 off
215             digitalWrite(MOOD_G, LOW); // 무드등 초록색 off
216             digitalWrite(MOOD_Y, HIGH); // 무드등 노란색 on
217             digitalWrite(LED_1, LOW); // 소화기 위치 알람등 off
218             digitalWrite(LED_2, LOW); // 소화기 위치 알람등 off
219
220             for(int i=0;i<4;i++){
221                 strip_1.setPixelColor(i, 0, 0, 0);
222                 strip_2.setPixelColor(i, 0, 0, 0);
223                 strip_3.setPixelColor(i, 0, 0, 0);
224                 strip_4.setPixelColor(i, 0, 0, 0);
225                 strip_5.setPixelColor(i, 0, 0, 0);
226                 strip_1.show();
227                 strip_2.show();
228                 strip_3.show();
229                 strip_4.show();
230                 strip_5.show();
231             }

```

상황5) 일산화탄소 센서는 작동하지 않고, 불꽃감지 센서만 작동하는 경우 상황 4번과 마찬가지로 무드등의 색깔을 노란색으로 출력하여 경고한다.

```

234     else{
235         situation = 6;                // 현재 상황 : 안전
236         Serial.println("SAFE");      // 불꽃 감지 x
237         analogWrite(6,0);            // 진동모터 off
238         digitalWrite(MOOD_R, LOW);    // 무드등 빨간색 off
239         digitalWrite(MOOD_G, HIGH);   // 무드등 초록색 on
240         digitalWrite(MOOD_Y, LOW);    // 무드등 노란색 off
241         digitalWrite(LED_1, LOW);     // 소화기 위치 알람등 off
242         digitalWrite(LED_2, LOW);     // 소화기 위치 알람등 off
243         for(int i=0;i<4;i++){
244             strip_1.setPixelColor(i, 0, 0, 0);
245             strip_2.setPixelColor(i, 0, 0, 0);
246             strip_3.setPixelColor(i, 0, 0, 0);
247             strip_4.setPixelColor(i, 0, 0, 0);
248             strip_5.setPixelColor(i, 0, 0, 0);
249             strip_1.show();
250             strip_2.show();
251             strip_3.show();
252             strip_4.show();
253             strip_5.show();
254         }

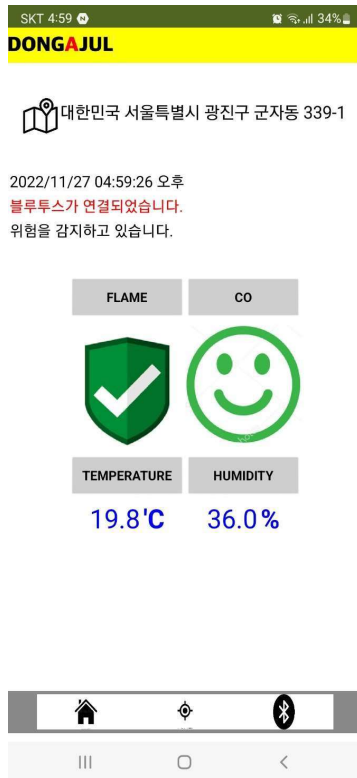
```

상황6)

일산화탄소 센서와 불꽃감지 센서 모두 작동하지 않는 경우 안전한 상황으로 판단하여 led 스트립의 동작을 멈추고 무드등의 색깔을 초록색으로 출력한다.

II -3. 어플리케이션

앱인벤터 프로그램을 사용하여 어플리케이션을 제작해 현재 상황을 직관적으로 확인할 수 있도록 하였다.



<어플리케이션 동작 사진>

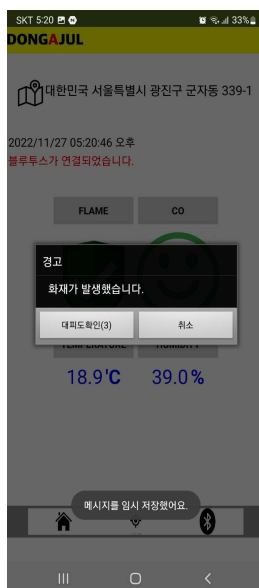
어플리케이션은 아두이노의 블루투스 모듈을 이용하여 데이터를 받아 다음과 같은 기능을 제공한다.

1. 실시간 gps 위치정보 및 현재 시간
2. 온습도 확인
3. 불꽃 감지 및 일산화탄소 감지 정도에 따른 경고이미지
4. 화재재난 시 스마트폰 진동 및 경고메시지 출력
5. 화재재난 시 현재 gps 정보와 구조 요청을 문자메시지에 담아 자동으로 소방서에 전달하는 시스템
6. 화재재난 시 화재가 발생한 위치에 따라 각기 다른 피난안내도 및 소화시설 확인

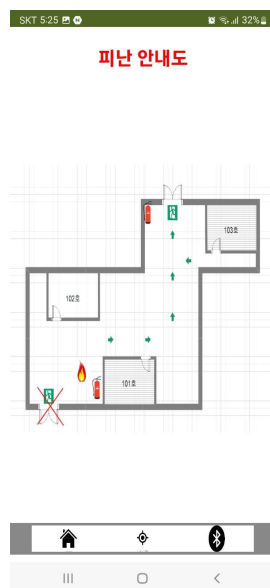


먼저 아두이노에서 받은 데이터들을 'input' 변수에 저장하고, 구분자인 ","를 사용하여 분리해 다시 'list' 변수에 배열의 형태로 저장한다.

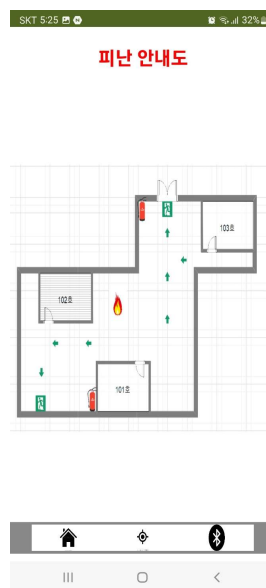
'list' 배열의 첫 번째 요소인 온도와 두 번째 요소인 습도는 각각 'label_temp'와 'label_humi'에 저장해 화면을 통해 소수점 한자리 형태로 출력하도록 하였다.



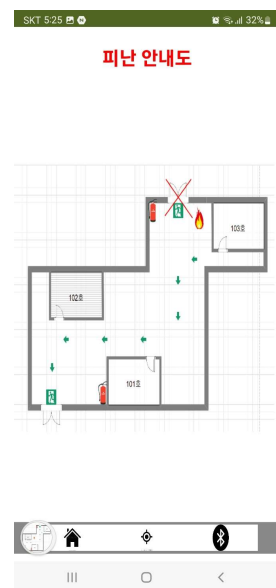
<화재발생 경고창>



< 상황1 피난안내도 >



< 상황2 피난안내도 >



< 상황3 피난안내도 >

배열 4번째 요소인 '현재 상황'을 6가지로 나누어 탄력적으로 대처할 수 있도록 블록을 구성하였다.

상황 1) 1번 구역에서 화재가 발생했을 경우 스마트폰이 진동하면서 화재발생 경고창이 떠 사용자에게 위험 상황을 인지시킨다. 대피도 확인 버튼을 누르면 '상황1 피난안내도'가 화면에 뜨고 위쪽으로 대피할 수 있다.



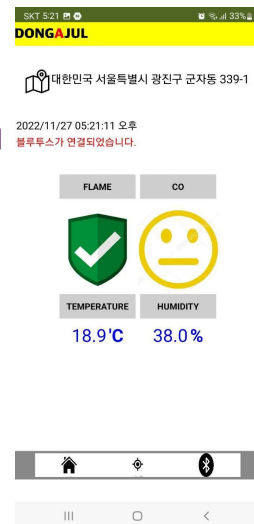
<상황 2 블록도>



<상황 3 블록도>



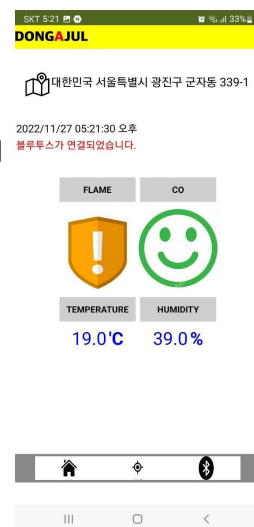
상황 2와 상황 3의 경우에도 마찬가지로 상황 1과 동일한 기능을 수행하고 대피도만 다르게 보이도록 하였다.



상황 4)

상황 4에서의 블록도와 어플리케이션 화면이다.

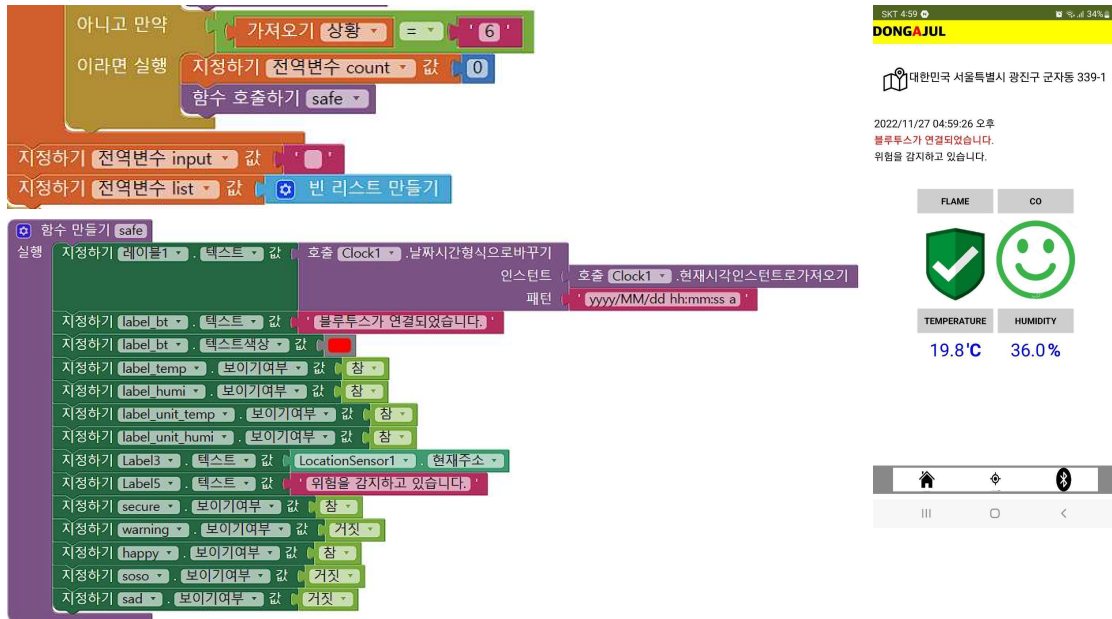
gas함수를 호출하여 화면의 일산화탄소 부분이 경고 이미지로 바뀌어 위험을 알린다.



상황 5)

상황 5에서의 블록도와 어플리케이션 화면이다.

flame 함수를 호출하여 화면의 불꽃감지 부분이 경고이미지로 바뀐다.



상황 6)

상황 6에서의 블록도와 어플리케이션 화면이다.

변수 'count', 'input', 'list' 의 값을 모두 초기화해주어 다시 불꽃과 일산화탄소 농도를 감지하고 위의 기능을 반복할 수 있도록 하였다.

Ⅲ. 결론

아두이노와 앱인벤터 프로그램을 활용하여 기존의 소리에 의존한 화재경보기와 차별성을 두어 청각장애인을 대상으로 한 화재경보기를 목표로 프로젝트를 진행하였다.

불꽃 감지 센서, 일산화탄소 센서를 사용해 화재를 인식하고, 진동모터와 led, led 스트립, 어플리케이션을 통해 청각장애인에게 화재 발생 위험 상황을 인지시켜 신속히 대피시키며 동시에 소방서에 자동으로 문자를 보내 큰 인명피해를 막을 수 있도록 설계하였다. 이 프로젝트에 사용된 기능을 바탕으로 부족한 부분을 개선해 본다면 인명피해를 줄이는 데 기여할 수 있을 것으로 기대한다.

[참고문헌]

I. 아두이노 #define

<https://arduinogetstarted.com/ko/reference/arduino-define>

II. 아두이노 led스트립

<https://m.blog.naver.com/boilmint7/221888306729>

III. 앱인벤터 문자전송

<https://sites.google.com/site/appinventorprogramming/examplemodule/munjamesiji-bonaegi?pli=1>

IV. 앱인벤터 블루투스

<https://ael.chungbuk.ac.kr/lectures/undergraduate/microwave-engineering/2017-1/lab/lab09-%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB%B8%94%EB%A3%A8%ED%88%AC%EC%8A%A4%ED%86%B5%EC%8B%A0.htm>