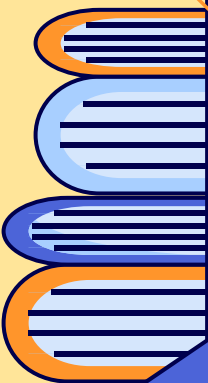
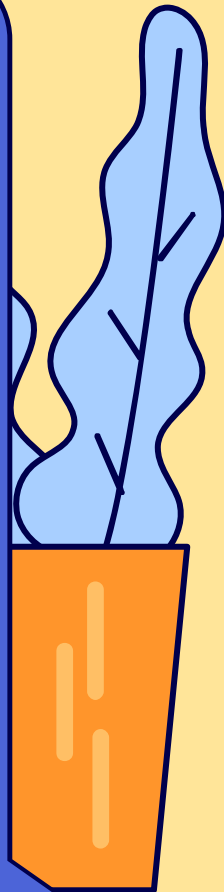


Pallang

**지기지기 백전백승
& 전공심화 공동체**



Contents

01

팀 소개

02

제작 동기 및 목적

03

결과물 제작 위한 공부

04

웹 제작 과정 & 시연

05

앱 제작 과정 & 시연

06

개선점

07


느낀점



01

팀 소개

못 먹어도 고! 21들의 우당탕탕 개발기



Pallang

21 유지원



21 김용현



21 이현성



21 노지민





02

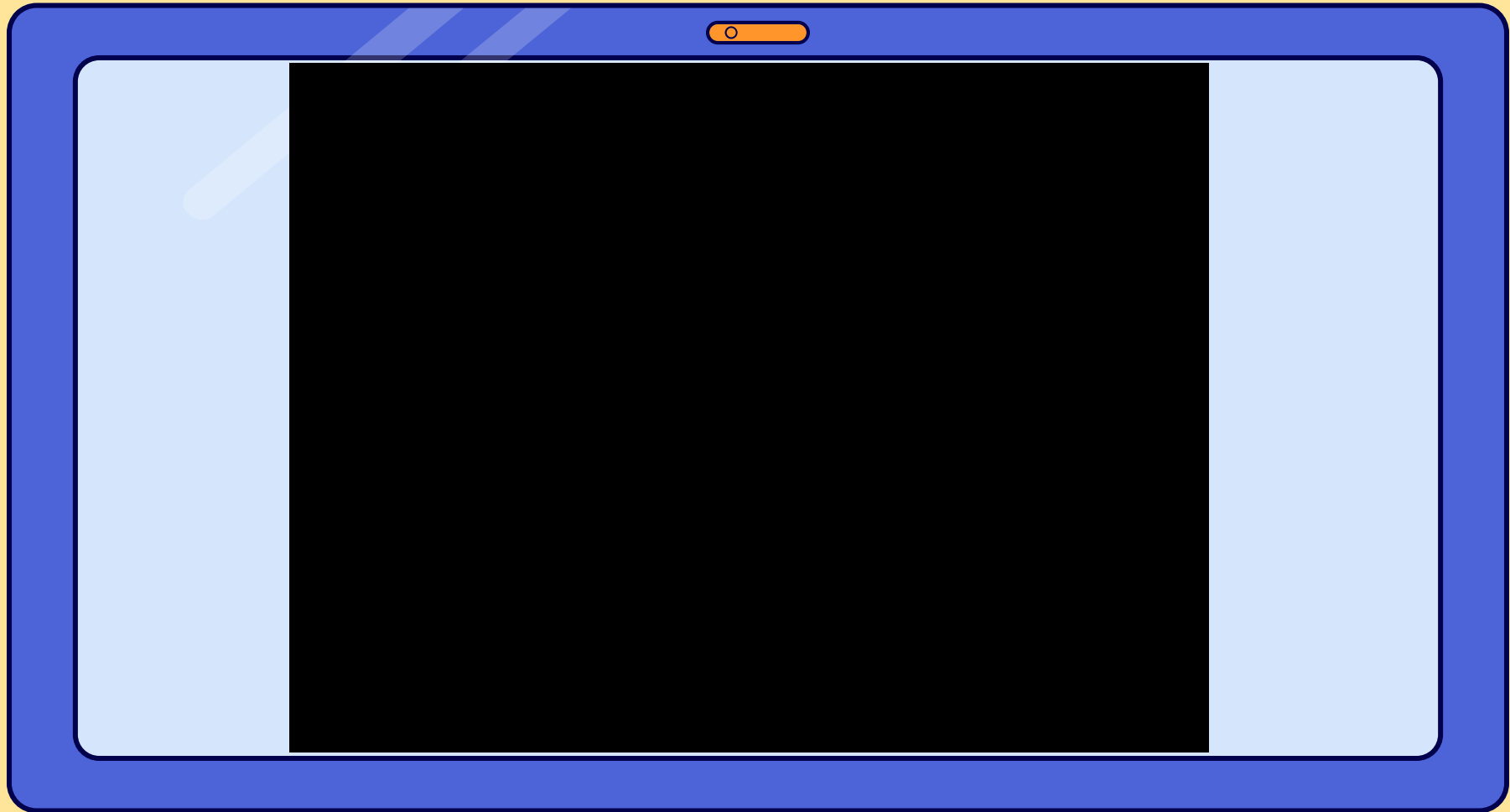
제작 동기 및 목적

지기전이 뭐야? 대단한 사람들이지~(하하;;)



- 👉 과를 위한 프로젝트를 하기로 결심
- 👉 지능기전의 비전: hard + soft -> 아두이노 + 웹/앱
- 👉 우리가 겪는 곤란함 생각 -> 과방 사용시간 겹칠 때
- 👉 과방 내의 인원을 알고 싶다

IoT 과방, 즉 스마트 과방을 만들자!





IoT 과방 관리 웹&앱; 자기 smart room

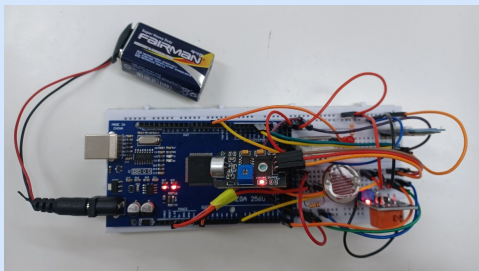
〈앱〉

보드1 - bluetooth

과방 안을 홈오토메이션으로

- 전등 상태
- 온습도
- 알콜
- 소음(데시벨)
- 날씨 예측

=> 무선 구축 및 결과 값 실시간 반영



〈웹〉

보드2 - wi-fi

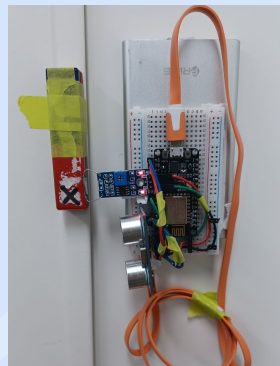
과방의 물리적 상태 확인

- 과방 출입 인원 수 확인
- 문 닫힘, 열림 여부 확인

1차 서버: 아두이노에서 웹 서버 open

2차 서버: wifi 웹서버 크롤링 후
가공한 웹사이트

=> 무선 구축 및 결과 값 실시간 반영





03

결과물 제작 위한 공부

이건 뭐고 저건 뭐? @_@



3D 모델링

- fusion 360

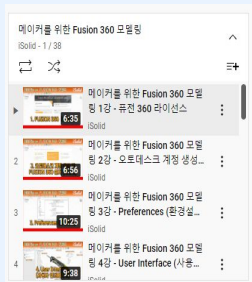
1 창의 SW CAD 강의

CAD 설계 입문 1강

창의SW융합노마드

강병현

2 유튜브 강좌

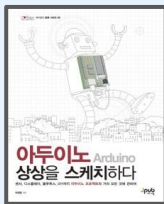


아두이노

1 WIFI ESP8266 MCU

모듈 스터디

- IP 할당 받기 위한 STA MODE 공부
- MCU-아두이노 센서와 결합 공부



2

아두이노
모듈 스터디

- 센서값 입출력

3 알고리즘 구축 스터디

- 사람 수 세기 알고리즘
- 날씨 예측 알고리즘

네트워크

전반적인 통신 공부 및 연구

1 아두이노-아두이노

- 블루투스
- 지그비

2 아두이노-컴퓨터

3 아두이노-스마트폰

- 블루투스

4 아두이노-인터넷

- wifi

웹 & 서버

프론트엔드 및 백엔드 공부

1 프론트엔드 & 백엔드

노마드 코더 강의



2 아두이노의 센서값을
어떻게 웹으로 가져오는지
연구

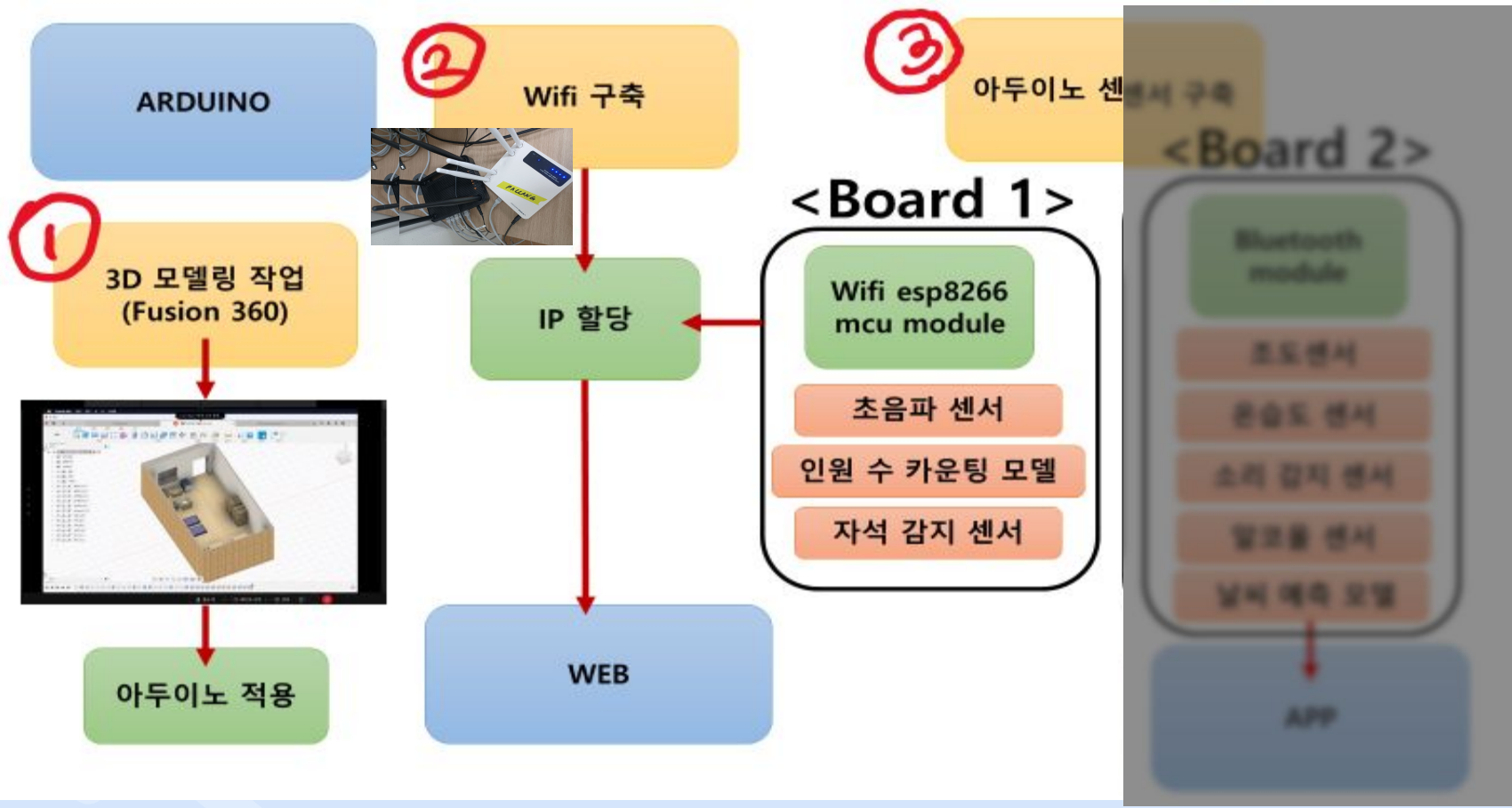


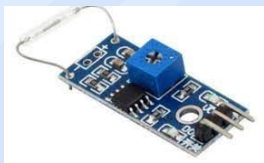
04

웹 제작 과정 & 시연

만땅에 헤딩만 2개월







자석 감지 센서
자기장 검출 여부 전송



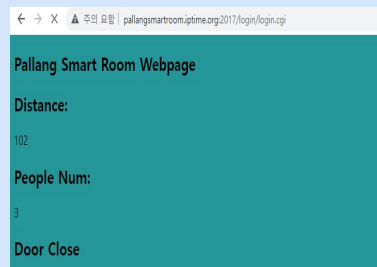
IP 할당

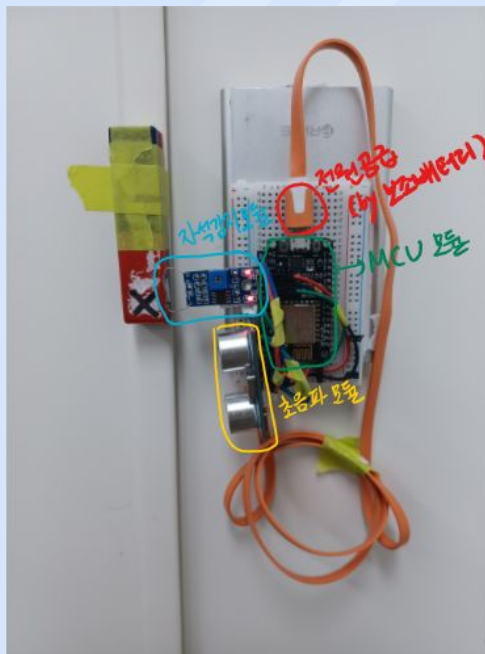


와이파이 Esp8266
MCU 모듈을 이용하여 IP
웹서버에 정보 전달



초음파 센서
초음파로 측정한 물체의 거리 값 전송





```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

/*Put your SSID & Password*/
const char* ssid = "pallang_ipitime"; // Enter SSID here
const char* password = "arduinoweb"; //Enter Password here
```

```
WiFi.mode(WIFI_STA);/
Serial.print("Got IP: "); Serial.println(WiFi.localIP());
```

```
int person_num = Function_Ultra(trig,echo);
if(person_num == 1){
  tot_person ++;
  Serial.print("Total Person : ");
  Serial.println(tot_person);
}
```

```
77 while(client.available()) {
78   client.read();
79 }
80
81   client.print("<!DOCTYPE HTML>");
```

```
if(digitalRead(magnet)== LOW){
  Serial.println("DOOR CLOSE");
  is_door = 0;
  delay(1000);
}else{
  Serial.println("DOOR OPEN");
  is_door = 1;
  delay(1000);
}
delay(500);
```

<ESP8266WiFi.h>
<ESP8266WebServer.h>
라이브러리 사용

STA 모드로 공유기에서 ip 할당 -> 1차 서버

초음파 센서로 인원 수 세는 알고리즘,
자석 감지 센서로 문 닫힘 알고리즘 적용

```
int Function_Ultra(int tr,int ec){

  int person_num = 0;
  digitalWrite(tr,LOW);
  delayMicroseconds(2);

  digitalWrite(tr,HIGH);

  delayMicroseconds(10);
  digitalWrite(tr,LOW);
  long uncalculation = pulseIn(ec,HIGH);
  long result = (uncalculation/2)/29.1;

  delay(1000);

  if(result < 65){
    return 1;
  //   delay(3000);
  } else return 0;
```

webserver에 결과 값을 html로 작성해 전송

← → × 주의 요함 | pallangsmartroom.iptime.org:2017/login/login.cgi

Pallang Smart Room Webpage

Distance:

102

People Num:

3

Door Close

1차 서버: WIFI 공유기에서 IP 할당 받아 open한 webserver

```
var cheerio = require("cheerio"); // 크롤링한 정보를 다듬어주는 모듈
var request = require("request"); // HTTP 네트워크 라이브러리

var url = "http://pallangsmartroom.iptime.org:2017/"; // 1차 서버 주소

export const home = function (req, res) {
  let abc = (err, person, door) => res.render("home", { person, door }); // 전달받은 정보를 서버에 보내기
  request(url, function (error, response, html) {
    if (error) {
      throw error;
    }
  });

  var $ = cheerio.load(html);

  var person = $("#per_n").text();
  var door = $("#is_door").text();

  abc(error, person, door);
}; // 정보를 추출하고 함수에 전달하기
```

지기 SMART-ROOM

과방 Status 과방 예약(오픈 예정) 아두이노 페이지

Reload

과방 Status

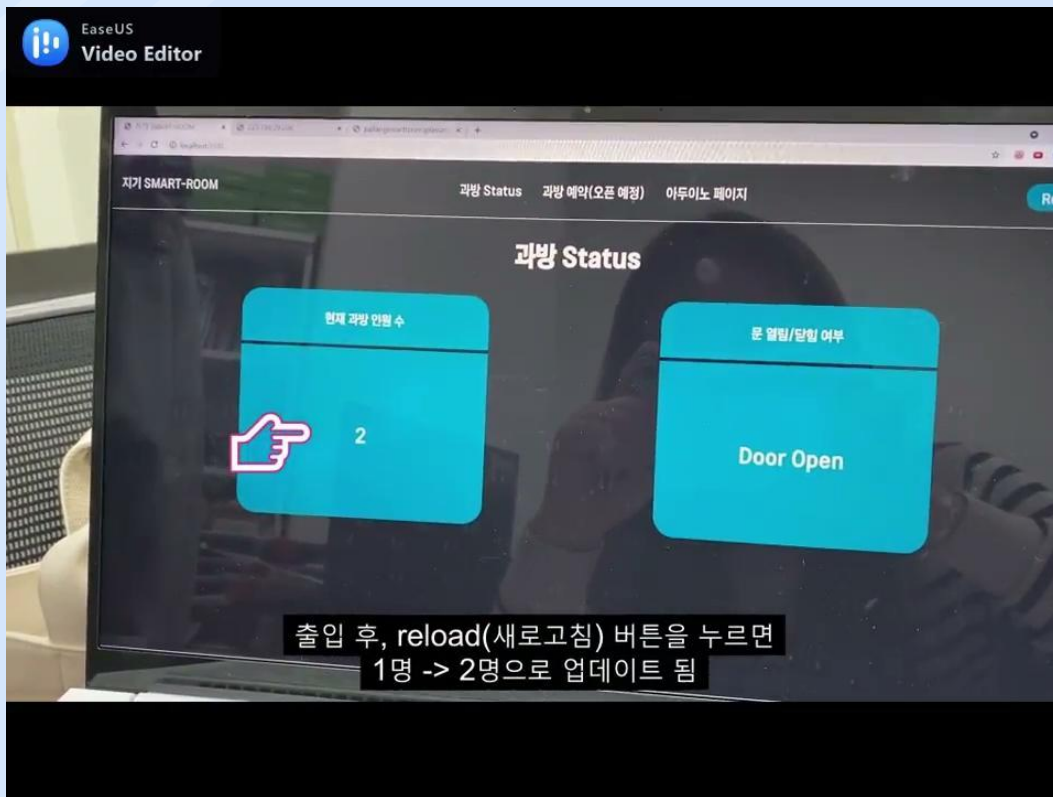
현재 과방 인원 수

2

문 열림/닫힘 여부

Door Close

2차 서버:
id/class로 정보 크롤링 해오고 nodeJS로 서버를 열어 운영





05

앱 제작 과정 & 시연

쇼미왓유갓





센서 구축

<Board 2>

Bluetooth module

조도센서

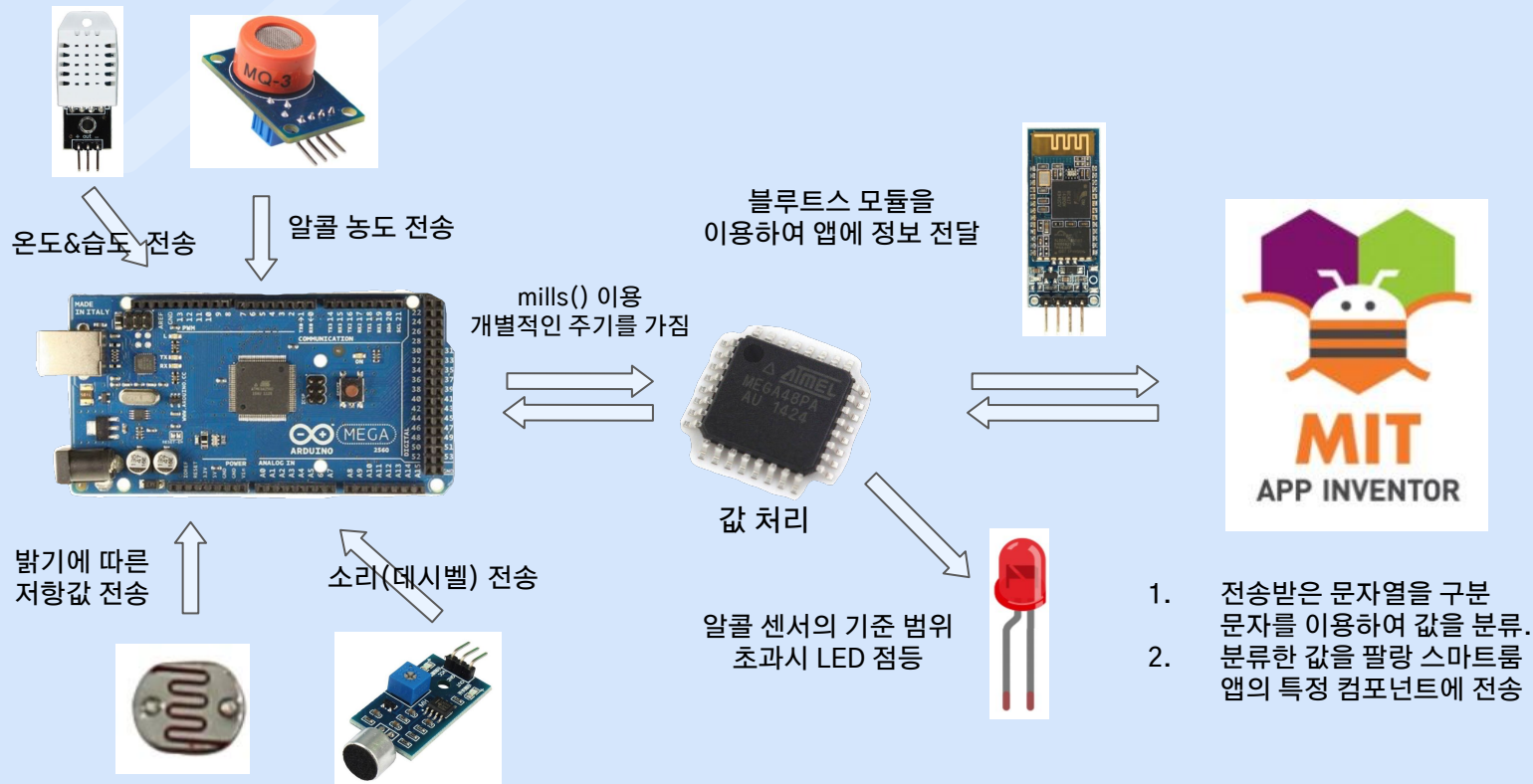
온습도 센서

소리 감지 센서

알코올 센서

날씨 예측 모델

APP



조명 상태

현재 온습도를
실시간으로 출력 ON

현재 조명상태를
실시간으로 표시
(켜짐: 노랑, 꺼짐: 회색)

습도

현재 현재 알콜 상태를
LED 형식으로 표현

average

현재 데시벨을 슬라이더로 표시

예상 날씨

습도값을 바탕으로 예상날씨 출력
연결하기 (4단계) 연결끊기

앱 작동 모습

```

1 //include chrono.h
2 //include cstdint.h
3
4 //소스 생성 함수
5 const int N = 100000000; // 코드 범위
6 int left_status = false, key = 0;
7
8 //소스 관리 함수
9 const int r = 10; //소스 개수
10 const int w = 10;
11 int test_person = 0; //시도할 사람
12
13 //소스 관리 함수
14 const int N2 = 100000000;
15 int test_person2 = 0;
16
17 //소스 관리 함수
18 int main() {
19     //test person
20     for (int i = 0; i < N; i++) {
21         int red = 0;
22         int red2 = 0;
23
24         //소스 생성
25         long long u1 = 0; //소스1을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
26         long long u2 = 1000000000; //소스2 생성
27
28         //소스 생성
29         long long u3 = 0; //소스3을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
30         long long u4 = 1000000000; //소스4 생성
31
32         long long interval = 0; //소스1 간격
33         long long interval2 = 0; //소스2 간격
34
35         long long u1_ac = 0; //소스1을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
36         long long u2_ac = 1000000000; //소스2 간격
37
38         //소스1을 생성
39         long long u3_ac = 0; //소스3을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
40         long long u4_ac = 1000000000; //소스4 생성
41
42         long long interval3 = 0; //소스3 간격
43         long long interval4 = 0; //소스4 간격
44     }
45
46     //test person2
47     for (int i = 0; i < N2; i++) {
48         int red = 0;
49         int red2 = 0;
50
51         //소스 생성
52         long long u1 = 0; //소스1을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
53         long long u2 = 1000000000; //소스2 생성
54
55         //소스 생성
56         long long u3 = 0; //소스3을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
57         long long u4 = 1000000000; //소스4 생성
58
59         long long interval = 0; //소스1 간격
60         long long interval2 = 0; //소스2 간격
61
62         long long u1_ac = 0; //소스1을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
63         long long u2_ac = 1000000000; //소스2 간격
64
65         //소스1을 생성
66         long long u3_ac = 0; //소스3을 생성하고 비교할 이전 시선, 주어진 시간간격 차에 시선 (종가)
67         long long u4_ac = 1000000000; //소스4 생성
68
69         long long interval3 = 0; //소스3 간격
70         long long interval4 = 0; //소스4 간격
71     }
72
73     return 0;
74 }

```

```

136 void Function_sd(){
137     /*소리감지 센서*/
138     unsigned long crmil_sd = millis();
139     if((crmil_sd - pwm_sd) ≥ interval_sd){
140         pwm_sd = crmil_sd;
141         sensorValue = analogRead(SOUND_SENSOR_PIN);
142         sensorValue /= 2;
143
144         // Serial1.print("*");
145         // Serial1.print(sensorValue);
146         // Serial1.print("°");
147     }
148
149 }

```

소리감지

소리감지

```

65 int Function_Light(int val){
66     unsigned long crml_Light = millis();
67     if(crml_Light - pos_Light > interval_Light){//현재
68         //상태가 crml_Light - pos_Light이면 시간 업데이트 - 시간간
69         //수출할 200ms 코드
70         if(val > 200){light_status = false;key = 1 ? 0 : 1;}
71         else {light_status = true;key = 1 ? 0 : 1;}
72     }
73     if(light_status = 1 & key = 0){
74         Serial.print("Light Status is ON");
75         Serial.print("ON");key = 1 ? 1 : return 1;
76     } else if(light_status = 0 & key = 1){
77         Serial.print("Light Status is OFF");
78         Serial.print("OFF");key = 0 ? return 0;
79     }
80     return return 3;
81 }
82 }
83 }

```

```

580  def __init__(self, model):
581      self.model = model
582      self.device = torch.device('cuda:0')
583      self.encoder = self.model.encoder.to(self.device)
584      self.decoder = self.model.decoder.to(self.device)
585      self.loss = torch.nn.Loss()
586      self.optimizer = optim.Adam(self.model.parameters())
587
588  def train(self):
589      for epoch in range(1, 100):
590          for i in range(0, len(self.train_loader)):
591              self.optimizer.zero_grad()
592              data, target = self.train_loader[i]
593              data = data.to(self.device)
594              target = target.to(self.device)
595              output = self.model(data)
596              loss = self.loss(output, target)
597              loss.backward()
598              self.optimizer.step()
599          print('Epoch: %d, Loss: %f' % (epoch, loss.item()))
600
601  def test(self):
602      for i in range(0, len(self.test_loader)):
603          data, target = self.test_loader[i]
604          data = data.to(self.device)
605          target = target.to(self.device)
606          output = self.model(data)
607          loss = self.loss(output, target)
608          print('Test Loss: %f' % loss.item())
609
610  def main():
611      model = MyModel()
612      trainer = MyTrainer(model)
613      trainer.train()
614      trainer.test()
615
616  if __name__ == '__main__':
617      main()

```

```

87 void Function Alcohol()
88 {
89     unsigned long crml_ac = millis();
90     if((crml_ac - psw_ac) > interval_ac){//현재 파일값과 이전 시간값과의 차이가 시간 간격
91         crml_ac = crml_ac; //이제 시시 업데이트 // 시간값의 동기화
92         psw_ac = analogRead(PIN);
93         if(val > 0) Serial.println("Ac1");
94         else if (val > 200) Serial.println("Ac12");
95         else if (val > 400){
96             digitalWrite(res,HIGH);
97             Serial.println("Ac13");
98         }
99         Serial.println("recent Alcohol:");
100         Serial.println(val);
101     }
102 }

```

조도센서

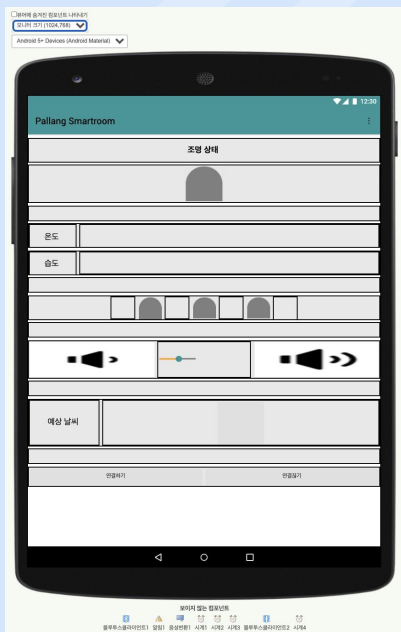
온습도

입출력 정의
통신속도 정의

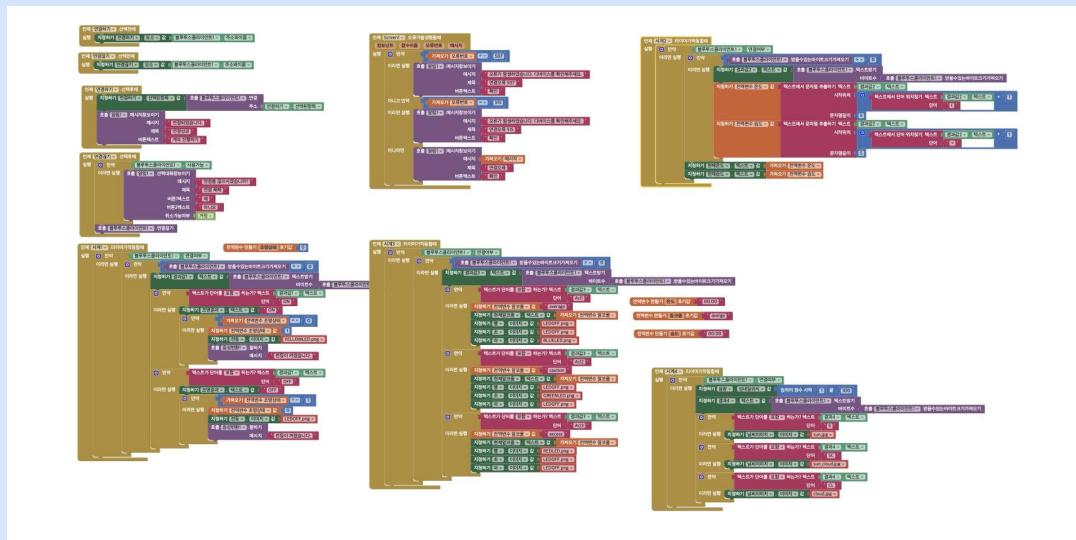
함수실행

알콜 농도 기준을 3가지(average:파랑,caution:초록,excess:빨강)로 분류하여 점등상태를 바꾼다.

기준 초과 시 앱 내에서 빨간색의 점등과 함께
아두이노에서 빨간 LED가 점등됨.



앱 디스플레이 로직



앱 프로그래밍(스크레치) 로직





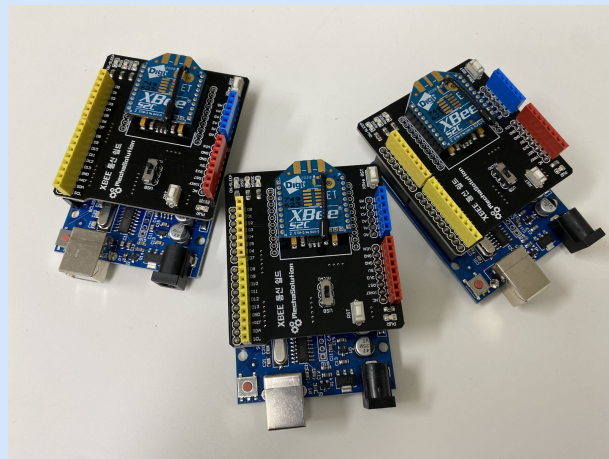
06

개선점

우린 아직도 배가 고프다!



- ① 여러 노드(보드) 간 지그비 통신
- ② 인원 수 카운팅 알고리즘 개선
- ③ 웹사이트/앱 외부 접속
- ④ 과방 예약 시스템 개발





07

느낀점

세상은 넓고 공부할 건 많다.





유지원

주로 통신과 네트워크 작업을 맡아 한 발 앞서 프로젝트의 전반적인 흐름을 구상해야한다는 부담감이 있었는데 실제로 성공해내니 너무나도 신기하고 재미있다.

통신에 대해 본격적으로 깊이 들어가보고 싶다는 생각이 들어 진로 활동에도 변화를 주게 되었을 만큼 의미있는 활동이었다.

팀 프로젝트의 팀장 자리에 대해서도 많이 고민하고 성장할 수 있는 시간이었다.



김용현

평소 웹으로 프로젝트를 만들거나 공부를 하면 하드웨어(아두이노)와는 거리가 멀다고만 생각했다.

하지만 이번 지기지기를 통해 아두이노와 웹의 통신을 공부할 수 있었고, 하드웨어와 소프트웨어의 융합을 경험할 수 있는 좋은 기회를 얻었다.



이현성

앱 제작에서 많은 실패와 오류를 거치면서 개발자가 다시 한번 대단하게 느껴졌다. 또한 아두이노간의 통신에서 웹 서버의 중요성을 느꼈고 깊이 배워보고싶다는 생각을 하게되었다.

혼자서 하기 힘들었을 이 프로젝트를 팀원들과 함께 하면서 서로 모르는 것을 공유하고 알아가며 많은 것을 배우게된 뜻 깊은 시간이었다.



노지민

구글링의 위대함을 몸소 느낄 수 있었다. 또 끈기와 도전의 중요성도.....ㅎ

아두이노 무선 통신을 직접 와이파이 공유기 설치 작업부터 구축해 나가면서 통신에 대한 이해를 재밌게 또 확실히 할 수 있었다.

아두이노로 IoT 서비스를 구현할 수 있다는 것에 놀랐다. 또 하드웨어와 소프트웨어의 융합이 중요하다는 것을 느낄 수 있었다.



감사합니다!

