

범죄 예방을 위한 미허가 출입 탐지/대응 모델: 중앙대학교 310관 적용 예시 – 중간 발표

김영빈, 조현우, 최경식



CONTENTS

01

프로세스 전반 설명

02

크롤링

03

데이터 정제

04

Arduino

05

최종 발표까지 수행 목표





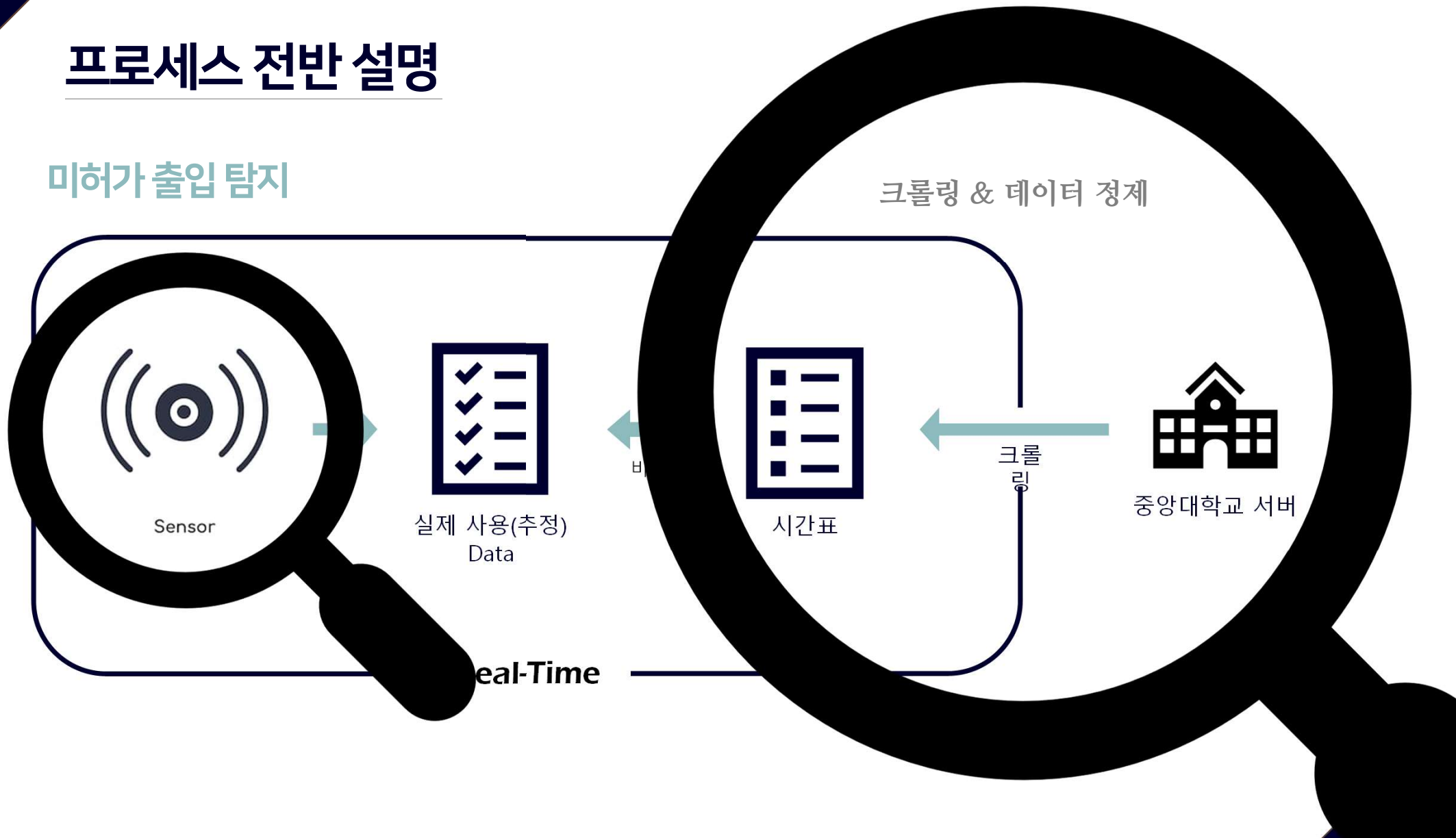
01

프로세스 전반 설명



프로세스 전반 설명

미허가 출입 탐지



02

중앙대학교 사이트 크롤링

크롤링

1. 강의실 정보

서울캠퍼스 안성캠퍼스

전체

검색어를 입력하세요.



<중앙대학교 캠퍼스 안내>



구분	건물명	호실	호실명
강의실	310관(100주년기념관)	B603	대형강의실
강의실	310관(100주년기념관)	B602	대형강의실
강의실	310관(100주년기념관)	B601	대형강의실
강의실	310관(100주년기념관)	B502	대형강의실
강의실	310관(100주년기념관)	B501	대형강의실
강의실	310관(100주년기념관)	B309	전자전기공학부 PC강의실
강의실	310관(100주년기념관)	305	강의실
강의실	310관(100주년기념관)	310	강의실
강의실	310관(100주년기념관)	311	강의실
강의실	310관(100주년기념관)	312	강의실
강의실	310관(100주년기념관)	315	강의실
강의실	310관(100주년기념관)	316	강의실

<310관 검색시>

크롤링

2. 강의 시간표 정보

2-1. Selenium 활용 시도 (현 포탈)

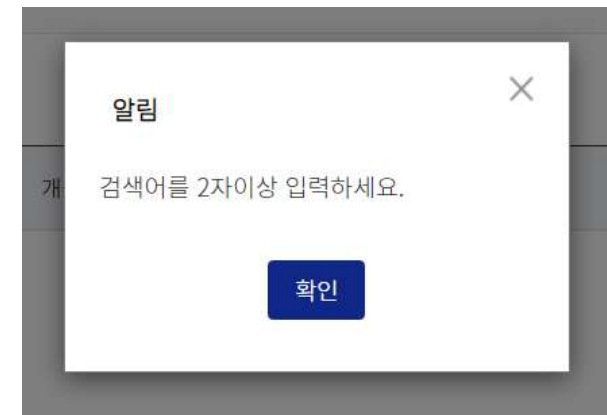
강의계획서검색

검색년도 2020 학기 1 검색대상 ☒ 과목명 ☐ 교수명 검색조건 조회

캠퍼스	과정	과목번호	과목명	개설학과	이수구분	대표강사	강의실/강의시간
-----	----	------	-----	------	------	------	----------

현재 조회된 데이터가 없습니다.

<강의계획서 검색 탭>



<검색 조건>

크롤링

2. 강의 시간표 정보

2-2. Request 활용 시도 (현 포탈)

```
▼ Request Payload    view source
▼ {year: "2020", shtm: "1", choice: "sbjt", searchnm: "사이버"}
  year: "2020"
  shtm: "1"
  choice: "sbjt"
  searchnm: "사이버"
```

<현 포탈 Request Body>

```
▼ General
Request URL: https://mportal.cau.ac.kr/std/usk/sUskSif002/selectList.ajax
Request Method: POST
Status Code: 200 OK
```

<현 포탈 Request URL>

세션 유지
&
헤더값??

크롤링

2. 강의 시간표 정보

2-3. 구 포탈 이용

campus.cau.ac.kr 내용:
검색조건을 입력하세요.

확인

강의계획서 조회

검색하기
검색대상
검색조건

<검색조건을 입력하지 않았을 때>

campus.cau.ac.kr 내용:
검색조건은 2글자 이상 입력하셔야 합니다.

확인

강의계획서 조회

검색하기
검색대상
검색조건

<검색조건을 한 자리만 입력했을 때>

▼ Form Data view source view URL encoded

year: 2020
shtm: 1
choice: sbjt
q_txt: English

<구 포탈 Request Body>

크롤링

2-4. 파이썬 코드

```
5 class Scraper():
6     def __init__(self):
7         self.url = "https://campus.cau.ac.kr/servlet/UskLecPl10"
8         self.filename = ""
9
10    def setRequestBody(self, year, semester, searchType):
11        if searchType == "subject":
12            searchType = "sbjt"
13        elif searchType == "professor":
14            searchType = "prof"
15        else :
16            searchType = None
17
18        #name = name.encode("euc-kr")
19
20        formData = {
21            "year" : year,
22            "shtm" : semester,
23            "choice" : searchType
24        }
25        return formData
26
27    def getHTML(self, formData) :
28        headers = {'Content-Type': 'application/x-www-form-urlencoded'}
29        res = requests.post(self.url, data = formData, headers = headers)
30
31        if res.status_code != 200:
32            print("Request Error :", res.status_code)
33
34        html = res.text
35        return BeautifulSoup(html, "html.parser")
```

```
37 def getTimeTable(self, soup, buildingName):
38     timeTables = soup.select("table > tr")
39
40     campus = []
41     sbjtName = []
42     major = []
43     prof = []
44     room_time = []
45
46     for i in range(1, len(timeTables)):
47         data = timeTables[i].find_all("td")
48         if buildingName in data[7].text:
49             campus.append(data[0].text.strip())
50             sbjtName.append(data[3].text.strip())
51             major.append(data[4].text.strip())
52             prof.append(data[6].text.strip())
53             room_time.append(data[7].text.strip())
54
55     self.writeCSV(campus, sbjtName, major, prof, room_time)
56
57 def writeCSV(self, campus, sbjtName, major, prof, room_time) :
58     file = open(self.filename, "a", newline="")
59
60     wr = csv.writer(file)
61     for i in range(len(campus)) :
62         wr.writerow([str(i + 1), campus[i], sbjtName[i], major[i], prof[i], room_time[i]])
63
64     file.close()
65
66 def scrap(self, year, semester, searchType, buildingName):
67     formData = self.setRequestBody(year, semester, searchType)
68     soupPage = self.getHTML(formData)
69
70     self.filename = "timeTable_" + searchType + "_" + buildingName + ".csv"
71     file = open(self.filename, "w", newline="")
72     wr = csv.writer(file)
73     wr.writerow(["No.", "월페스", "과목명", "개설학과", "대표강사", "강의실 / 강의시간"])
74     file.close()
75
76     self.getTimeTable(soupPage, buildingName)
```



03

데이터 정제



데이터 정제

1. 데이터 수정

1-1. 건물과 호실의 구분

```
{"campus": "서울", "course": "학부", "code": "52485 - 01", "sbj": "기계진동", "faculty": "공과대학기계공학부", "major": "전공",
```

```
"professor": "남우철", "lecture": "310관 514호 <강의실> / 520호 <강의실> 월5 / 수5,6"},
```



```
{"campus": "310관", "sbj": "기계진동", "lecture": "514호 / 520호 <강의실> 월5 / 수5,6"},
```

1-2. 강의실을 두 개 이상 사용하는 경우

```
{"campus": "310관", "sbj": "기계진동", "lecture": "514호 / 520호 <강의실> 월5 / 수5,6"},
```



514호 <강의실> 월 5

520호 <강의실> 수 5, 6

데이터 정제

1. 데이터 수정

1-3. 시간 형식 데이터와 교시 형식 데이터

```
"lecture": "310관 921호 <강의실> 월(10:30~11:45) / 수(10:30~11:45)",  
"lecture": "310관 921호 <강의실> 금7,8,9"},|
```

따로 처리하는 방법 사용!

1-4. <강의실>이 없는 경우

```
{  
  "campus": "서울",  
  "course": "학부",  
  "code": "49950 - 14",  
  "sbj": " ACT",  
  "faculty": "대학(전체)",  
  "major": "교양",  
  "professor": "도선재",  
  "lectureRoom": "310관 718호 <강의실>",  
  "lectureTime": "화7,8"  
},
```

```
{  
  "campus": "서울",  
  "course": "학부",  
  "code": "49950 - 15",  
  "sbj": " ACT",  
  "faculty": "대학(전체)",  
  "major": "교양",  
  "professor": "허연정",  
  "lectureRoom": "310관 804호",  
  "lectureTime": "(수1,2)"  
},
```



04

Arduino



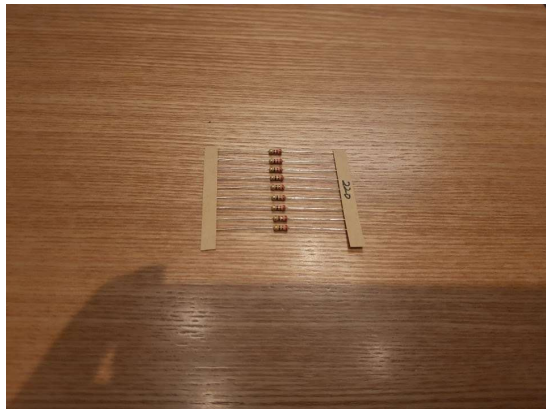
Arduino

1. 구매 물품

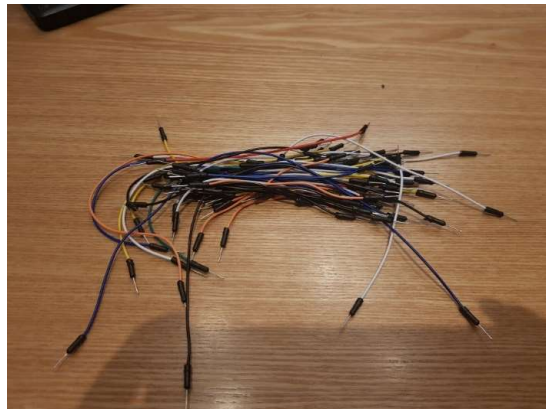
우노 WIFI D1 R2 보드	안드로이드 5핀 젠더
브레드 보드	AA 전지
점퍼 케이블	저항 220 옴
LED 전구	저항 110 옴
1.5V AA 6칸 배터리 홀더	소형 적외선 PIR 인체감지 모션센서



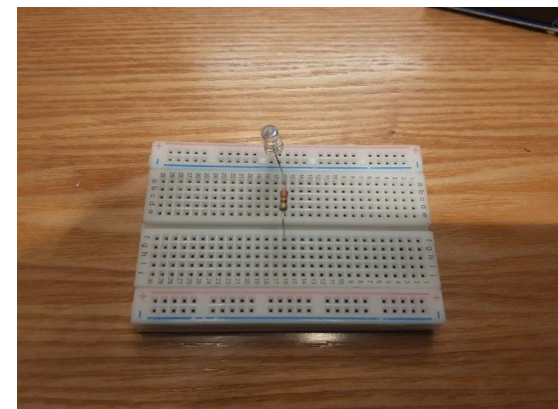
<WeMos D1 R2 Board>



<220옴 저항>



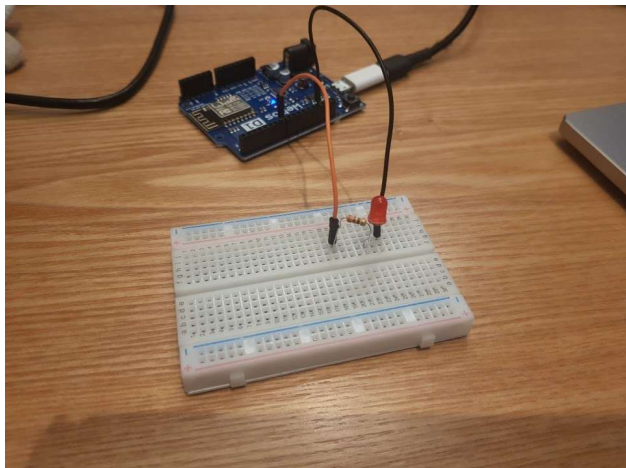
<점퍼 케이블>



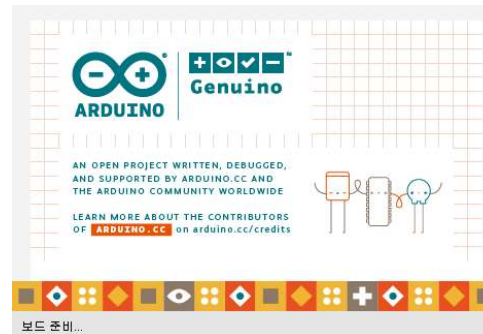
<브레드보드 + LED전구 + 저항>

Arduino

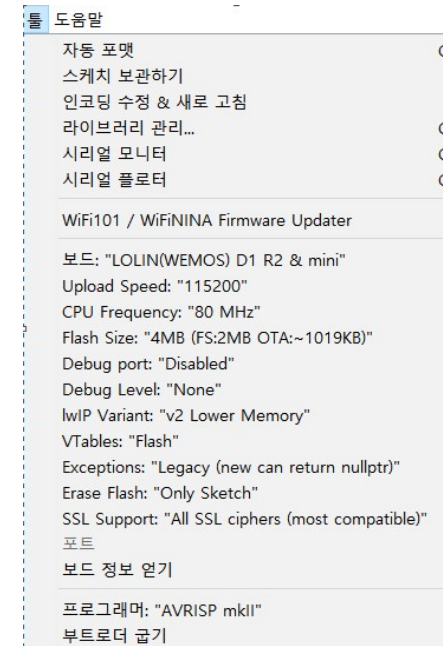
2-1. 환경 구축



<환경 구축과 테스트를 위해 노트북과 연결한 모습>



<아두이노 IDE>



<아두이노 IDE내 WeMos 보드 설정값>

보드	LOLIN(WEMOS) D1 R2 & mini
Upload Speed	115200 bps
CPU Frequency	80 MHz
VTables	Flash
Flash Size	4MB

<WeMos 보드 설정값 요약>

Arduino

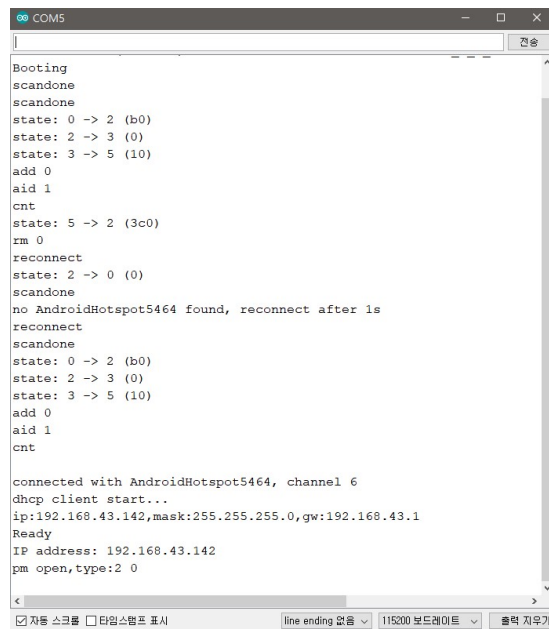
2-2. 네트워크에 무선 포트 오픈, 연결

```
#define STASSID "your-ssid"
#define STAPSK "your-password"
#endif
```

```
const char* ssid = "AndroidHotspot5464";
const char* password = " ";
```

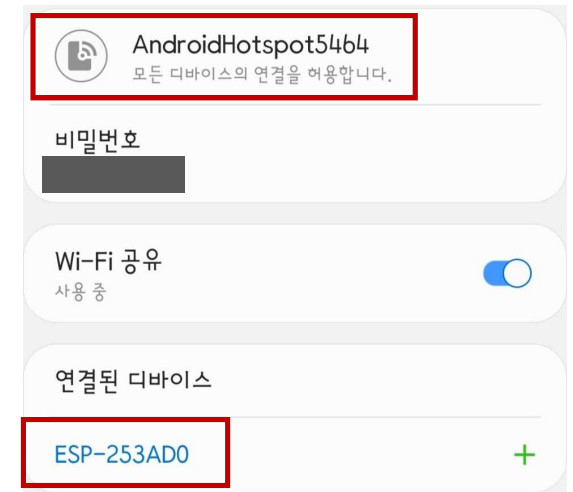
```
void setup() {
  Serial.begin(115200);
  Serial.println("Booting");
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.println("Connection Failed! Rebooting...");
    delay(5000);
    ESP.restart();
  }
}
```

<OTA 코드의 일부>



```
COM5
Booting
scandone
scandone
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0
aid 1
cnt
state: 5 -> 2 (3c0)
rm 0
reconnect
state: 2 -> 0 (0)
scandone
no AndroidHotspot5464 found, reconnect after 1s
reconnect
scandone
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0
aid 1
cnt
connected with AndroidHotspot5464, channel 6
dhcp client start...
ip:192.168.43.142,mask:255.255.255.0,gw:192.168.43.1
Ready
IP address: 192.168.43.142
pm open,type:2 0
```

<OTA코드 실행 후 아두이노 IDE 시리얼 모니터>



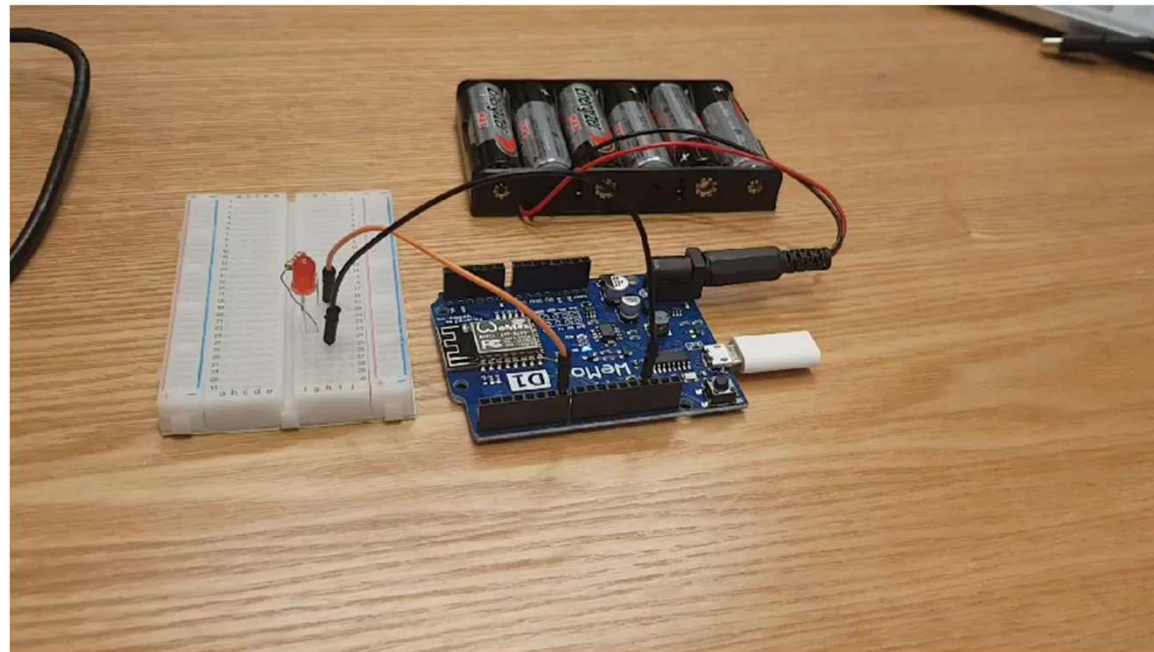
<모바일 기기에서 아두이노 보드 연결 확인>

Arduino

2-3. 결과 확인

```
void setup() {  
  pinMode(D5, OUTPUT);  
}  
void loop() {  
  digitalWrite(D5, HIGH);  
  delay(1000);  
  digitalWrite(D5, LOW);  
  delay(1000);  
}
```

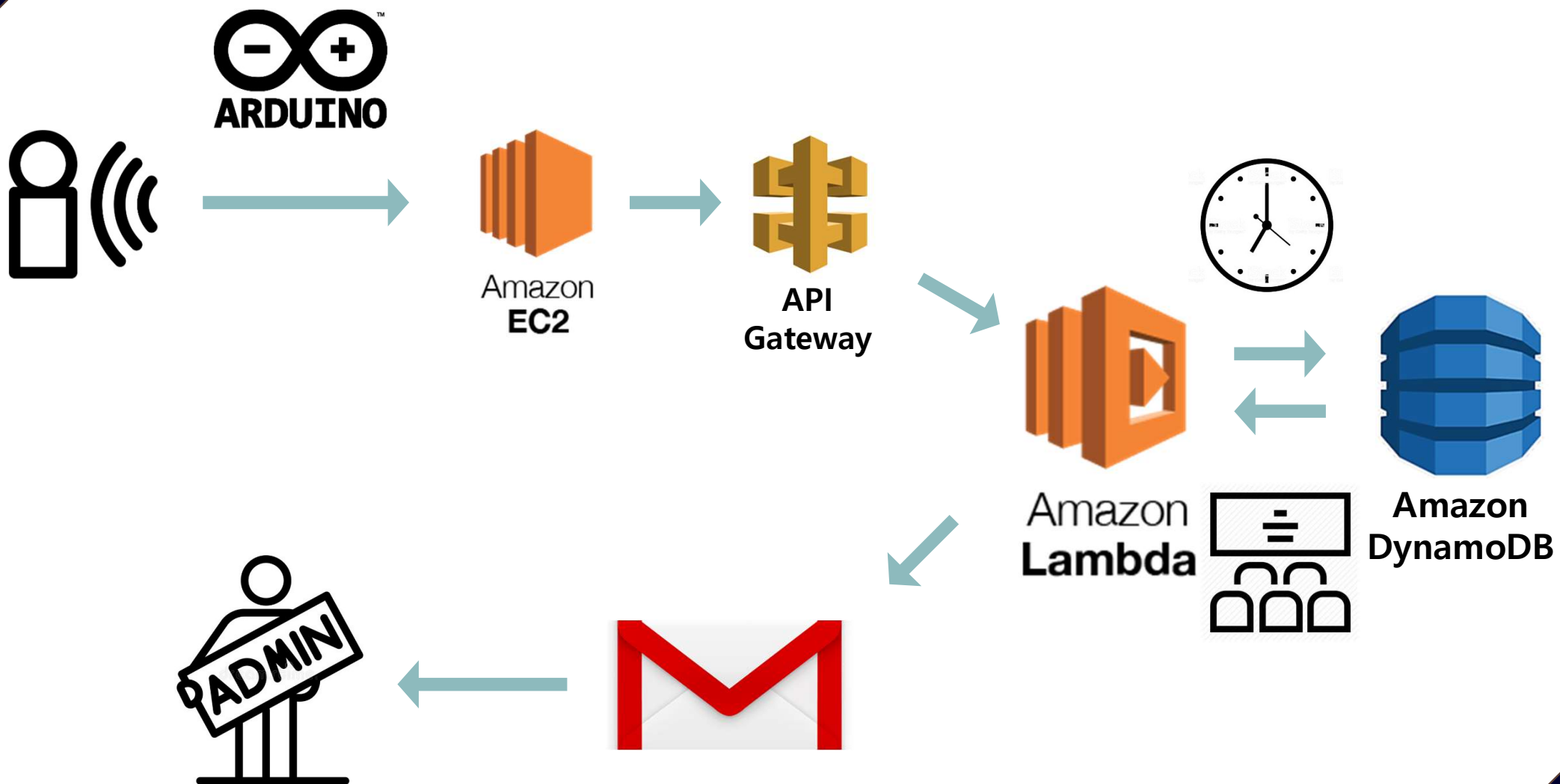
<테스트를 위한 함수>



<실행 성공 영상>

05

최종 발표까지 수행 목표





Thank you

Questions & Comments

