

123

인원 수에 따른

## 지하철 냉방 조절 시스템

IoT와임베디드소프트웨어 - 최종발표  
한다현, 이은주, 이은진, 진서연

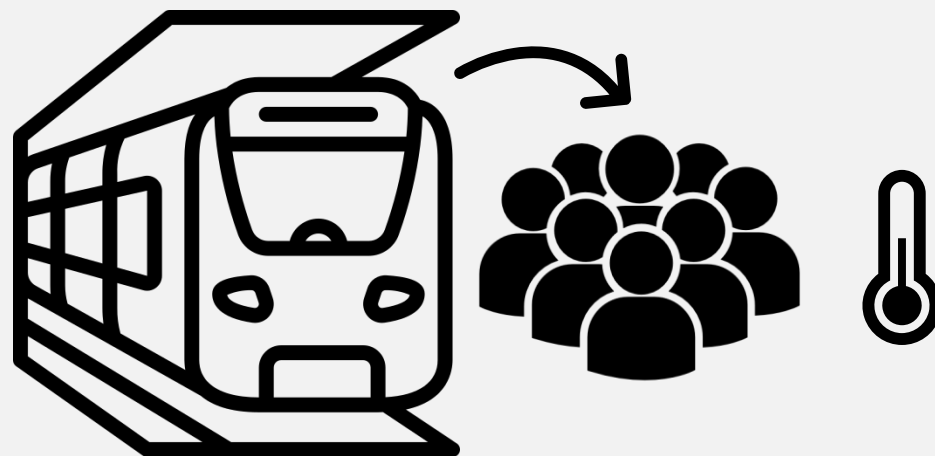
## 1

## 프로젝트 주제 및 목적

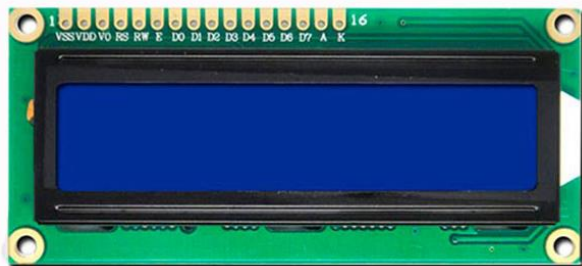


지하철 내의 **승객 분포**에 따른 실내 온도 조절  
⇒ 쾌적한 환경 제공 및 에너지 절약

승객에게 현재 여석 현황 및 내부 온도 정보 제공  
⇒ 대중 교통 사용 시 편의 제공



## 2 사용 부품



16x2 I2C LCD



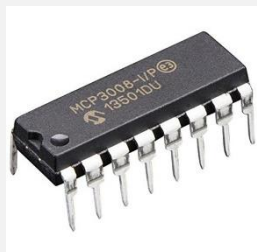
RGB LED



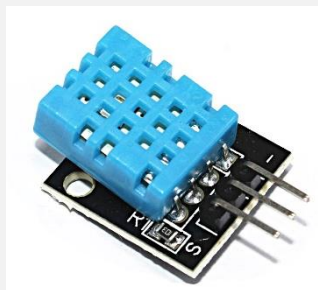
펄티어 열전 소자 쿨러



조도센서



MCP3008



DHT11 - 온습도 센서



릴레이 모듈



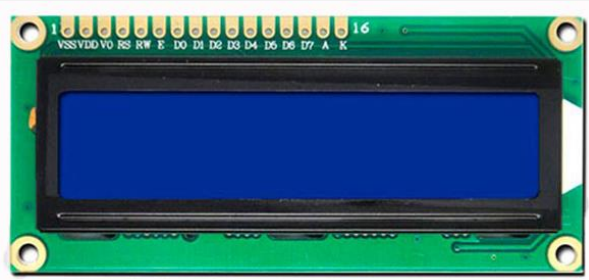
12V 6A 어댑터

3

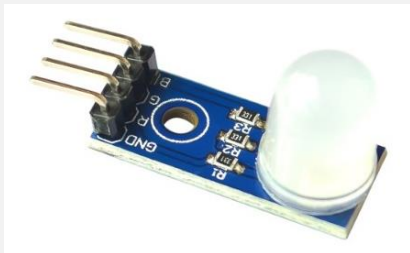
구현 기능

1) LED와 LCD로 현재 지하철 내부 상태 출력

## 출력 부품



16x2 I2C LCD



RGB LED

## 구현 기능

- **LCD**에는 현재 지하철 칸의 **온도**와 **탑승 승객 수**를 출력함
- **RGB LED**는 탑승 승객의 **포화도에 따른 빛의 색상**을 출력함

좌석 5개 기준) 4인 이하 착석 - 초록 불  
5인 착석 - 빨간 불

## 3

## 구현 기능

## 1) LED와 LCD로 현재 지하철 내부 상태 출력

```
// LCD에 온도와 사람 수를 출력하는 함수
void printLCD(int* temperature, int people) {
    fd = wiringPiI2CSetup(I2C_ADDR); //I2C 통신 시작

    lcd_init(); // LCD 설정 초기화
    clrLcd(); // LCD 초기화

    lcdLoc(LINE1); // LCD 첫번째 줄
    typeIn("now : "); // "now: " 출력
    typeInt(temperature[0]); // 온도 정수값 출력
    typeIn("."); // "." 출력
    typeInt(temperature[1]); // 온도 소숫값 출력
    typeIn("C"); // "C" 출력

    lcdLoc(LINE2); // LCD 두번째 줄
    typeIn("people : "); // "people: " 출력
    typeInt(people); // 사람 수 출력
    typeIn("/5"); // "/5" 출력
}
```

```
void TurnLED(int people) { // LED를 키는 함수
    ledInit(); // LED 설정 초기화
    if (people >= 5) { // 승객이 5명 이상이라면
        ledInit();
        ledColorSet(0xff, 0x00, 0x00); // Red 켜기
    }
    else { // 4명 이하라면
        ledInit();
        ledColorSet(0x00, 0xff, 0x00); // Green 켜기
    }
}

// LED 초기화 함수
void ledInit(void)
{
    softPwmCreate(LedPinRed, 0, 100);
    softPwmCreate(LedPinGreen, 0, 100);
}

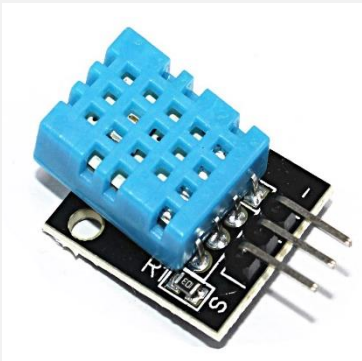
// LED 색 설정 함수
void ledColorSet(uchar r_val, uchar g_val, uchar b_val)
{
    softPwmWrite(LedPinRed, r_val);
    softPwmWrite(LedPinGreen, g_val);
}
```

3

구현 기능

2) DHT11 센서로 현재 온도 측정

입력 부품



DHT11 - 온습도 센서

구현 기능

- DHT11 센서를 사용하여 현재 지하철 칸의 내부 온도를 측정함
- 측정된 값은 LCD에 출력됨

## 3

## 구현 기능

## 2) DHT11 센서로 현재 온도 측정

```
void MeasureTemperature() // 온도 측정 함수
{
    uint8_t laststate = HIGH;
    uint8_t counter = 0;
    uint8_t j = 0, i;

    dhtVal[0] = dhtVal[1] = dhtVal[2] = dhtVal[3] = dhtVal[4] = 0;

    // MCU 18ms 동안 LOW signal 보냄
    pinMode(DHTPIN, OUTPUT);
    digitalWrite(DHTPIN, LOW); //pull down
    delay(18); // dealy

    //40micros 동안 HIGH signal 보냄
    digitalWrite(DHTPIN, HIGH); //pull up
    delayMicroseconds(40); // deay

    //DHT가 signal 받음
    pinMode(DHTPIN, INPUT);
```

```
// DHT가 MCU에 data 보내는 과정
for (i = 0; i < MAXTIMINGS; i++){
    counter = 0;
    while (digitalRead(DHTPIN) == laststate){
        counter++;
        delayMicroseconds(1);
        if (counter == 255)
            break;
    }
    laststate = digitalRead(DHTPIN);
    if (counter == 255)
        break;
    // 처음 3개 변환 무시
    if ((i >= 4) && (i % 2 == 0))
    {
        //스토리지 바이트에 각 비트 저장
        dhtVal[j / 8] <= 1;
        if (counter > 16)
            dhtVal[j / 8] |= 1;

        j++;
    }
}
return;
}
```

3

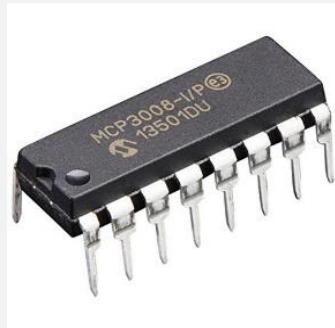
구현 기능

3) 조도 센서와 MCP3008로 현재 승객 수 측정

## 입력 부품



조도센서



MCP3008

## 구현 기능

- 조도 센서와 MCP3008를 사용하여 현재 지하철 칸의 **탑승 승객 수를 측정**함
- 이때 읽어온 **빛의 값이 20이하**인 경우, 해당 좌석은 **착석된 것으로** 감지함
  - => 약 8초 간격으로, 모든 좌석에 부착된 조도센서 값을 읽어서 착석이 감지되면 승객 수를 카운트하여 현재 지하철 내의 승객 수를 파악함



## 3

구현 기능

## 3) 조도 센서와 MCP3008로 현재 승객 수 측정

```
void People() { // 사람 측정 함수
people = 0; //사람 수 정보 초기화

int val1 = analogRead(BASE + SPI_CHAN1); // 1번째 조도
센서 값 읽기
int val2 = analogRead(BASE + SPI_CHAN2); // 2번째 조도
센서 값 읽기
int val3 = analogRead(BASE + SPI_CHAN3); // 3번째 조도
센서 값 읽기
int val4 = analogRead(BASE + SPI_CHAN4); // 4번째 조도
센서 값 읽기
int val5 = analogRead(BASE + SPI_CHAN5); // 5번째 조도
센서 값 읽기

// 각 조도 센서의 사람 유무 판단
detectPeople(val1);
detectPeople(val2);
detectPeople(val3);
detectPeople(val4);
detectPeople(val5);
}
```

```
void detectPeople(int val) { // 조도 센서의 사람 유무 판단
if (val <= 20) { //밝기가 20이하이면 사람 있는 것으로 감지
people = people + 1;
}
}
```

3

구현 기능  
4) 인원수에 따른 쿨러 조절

## 출력 부품



펄티어 열전 소자 쿨러



릴레이 모듈



12V 6A 어댑터

## 구현 기능

- 릴레이 모듈을 활용하여  
현재 탑승객 수에 따라 쿨러 작동을 조절함

좌석 5개 기준) 0명 착석 - 쿨러 off  
1명~4명 착석 - 쿨러 on/off 반복하여 온도 유지  
5명 착석 - 쿨러 on

## 3

구현 기능

## 4) 인원수에 따른 쿨러 조절

```
void AirconMaintain() { // 열전소자 쿨러 모듈을 켜고,  
    끄기를 반복하는 함수  
    pinMode(REL, OUTPUT);
```

```
    digitalWrite(REL, LOW); // 4초 동안 끄기  
    delay(4000);
```

```
    digitalWrite(REL, HIGH); // 3초 동안 켜기  
    delay(3000);  
}
```

```
void AirconTurnoff() { // 열전소자 쿨러 모듈을 끄는  
    함수  
    pinMode(REL, OUTPUT);  
    digitalWrite(REL, LOW); // 끄기
```

```
    return;  
}
```

```
void AirconTurnon() { // 열전소자 쿨러 모듈을 켜는  
    함수  
    pinMode(REL, OUTPUT);
```

```
    digitalWrite(REL, HIGH); // 켜기
```

```
    return;  
}
```

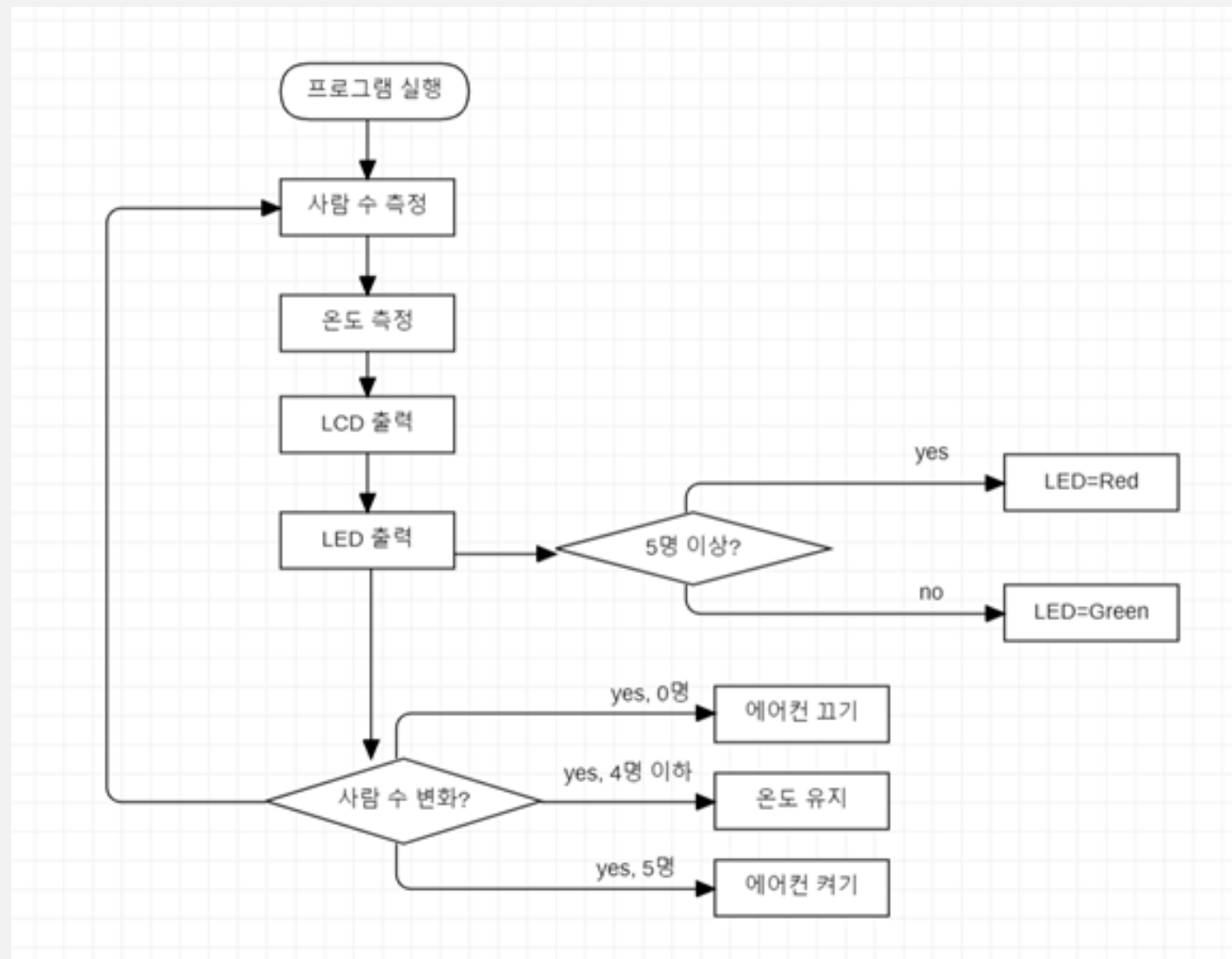
## 4

## 메인 함수 및 플로우 차트

```
// Setup 부분 생략
while (1) {
    int past_people = people; // 이전 승객 수 저장
    People(); // 현재 승객 수 측정
    MeasureTemperature(); // 온도 측정
    int temperature[2] = { dhtVal[2], dhtVal[3]}; // 온도의 정수값, 소숫값 배열에 저장
    printLCD(temperature, people); // 온도와 승객 수 LCD에 출력
    TurnLED(people); // 사람 수에 맞는 LED 불 ON
    if (past_people != people) { // 승객 수가 달라졌을 때
        if (people > 0 && people < 5) { //0명 이상 4명 이하 인 경우 - AirconMaintain()
            fp = AirconMaintain;
        }
        else if (people == 0) { //0명인 경우 - AirconTurnoff()
            fp = AirconTurnoff;
        }
        else if (people >= 5) { //5명인 경우 - AirconTurnon ()
            fp = AirconTurnon;
        }
    }
    //설정한 에어컨 모드 켜기
    fp();
    delay(8500); //8.5초 delay
}
```

## 4

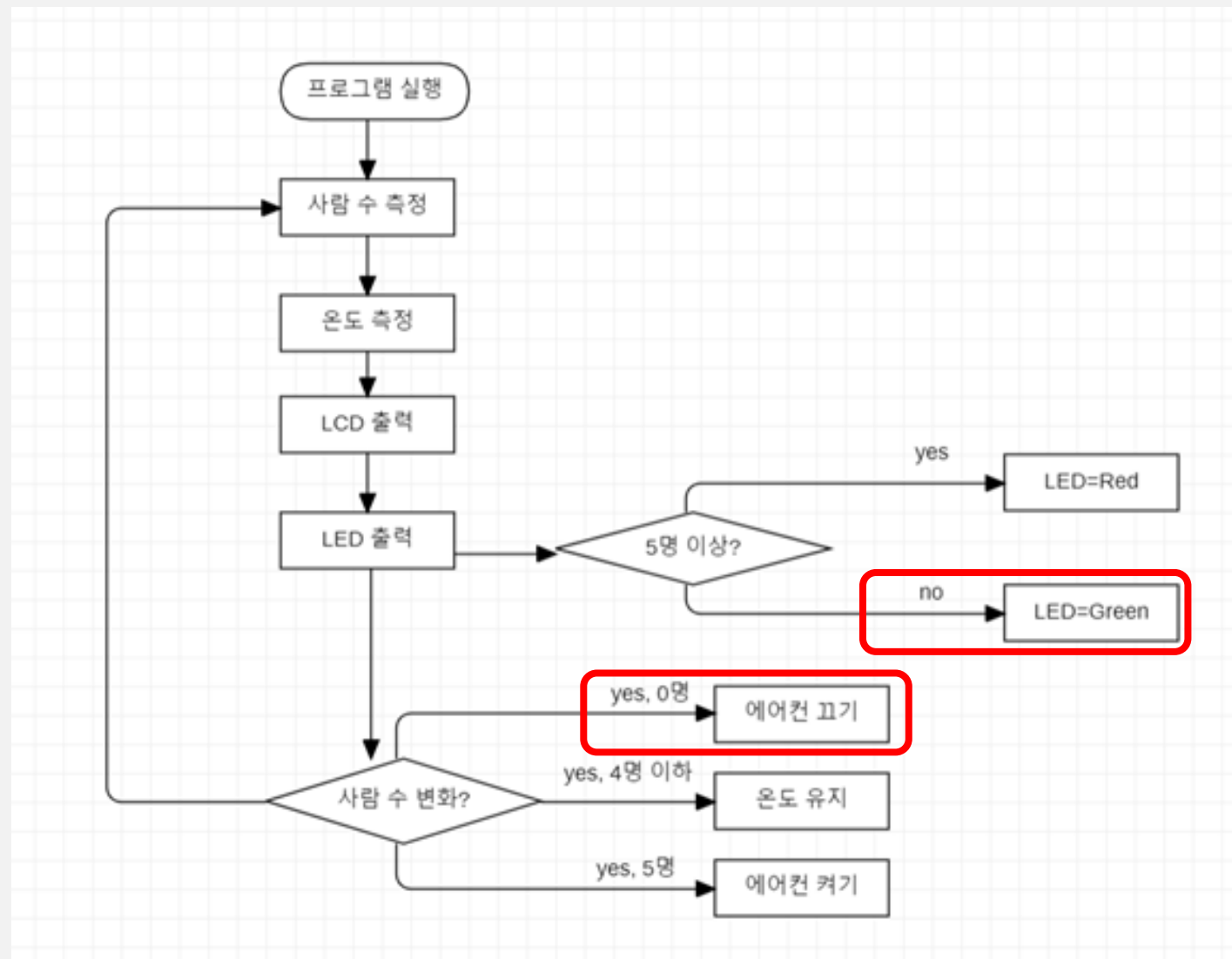
## 메인 함수 및 플로우 차트



## 5

플로우 차트 및 동작 시연

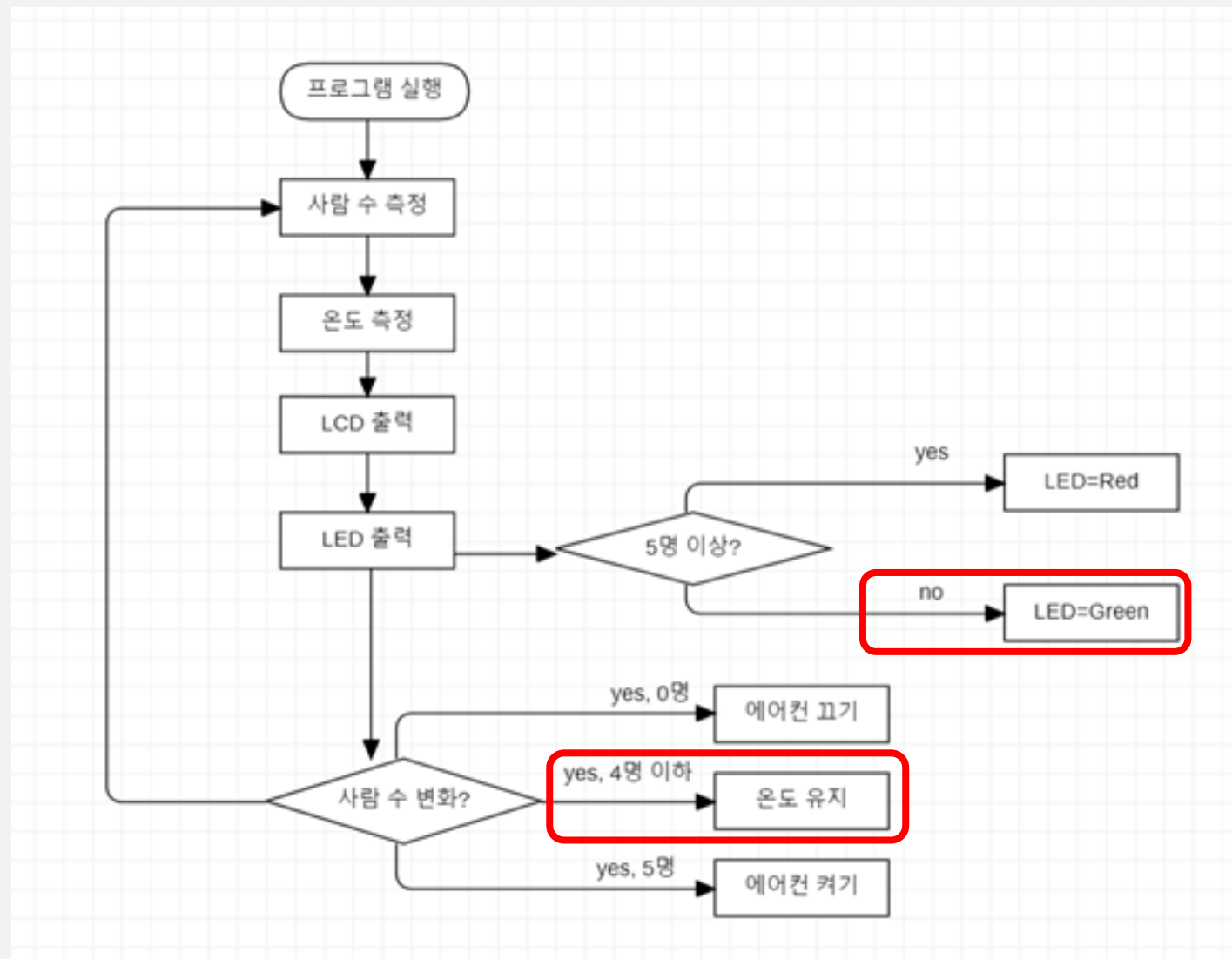
## 1) 0명일때



## 5

플로우 차트 및 동작 시연

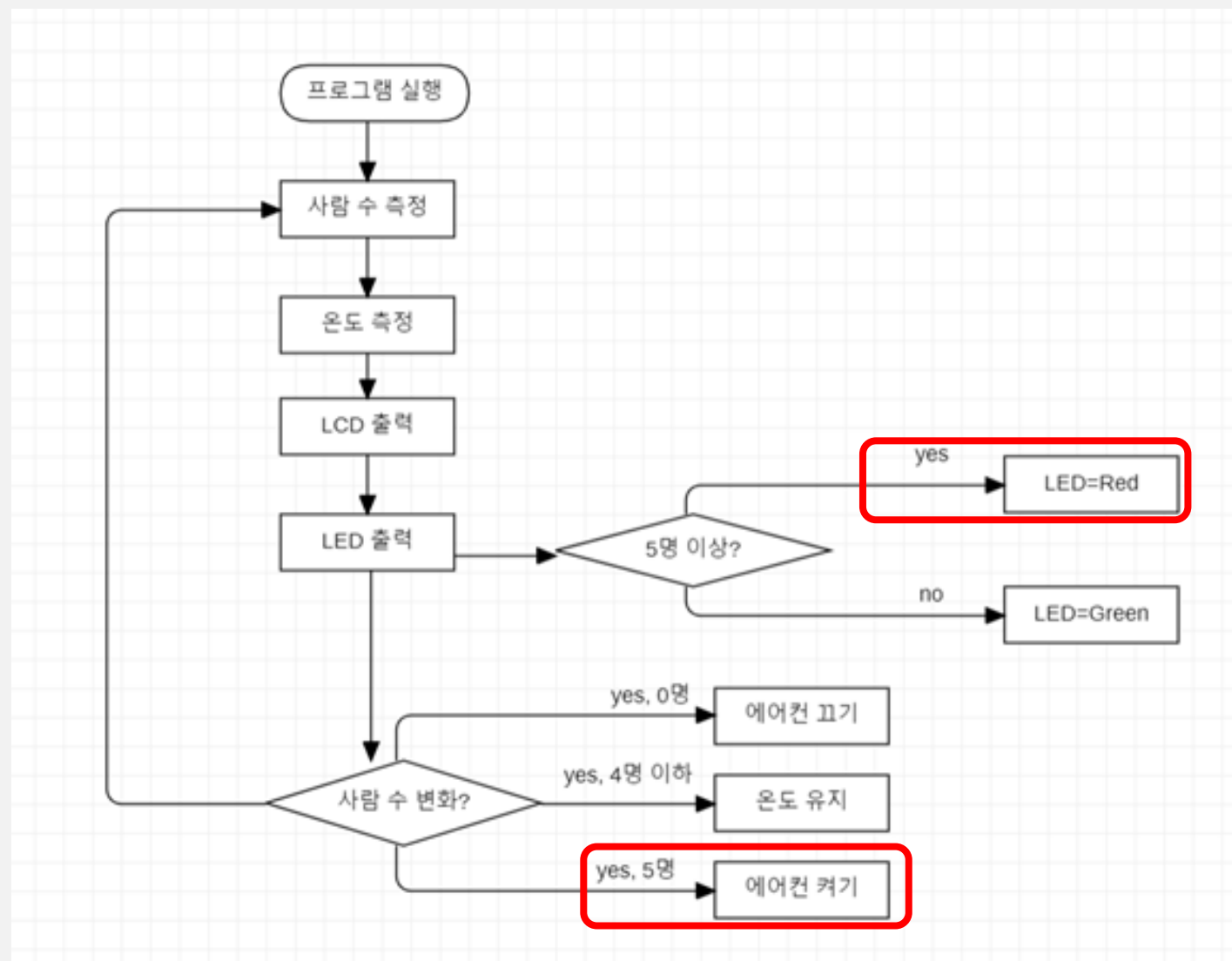
## 2) 3명일때



5

플로우 차트 및 동작 시연

## 3) 5명일때







- 본 프로젝트를 실생활에 적용하게 된다면, 자동 냉방 제어가 되어 불필요한 에너지 손실을 막을 수 있음
  - 또한, 승객들에게 좌석 수 및 내부 온도 정보를 제공하여 보다 편리한 대중 교통 이용에 도움을 줄 수 있을 것으로 기대함

123

**감 사 합 니 다**

IoT와임베디드소프트웨어 - 최종발표