

**TEAM. MBH**

# Display Show

강민우, 조보근, 정현우





# CONTENT

- 1**      프로젝트 설명

---

- 2**      프로젝트 수행 과정

---

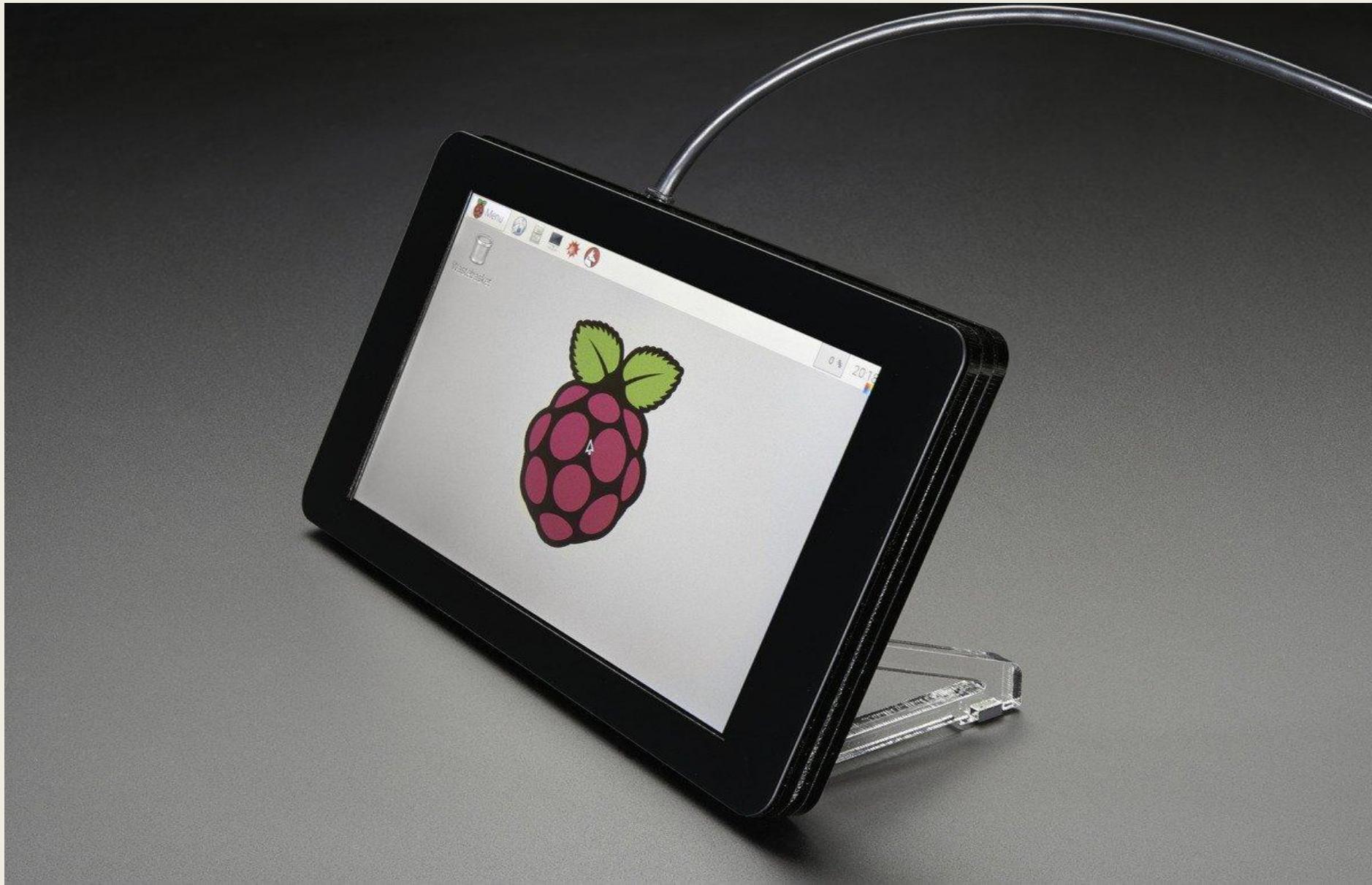
- 3**      활동 방안 & 추후 목표

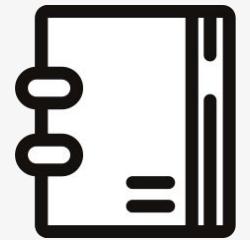
---

- 4**      코드 리뷰 및 해결방안



# DISPLAY SHOW





사전 영상 **PLAY**





# 개발자 양성과정 프로젝트 기안

## 프로젝트 설명

- 개발자 양성과정을 통해 중간 프로젝트를 진행
- 프로젝트를 통해 이전까지 배운 내용들을 활용
- 이러한 과정을 통하여 취업에 필요한 포트폴리오 내용에 추가

□ 프로젝트 명 : **DISPLAY SHOW**

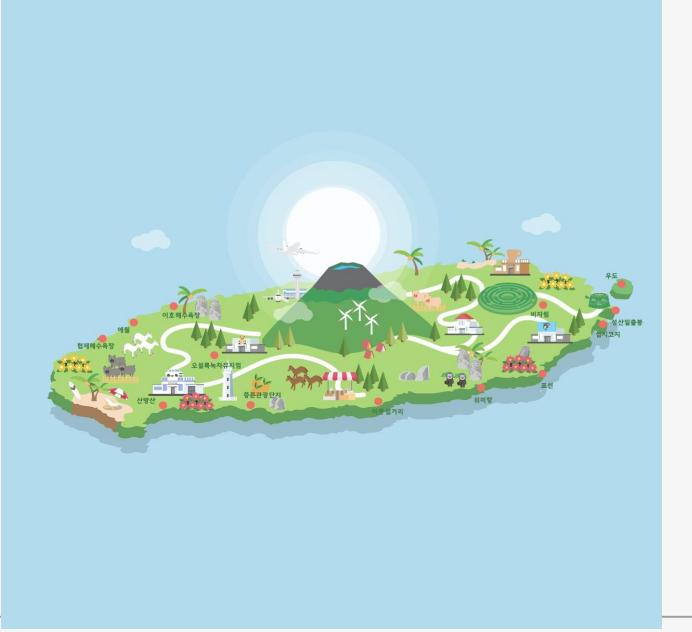
□ 목적 : 라즈베리파이를 활용해 시각적으로 원하는 데이터 표현하고자 함

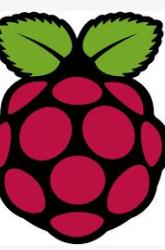
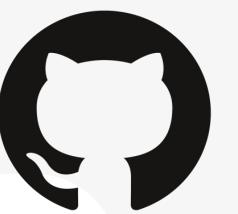
□ 기간 : 2024.05.10 (금) ~ 5. 20(월) (총 5일)

□ 인원 : 3명 (팀장 : 정현우, 팀원 : 강민우, 조보근)

□ 구 성 : 디스플레이 os - 라즈베리파이4 + 아두이노 uno

□ 주변요소 : WS2812B LED STRIP, 3D 프린트를 이용해 디스플레이 고정 틀 제작

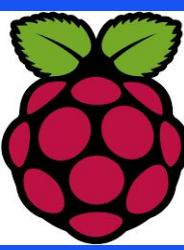
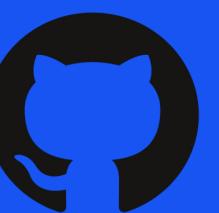
	무드 (베이지)	게임	여행( 제주도 )
구성	베이지색 바탕에 무드있는 디스플레이	게임컨셉에 맞는 디스플레이	돌하르방, 혹은 한라산 배경으로 제주도 여행에 알맞은 디스플레이
내용	• 데이터 ( 날짜와 시간)을 표현하고 컨셉에 맞는 구성요소를 통한 장식품		
타겟	집 인테리어에 관심이 있으신 고객	게임을 선호하는 고객	풍경이나 여행을 선호하는 고객
예시			



## ▣ 프로젝트 일정

날짜	1일차	2일차	3일차	4일차	5일차	6일차
	5/10 [ 금 ]	5/13 [ 월 ]	5/14 [ 화 ]	5/16 [ 목 ]	5/17 [ 금 ]	5/20 [ 월 ]
목적	프로젝트 방향성 확립 세부계획수립	프로젝트 수행 코딩 및 조립	이미지 디자인 시안 기상청 데이터	프로젝트 수행 코딩 및 조립	코드 병합 + LED 마무리	최종 마무리 시연영상 제작 + 발표 준비
내용	기안작성(slide) 컨셉 및 구상도 세부계획 수립		LED 표현, 디스플레이 표현, 데이터 통신 계획서작성, 기술서작성, doxygen작성, 발표 및 PPT준비			마무리 작성 및 발표
비고	회의	회의를 통한 display 결과	재료 구매 [다이소]	오후 특강 [자기소개서]	최종 마무리	



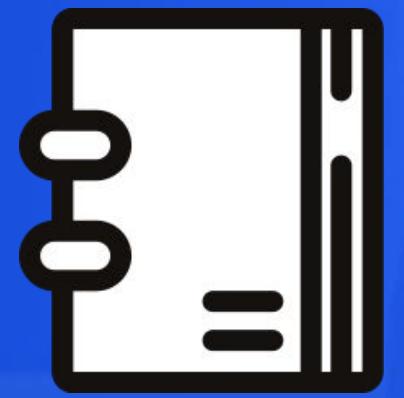


# 프로젝트 3가지

## 수행목표

- 1      실시간 날씨 DATA [기온, 강수량, 전운량] + 시간을 텍스트  
라즈베리파이4를 이용해 디스플레이에 표현
- 2      아두이노[uno]를 이용해  
WS2812B LED 스트랩 - 스위치로 컨트롤
- 3      이때까지 배운내용을 최대한 활용해  
하나로 표현해보고자 함

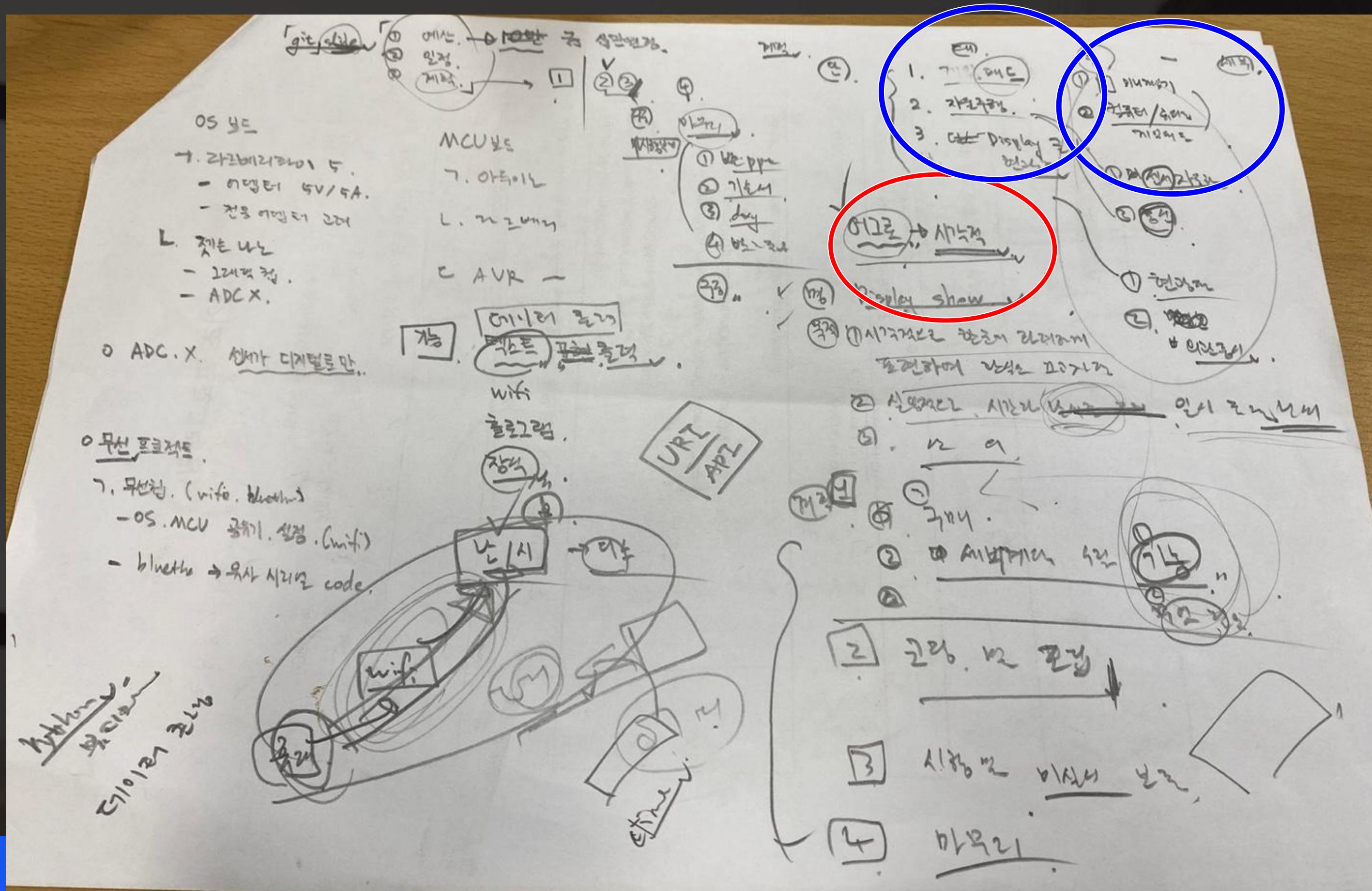




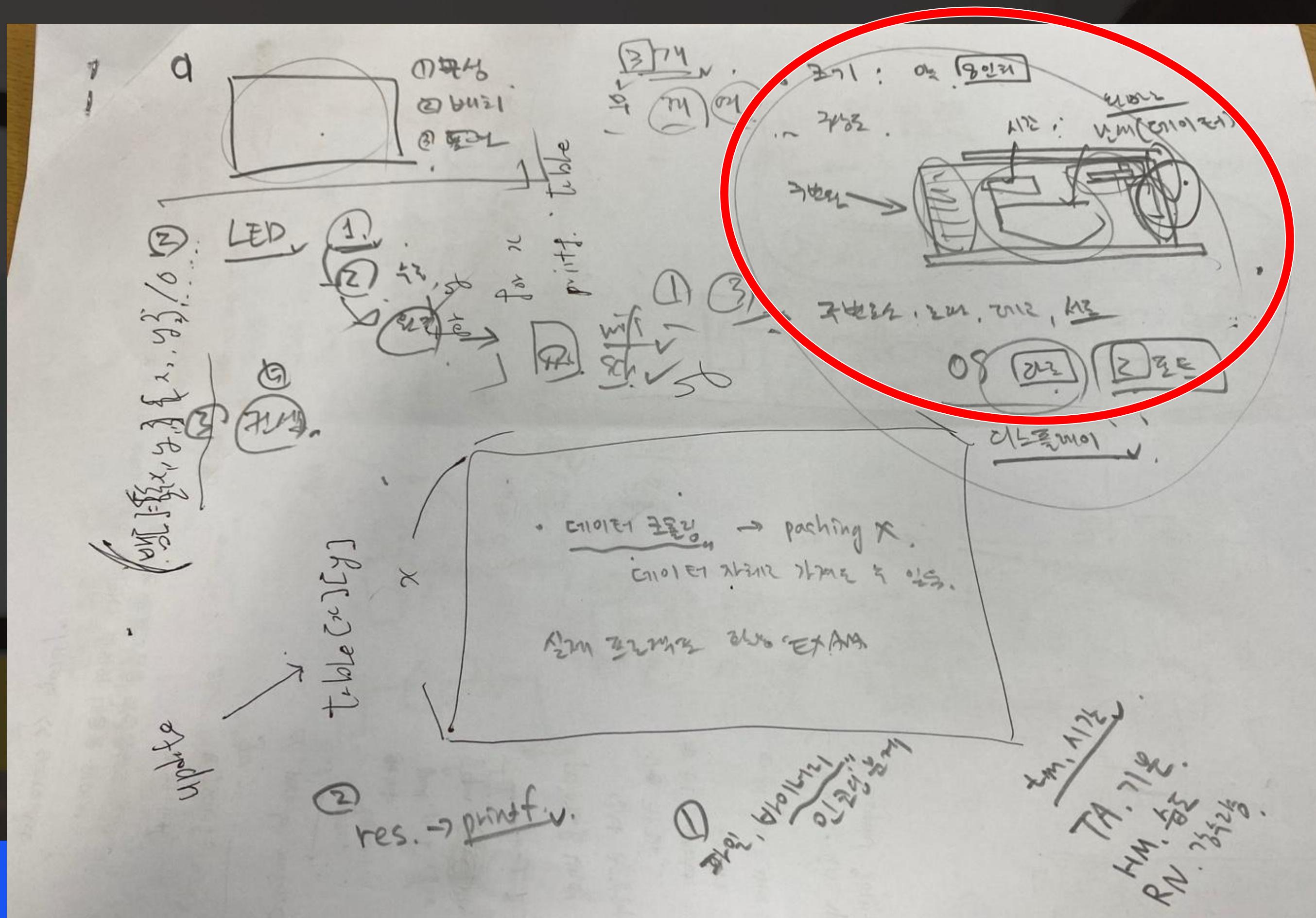
# 프로젝트 수행 과정

# Day1

## 회의 자료



Day1  
회의  
자료

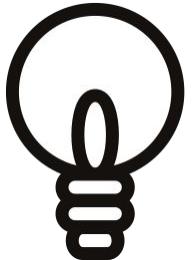


# 3가지 주요 목표 [월]

날씨 데이터 구축 + 코딩

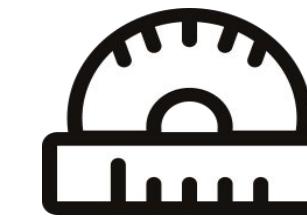


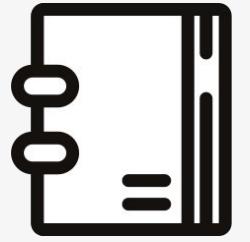
아두이노  
[네오픽셀 사용가능 여부 테스트]  
LED 실행가능 여부  
[WS2812B LED STRIP]



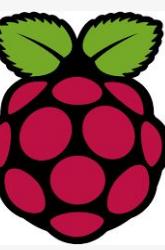
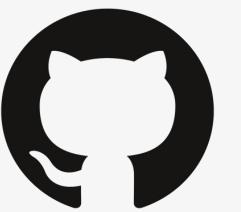
3D프린터 목업

기본틀 제작





# 13일 월요일 프로젝트 진행 과정



## 날씨 API 데이터 구축 + 기상청 사이트 활용

기상청 API허브

지상관측

총관기상관측(ASOS)

AWS 관측분석

기상, 강수, 바람, 환경, 자외선 등

96지점 (2020.4.1. 기준)

1904년 4월 ~ 현재(지점별 상이함)

본, 시간, 일, 월, 연, 지역

11 문의

대용량 API 신청

1. 지상 관측자료 조회

국제기상전보식(2012) 참고자료

호출URL정보

https://apihub.kma.go.kr/api/tpl01/url/kma\_sfctm2.php?tm=202211300900&strn=0&help=1&authKey=93hdJMSUQ604XVD0V0OjwA

URL복사

요청인자

인자명	의미	설명
tm	년월일시분(KST)년월일(KST)	해당 시간 (없으면 현재시간)
strn	지점번호	해당 지점들(로 구분)의 정보 표출 (0 이거나 없으면 전체지점)

기상청 API허브로 기상기후데이터 알차게 활용하기

2023. 2. 6. 15:00

기상청이야기

기상청 API허브로 기상기후데이터 알차게 활용하기

기상청은 매일 하늘에서 망 위에서, 레이더와 기상위성을 통해 현재의 기상 상태를 관측합니다. 또한, 현재의 관측자료를 활용해 수치모델을 돌려 미래의 예측 자료를 만들어내기도 하죠! 이뿐만 아니라 100년 후면 미래를 예측한 기후변화 시나리오까지 만들어내고 있습니다. 이러한 자료들이 쌓이고 쌓여 매우 방대한 양의 데이터를 보유하고 있습니다.

기상청 API허브로 기상기후데이터 활용하기

기상서비스진흥국

기상청이야기 카테고리 글

콘텐츠를 통해 돌아보는 진진, 진진해일, 화산

2023. 2. 9.

국립전북기상과학관을 100% 활용하는 방법

2023. 2. 7.

기상청 API허브로 기상기후데이터 알차게 활용하기

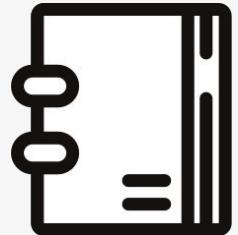
2023. 2. 6.

마서운 동장군도 '움찔'하는 「한파 피해 예방 캠페인」

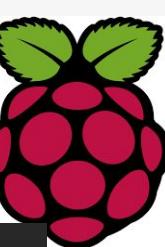
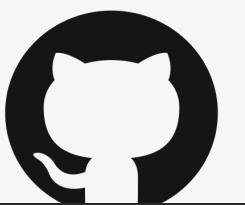
2023. 2. 3.

겨울철 도로 안전, 기상관측 차량이 함께 합니다!

2023. 1. 30.



# 13일 월요일 프로젝트 진행 과정



## FINDTEXT.CPP



```
weather > C: findText.cpp > main()
1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 #include <string>
5
6 // 5번째 줄 12번, 14번, 16번에 있는 단어 출력
7 int main() {
8     std::ifstream inputFile("/home/hrd/Desktop/Project_MBH/weather/build/output.txt"); // 파일 경로를 수정하세요
9     std::string line;
10    std::string temperature; // 출력할 단어를 저장할 변수
11    std::string humidity; // 출력할 단어를 저장할 변수
12    std::string precipitation; // 출력할 단어를 저장할 변수
13
14    // 파일이 열렸는지 확인
15    if (!inputFile.is_open()) {
16        std::cerr << "파일을 열 수 없습니다." << std::endl;
17        return 1;
18    }
19
20    // 파일을 한 줄씩 읽어서 특정 줄과 특정 단어를 출력
21    int currentLine = 0;
22    while (std::getline(inputFile, line)) {
23        currentLine++;
24
25        // 5번째 줄인 경우
26        if (currentLine == 5) {
27            std::istringstream iss(line);
28            std::string word;
29            int wordCount = 0;
30
31            // 공백을 구분자로 하여 단어 출력
32            while (iss >> word) {
33                word // std::string temperature
34                if ( // 추출한 단어를 저장할 변수
35                    temperature = word;
36                    continue;
37                }
38                // 14번째 단어
39                if (wordCount == 14) {
40                    humidity = word;
41                    continue;
42                }
43                // 16번째 단어
44                if (wordCount == 16) {
45                    precipitation = word;
46                    break;
47                }
48            }
49        }
50
51        break;
52    }
53
54
55    // 단어를 출력
56    std::cout << "현재 세종시 기온은 " << temperature << "도입니다." << std::endl;
57    std::cout << "현재 세종시 습도은 " << humidity << "도입니다." << std::endl;
58    std::cout << "현재 세종시 강수량은 " << precipitation << "mm입니다." << std::endl;
59
60    inputFile.close();
61
62    return 0;
63}
```

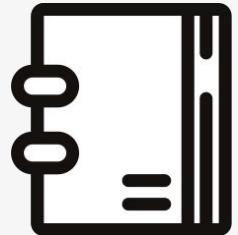
문제 출력 디버그 콘솔 터미널 포트

✓ 호출 스택

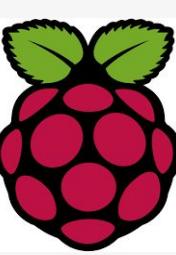
✓ 터미널

Consolidate compiler generated dependencies of target findText

줄 25, 열 21 공

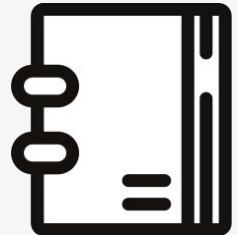


# 13일 월요일 프로젝트 진행 과정

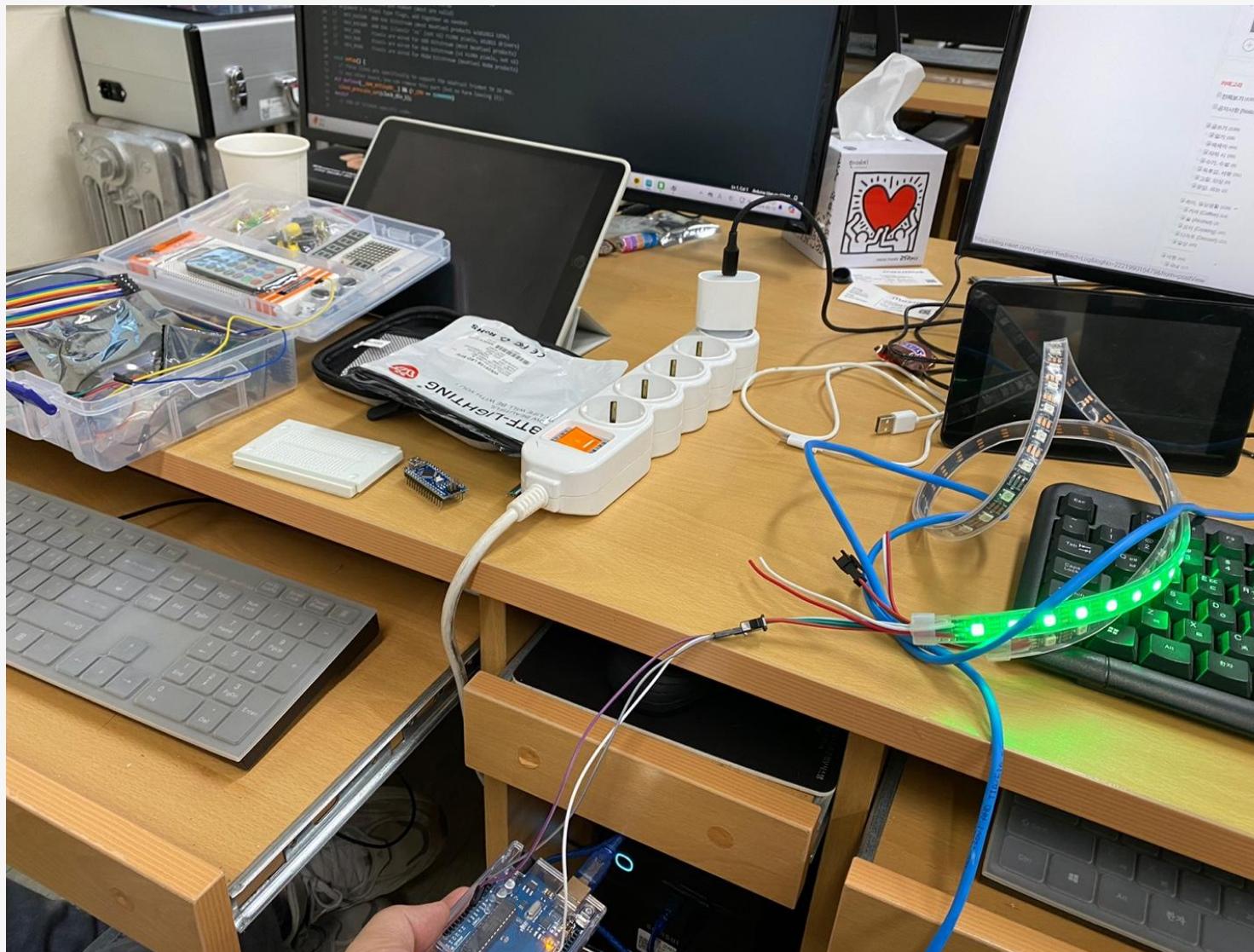
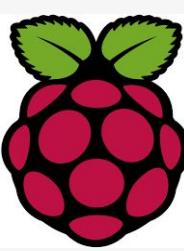
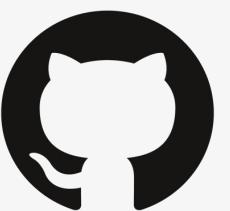


```
weather.cpp M X  ≡ site.txt

weather > weather.cpp > main()
1 #include <curl/curl.h>
2 #include <stdio.h>
3 #include <string>
4
5
6 int main() {
7     CURL* curl = curl_easy_init();
8     if (curl) {
9         FILE* fp = fopen("output.txt", "wb");
10        // 세종특별자치시 code : 239
11        const char* url = "https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=239&help=0&authKey=93hdUM5UQ604XVD0V00jwA" ; // URL 문자열 변수
12        curl_easy_setopt(curl, CURLOPT_URL, url); // URL 설정
13        curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp); // 데이터 쓰기 설정
14        CURLcode res = curl_easy_perform(curl); // 데이터 다운로드 수행
15        if (res != CURLE_OK) {
16            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res)); // 오류 처리
17        }
18        curl_easy_cleanup(curl); // CURL 정리
19        fclose(fp); // 파일 닫기
20    }
21
22    return 0;
23 }
24
25
```

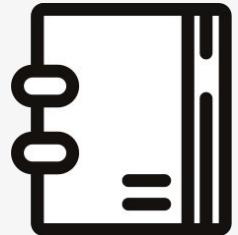


## 13일 월요일 프로젝트 진행 과정

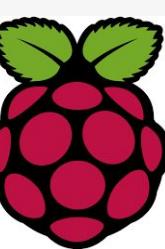


- 아두이노 UNO를 이용해, WS2812B LED Strip 불들어오게 하고,

가변저항을 통해 밝기 조절 까지 코드 적용 진행.



# 13일 월요일 프로젝트 진행 과정



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch\_may13a | 아두이노 1.8.13
- File Path:** sketch\_may13a | 아두이노 1.8.13
- Code Area:** The code is for an Adafruit NeoPixel LED project. It includes definitions for PIN 6 (button), SW 2 (button pin), BRIGHTNESS\_PIN A0 (brightness pin), and volatile int cnt = 0 (counter). It also defines parameters for the NeoPixel strip, including pixel type flags (NEO\_KHZ800, NEO\_KHZ400, NEO\_GRB, NEO\_RGB, NEO\_RGBW) and bitstream types (v1 or v2). The code initializes the NeoPixel strip, sets brightness to 50, and begins serial communication at 115200 bps. It also attaches an interrupt to the button pin to handle button presses.
- Serial Monitor:** Shows the message "업로드 완료." (Upload completed).
- Bottom Status:** Arduino Uno on /dev/ttyACM0

```
#include <Adafruit_NeoPixel.h>
#ifndef __AVR__
#include <avr/power.h>
#endif

#define PIN 6
#define SW 2 // 버튼 연결 핀
#define BRIGHTNESS_PIN A0 // 가변 저항 연결 핀
volatile int cnt = 0; // 인터럽트에 의해 변경되므로 volatile 선언

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
// NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)
// NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
// NEO_RGBW Pixels are wired for RGBW bitstream (NeoPixel RGBW products)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(60, PIN, NEO_GRB + NEO_KHZ800);

// IMPORTANT: To reduce NeoPixel burnout risk, add 1000 uF capacitor across
// pixel power leads, add 300 - 500 Ohm resistor on first pixel's data input
// and minimize distance between Arduino and first pixel. Avoid connecting
// on a live circuit...if you must, connect GND first.

void setup() {
  // This is for Trinket 5V 16MHz, you can remove these three lines if you are not using a Trinket
  #if defined (__AVR_ATtiny85__)
  if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
  #endif
  // End of trinket special code

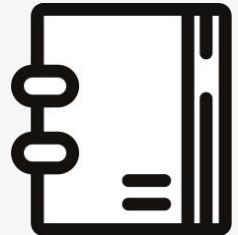
  pinMode(SW, INPUT_PULLUP); // 버튼 핀을 풀업 입력 모드로 설정

  strip.begin();
  strip.setBrightness(50);
  strip.show(); // Initialize all pixels to 'off'
  Serial.begin(115200);

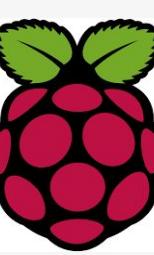
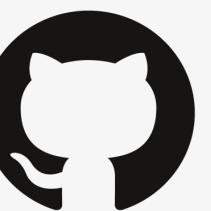
  // 인터럽트 설정 (버튼 핀에 FALLING 모드 설정)
  attachInterrupt(digitalPinToInterrupt(SW), buttonISR, FALLING);
}

업로드 완료.

스케치는 프로그램 저장 공간 5416 바이트(16%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 244바이트(11%)를 사용, 1804바이트의 지역변수가 남음. 최대는 2048 바이트.
```



# 13일 월요일 프로젝트 진행 과정



The screenshot shows the Arduino IDE interface. The top menu bar includes '파일' (File), '편집' (Edit), '스케치' (Sketch), '툴' (Tools), and '도움말' (Help). The toolbar below has icons for upload, download, and serial communication. The sketch window title is 'sketch\_may13a'. The code itself is written in C++ and controls an RGB LED strip via a NeoPixel library. It reads analog input from pin BRIGHTNESS\_PIN, maps it to a brightness value (0 to 255), and sets the NeoPixel strip's brightness. It then prints the current count (cnt) and brightness values to the Serial Monitor. A switch statement handles five different LED patterns based on the count value:

```
void loop() {
    int brightnessValue = map(analogRead(BRIGHTNESS_PIN), 0, 1023, 0, 255); // 아날로그 입력 값을 밝기로 변환합니다.
    strip.setBrightness(brightnessValue); // NeoPixel의 밝기를 설정합니다.
    Serial.print("cnt: ");
    Serial.println(cnt);
    Serial.print("Brightness: ");
    Serial.println(brightnessValue);

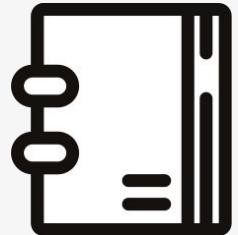
    // cnt 값에 따라 LED 패턴 실행
    switch (cnt) {
        case 1:
            colorWipe(strip.Color(255, 0, 0), 50); // Red
            colorWipe(strip.Color(0, 255, 0), 50); // Green
            colorWipe(strip.Color(0, 0, 255), 50); // Blue
            colorWipe(strip.Color(0, 0, 255), 50); // White RGBW
            break;
        case 2:
            theaterChase(strip.Color(127, 127, 127), 50); // White
            theaterChase(strip.Color(127, 0, 0), 50); // Red
            theaterChase(strip.Color(0, 0, 127), 50); // Blue
            break;
        case 3:
            rainbow(8);
            break;
        case 4:
            rainbowCycle(8);
            break;
        case 5:
            theaterChaseRainbow(8);
            break;
        default:
            break;
    }

    // cnt 값을 제한
    if (cnt > 5) {
        cnt = 0;
    }

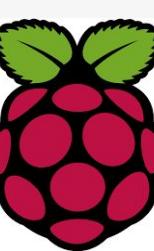
    // 작은 지연 추가하여 CPU 과부하 방지
    delay(10);
}
```

The status bar at the bottom indicates "업로드 완료." (Upload Complete) and provides memory usage details: "스케치는 프로그램 저장 공간 5416 바이트(16%)를 사용. 최대 32256 바이트." and "전역 변수는 동적 메모리 244바이트(11%)를 사용, 1804바이트의 지역변수가 남음. 최대는 2048 바이트."

## ARDUINO. LED



# 13일 월요일 프로젝트 진행 과정



The screenshot shows the Arduino IDE interface. The title bar indicates the sketch is named 'sketch\_may13a' and is based on '아두이노 1.8.13'. The code editor displays C++ code for an Arduino project. The code includes functions for handling button presses and creating rainbow patterns on a strip of LEDs. A message at the bottom of the code editor says '업로드 완료.' (Upload completed). Below the code editor, a status bar shows memory usage: 5416 bytes (16%) used, 32256 bytes available, and 244 bytes (11%) of dynamic memory left, with a maximum of 1804 bytes available.

```
sketch_may13a

}

// 인터럽트 서비스 루틴 (ISR)
void buttonISR() {
    cnt++;
}

// Fill the dots one after the other with a color
void colorWipe(uint32_t c, uint8_t wait) {
    for(uint16_t i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
        delay(wait);
        if(cnt != 1) {
            return;
        }
    }
}

void rainbow(uint8_t wait) {
    uint16_t i, j;

    for(j=0; j<256; j++) {
        for(i=0; i<strip.numPixels(); i++) {
            strip.setPixelColor(i, Wheel((i+j) & 255));
        }
        strip.show();
        delay(wait);
        if(cnt != 3) {
            return;
        }
    }
}

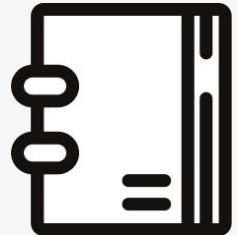
// Slightly different, this makes the rainbow equally distributed throughout
void rainbowCycle(uint8_t wait) {
    uint16_t i, j;

    for(j=0; j<256*5; j++) { // 5 cycles of all colors on wheel
        for(i=0; i< strip.numPixels(); i++) {
            strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) + j) & 255));
        }
    }
}

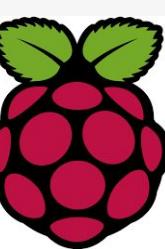
업로드 완료.

스케치는 프로그램 저장 공간 5416 바이트(16%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 244바이트(11%)를 사용, 1804바이트의 지역변수가 남음. 최대는 2048 바이트.

Arduino Uno on /dev/ttyACM0
```



# 13일 월요일 프로젝트 진행 과정



The screenshot shows the Arduino IDE interface. The title bar indicates the sketch is named 'sketch\_may13a' and is based on '아두이노 1.8.13'. The code in the editor is for an LED project, featuring functions like `theaterChase` and `theaterChaseRainbow` to create theater-style crawling lights with or without a rainbow effect. The code uses `strip` objects and `delay` functions. The status bar at the bottom shows the message '업로드 완료.' (Upload completed) and provides memory usage details: 5416 bytes (16%) used, 32256 bytes total, and 244 bytes (11%) of dynamic memory used.

```
sketch_may13a
strip.show();
delay(wait);
if(cnt != 4) {
    return;
}
}

//Theatre-style crawling lights.
void theaterChase(uint32_t c, uint8_t wait) {
    for (int j=0; j<10; j++) { //do 10 cycles of chasing
        for (int q=0; q < 3; q++) {
            for (uint16_t i=0; i < strip.numPixels(); i=i+3) {
                strip.setPixelColor(i+q, c); //turn every third pixel on
            }
            strip.show();
            delay(wait);

            for (uint16_t i=0; i < strip.numPixels(); i=i+3) {
                strip.setPixelColor(i+q, 0); //turn every third pixel off
            }
        }
        if(cnt != 2) {
            return;
        }
    }
}

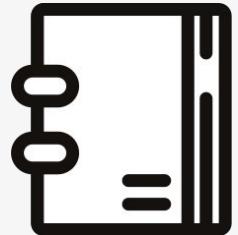
//Theatre-style crawling lights with rainbow effect
void theaterChaseRainbow(uint8_t wait) {
    for (int j=0; j < 256; j++) { // cycle all 256 colors in the wheel
        for (int q=0; q < 3; q++) {
            for (uint16_t i=0; i < strip.numPixels(); i=i+3) {
                strip.setPixelColor(i+q, Wheel((i+j) % 255)); //turn every third pixel on
            }
            strip.show();
            delay(wait);

            for (uint16_t i=0; i < strip.numPixels(); i=i+3) {
                strip.setPixelColor(i+q, 0); //turn every third pixel off
            }
        }
    }
}

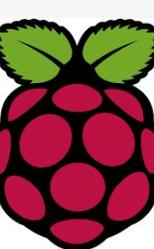
업로드 완료.

스케치는 프로그램 저장 공간 5416 바이트(16%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 244바이트(11%)를 사용, 1804바이트의 지역변수가 남음. 최대는 2048 바이트.
```

## ARDUINO. LED



# 13일 월요일 프로젝트 진행 과정



The screenshot shows the Arduino IDE interface. The title bar indicates the sketch is named 'sketch\_may13a' and is running on an 'Arduino Uno' connected via '/dev/ttyACM0'. The status bar shows the time as 12:28. The code in the editor is for a project involving an RGB LED strip, featuring a 'theaterChaseRainbow' function with a 'Wheel' color transition function. The code includes comments explaining the color transition logic. The bottom status bar shows memory usage: 5416 bytes used (16% of 32256), with 244 bytes of dynamic memory left. The page number '12' is at the bottom left, and the footer 'Arduino Uno on /dev/ttyACM0' is at the bottom right.

```
sketch_may13a
}
if(cnt != 2) {
    return;
}
}

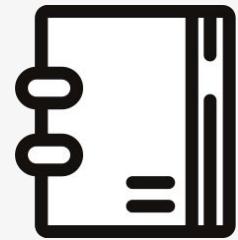
//Theatre-style crawling lights with rainbow effect
void theaterChaseRainbow(uint8_t wait) {
    for (int j=0; j < 256; j++) {      // cycle all 256 colors in the wheel
        for (int q=0; q < 3; q++) {
            for (uint16_t i=0; i < strip.numPixels(); i=i+3) {
                strip.setPixelColor(i+q, Wheel( (i+j) % 255));      //turn every third pixel on
            }
            strip.show();
            delay(wait);

            for (uint16_t i=0; i < strip.numPixels(); i=i+3) {
                strip.setPixelColor(i+q, 0);                  //turn every third pixel off
            }
        }
        if(cnt != 5) {
            return;
        }
    }
}

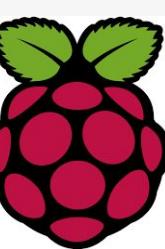
// Input a value 0 to 255 to get a color value.
// The colours are a transition r - g - b - back to r.
uint32_t Wheel(byte WheelPos) {
    WheelPos = 255 - WheelPos;
    if(WheelPos < 85) {
        return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
    }
    if(WheelPos < 170) {
        WheelPos -= 85;
        return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
    }
    WheelPos -= 170;
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

업로드 완료.

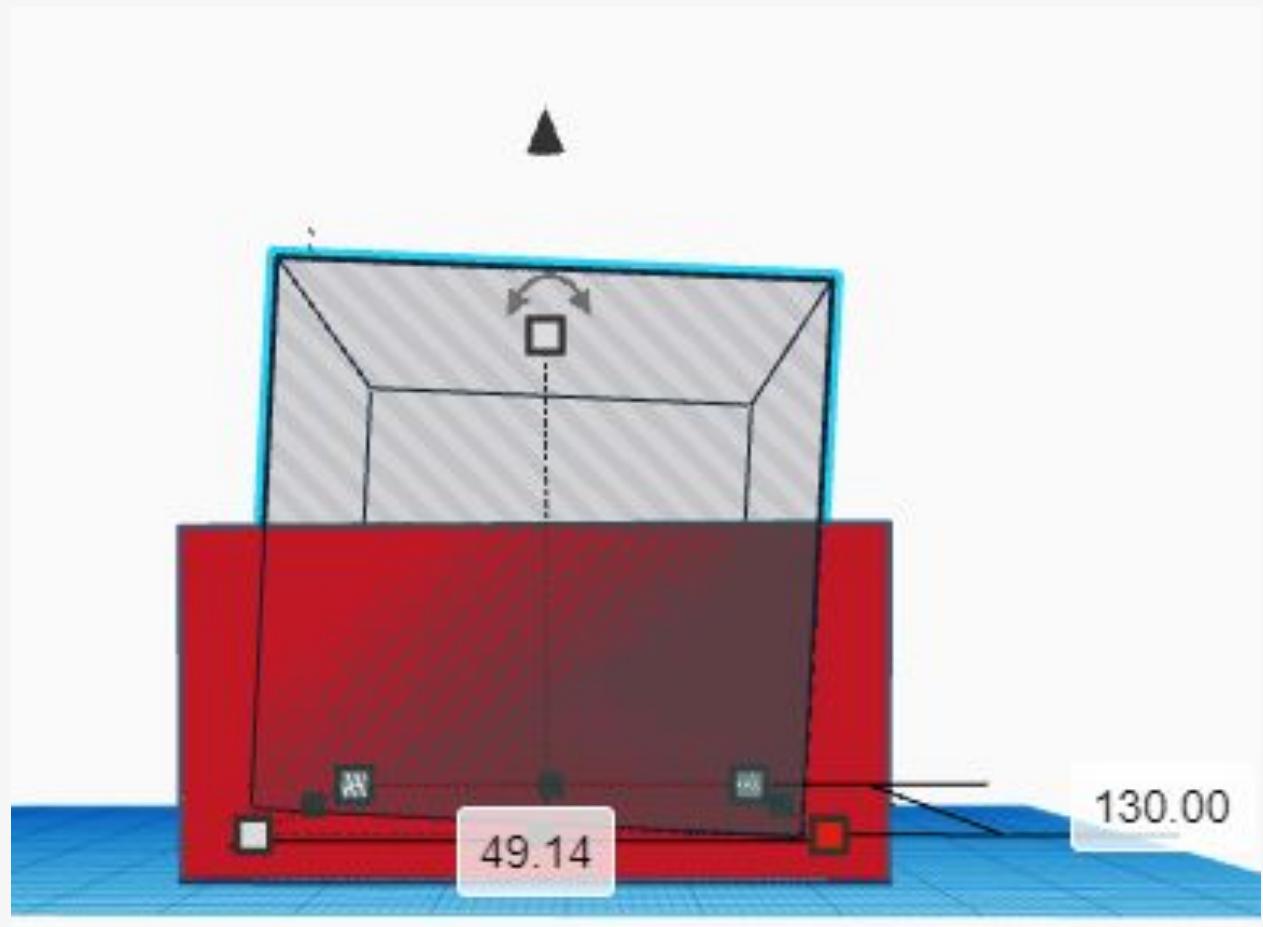
스케치는 프로그램 저장 공간 5416 바이트(16%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 244바이트(11%)를 사용, 1804바이트의 지역변수가 남음. 최대는 2048 바이트.
```



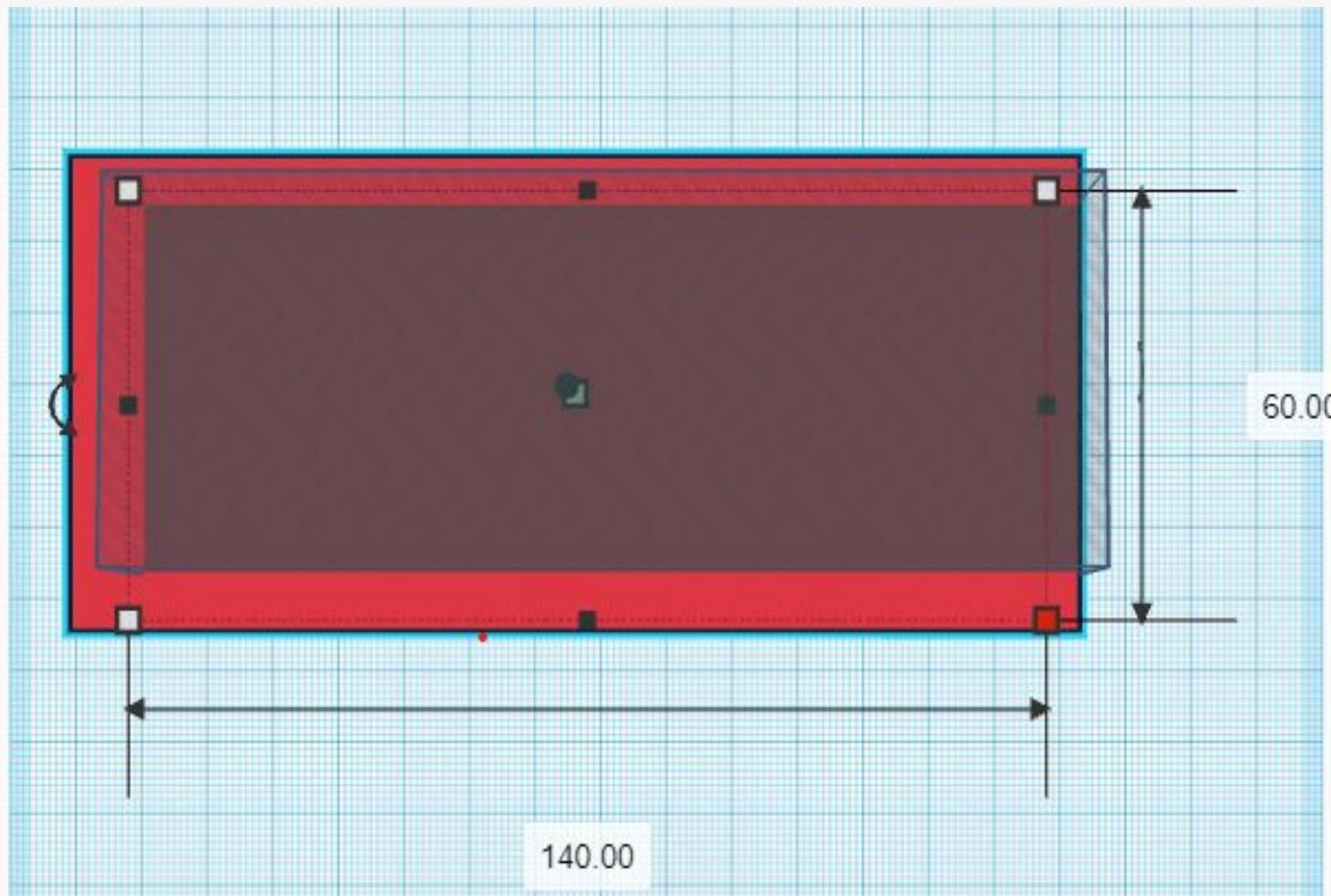
## 13일 월요일 프로젝트 진행 과정

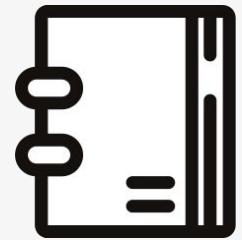


### 3D PRINT

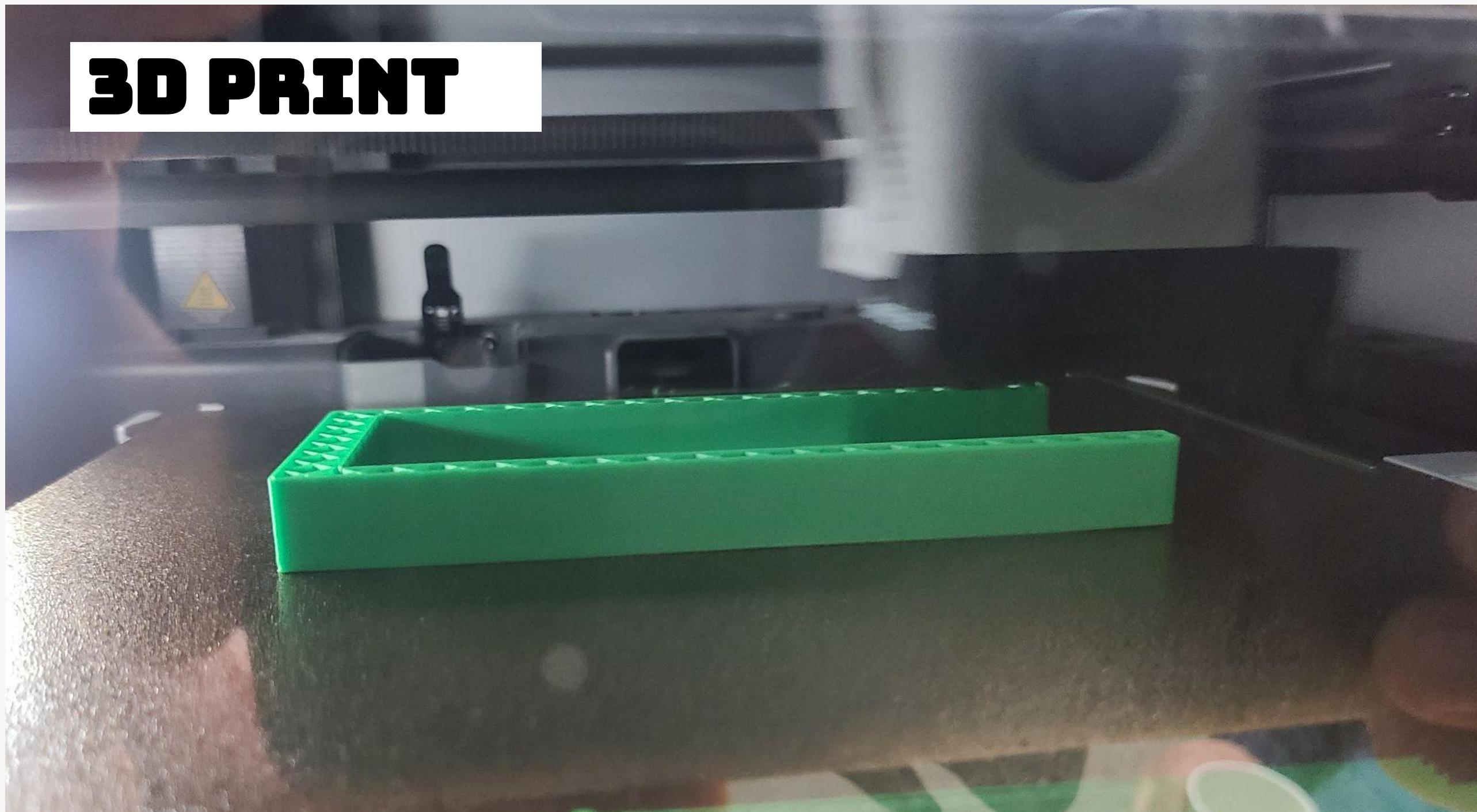
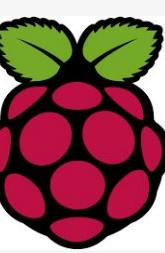
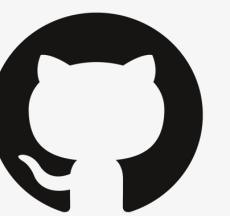


- 테스트용 3D 목업 제작 과정 [TinkerCAD - 인용]



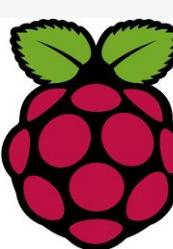


13일 월요일 프로젝트 진행 과정



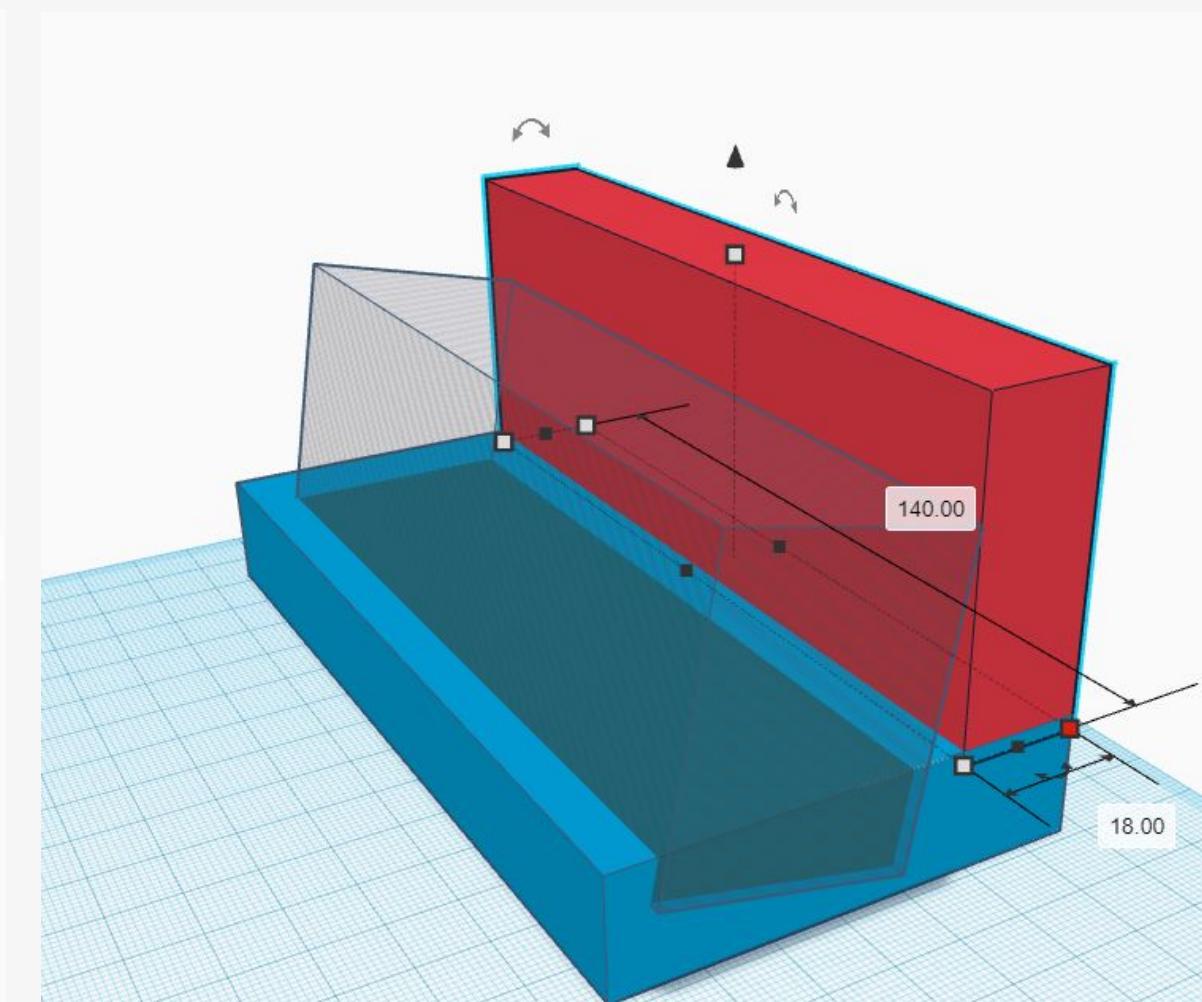
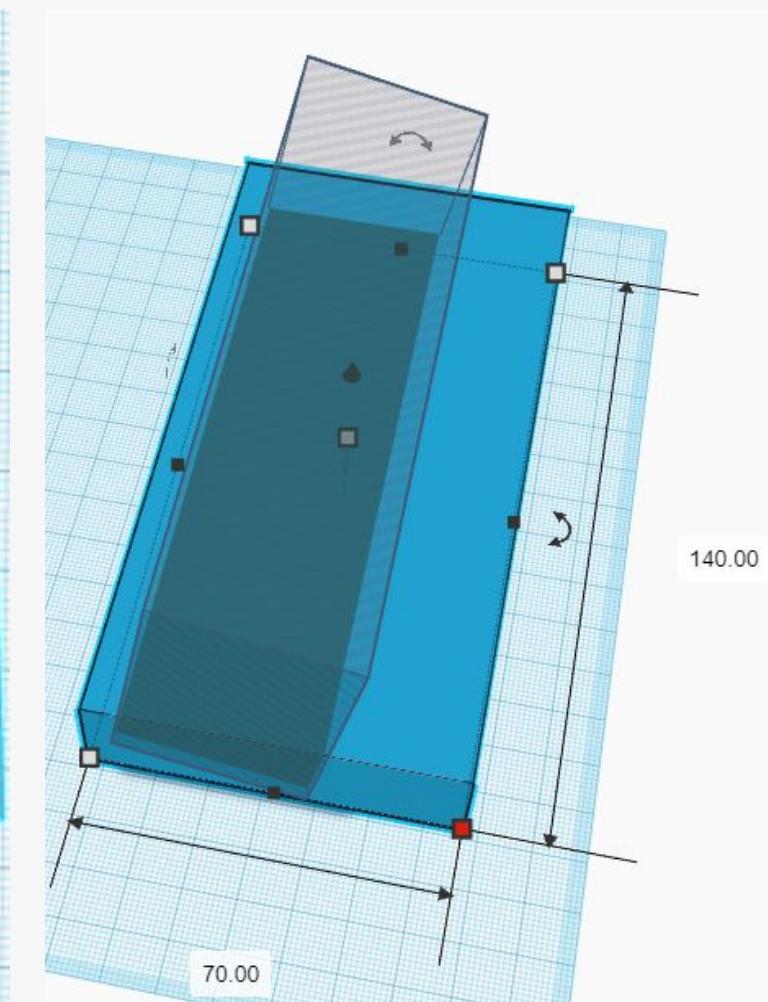
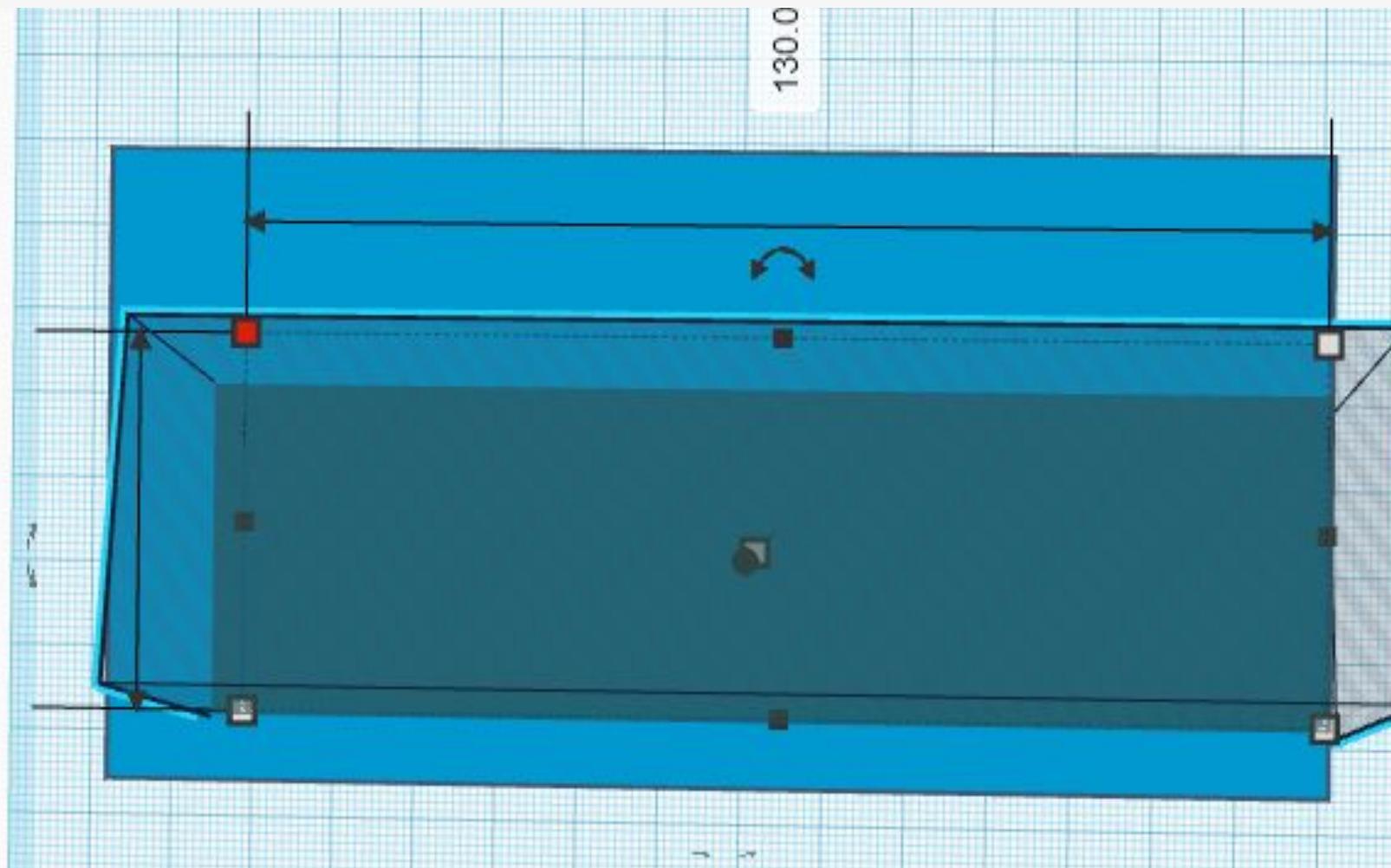


## 13일 월요일 프로젝트 진행 과정



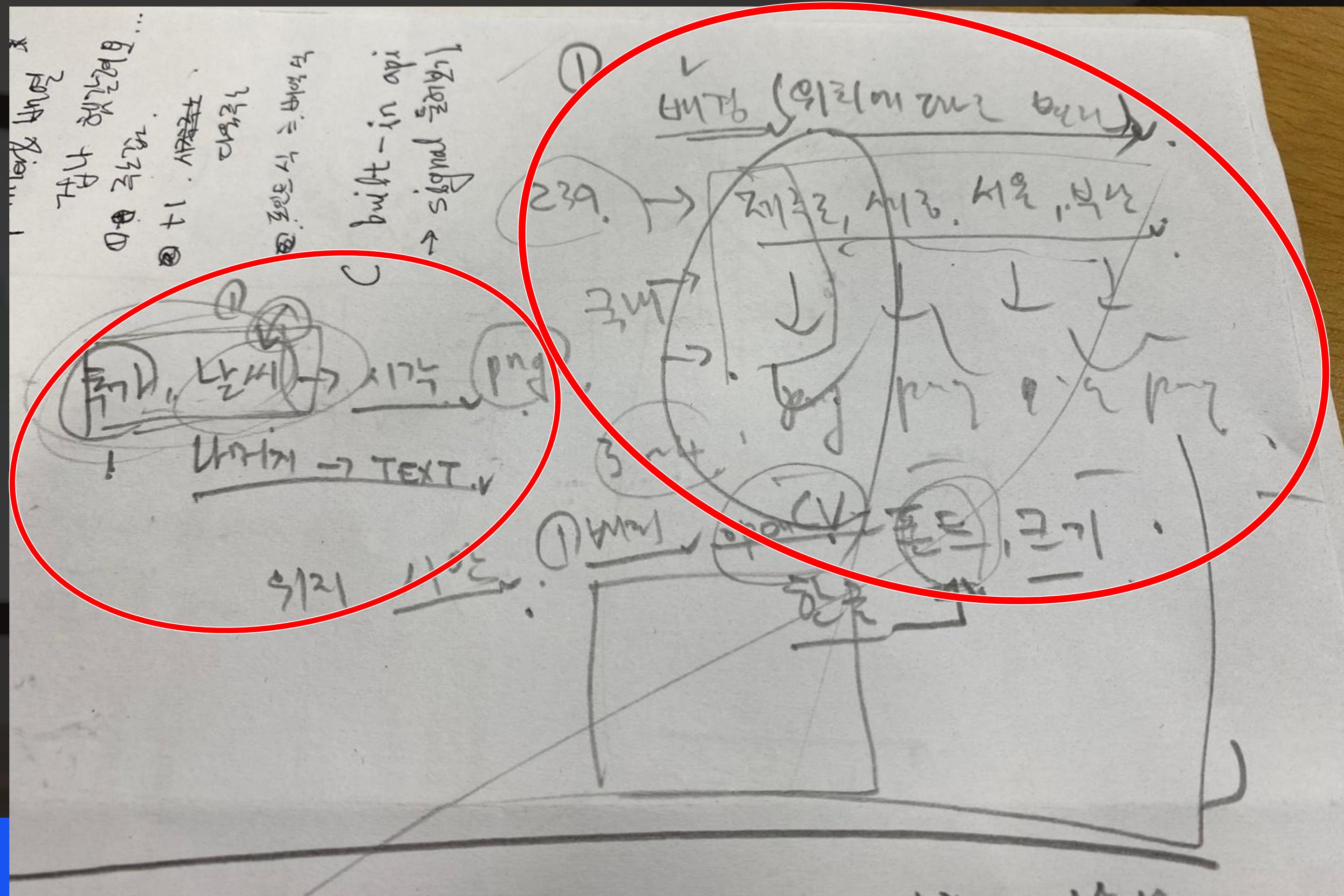
### 3D PRINT

- 보완점을 찾아 새롭게 도안 디자인 후, 최종 출력 준비 [ BAMBOO STUDIO ]



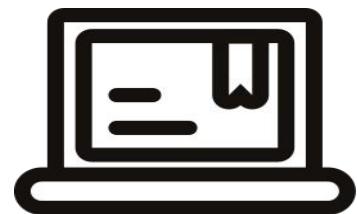
# Day 2

# 회의 자료



# 3가지 주요 목표 [화]

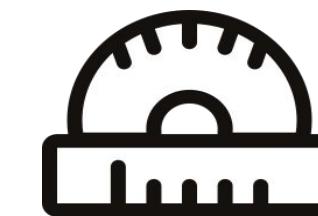
OPEN CV 구동 +  
한글 폰트- 이미지 시안-  
디스플레이 화면 구현

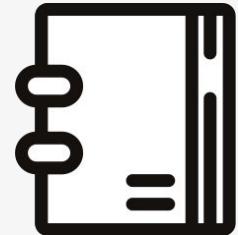


라즈베리파이 -  
LED 연동 여부  
[스위치 + 원격 가능]

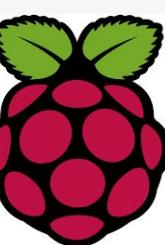


3D프린터 목업  
최종틀 제작완성



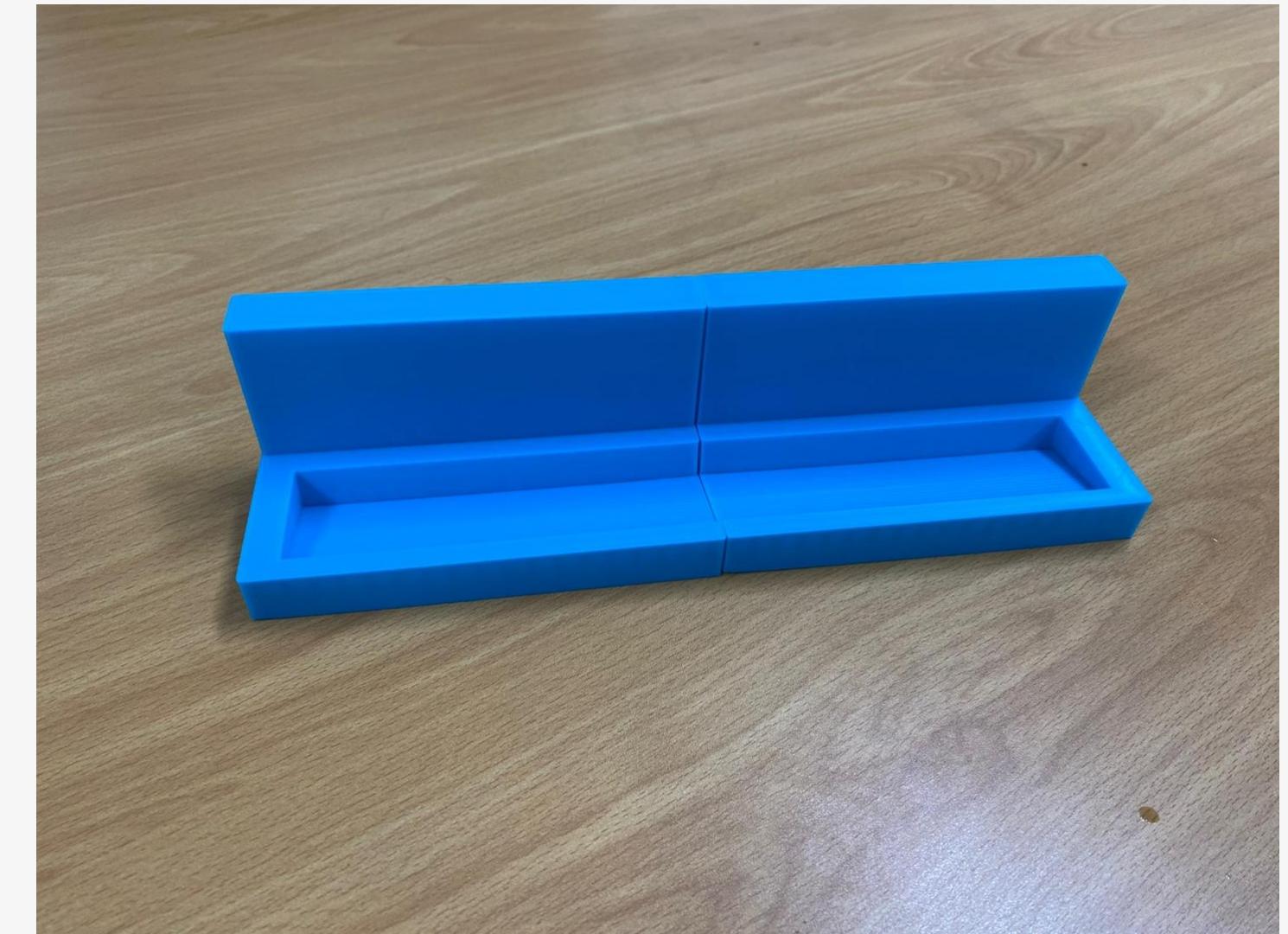


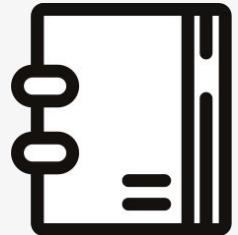
## 14일 화요일 프로젝트 진행 과정



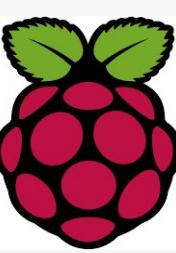
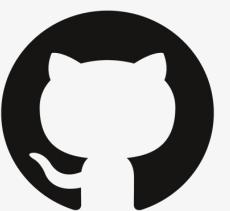
- 보완점을 찾아 새롭게 도안 디자인 후, 최종 출력 완성 [ BAMBOO STUDIO ]

### 3D PRINT





## 14일 화요일 프로젝트 진행 과정

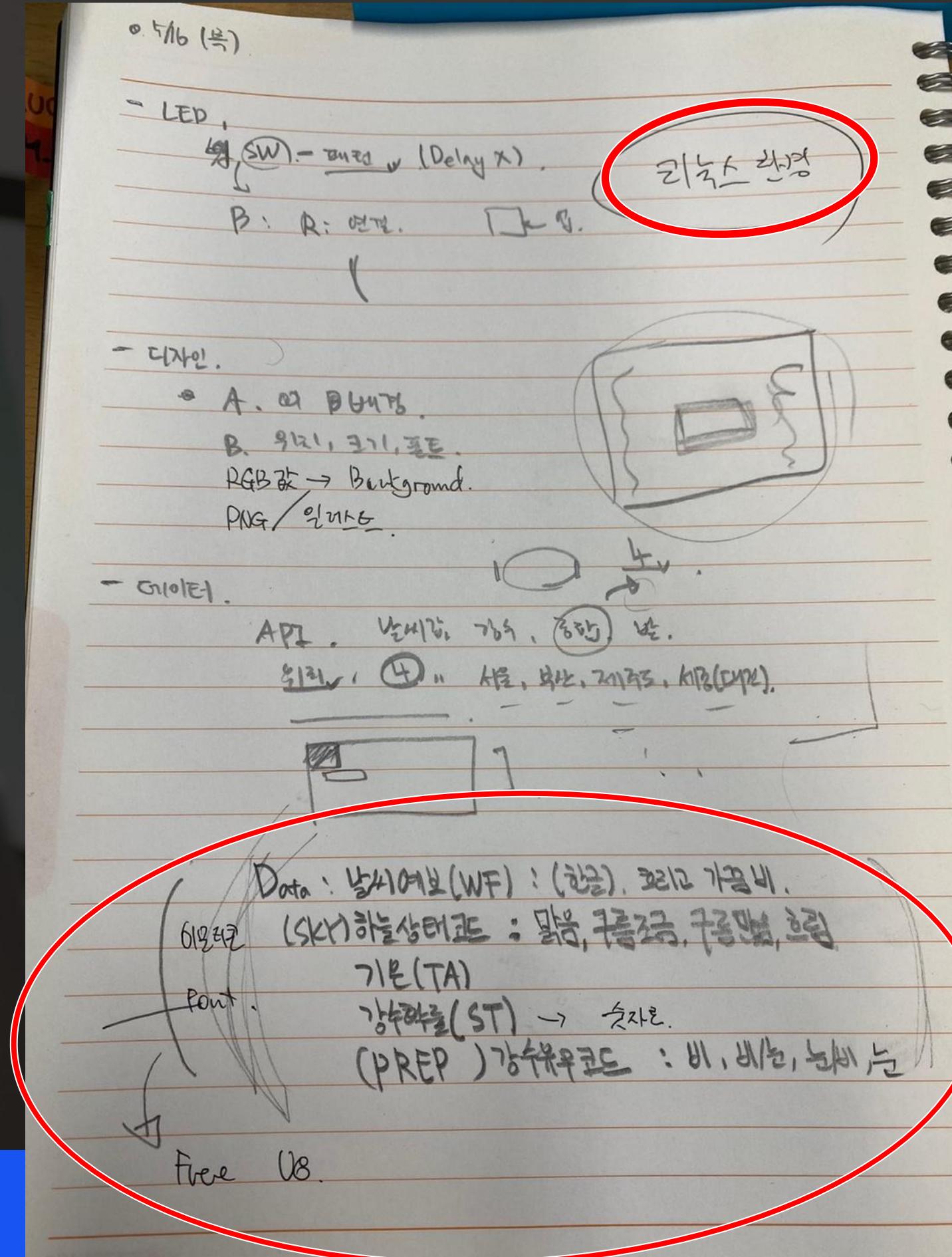


# TEST.CPP

```
text.cpp U    final.cpp U    main.cpp M    test.cpp U X    encoding.cpp U    README.md M ●  
project > openCV > test.cpp > main()  
1 #include <iostream>  
2 #include <string>  
3  
4 using namespace std;  
5  
6 string intToString(int number) {  
7     return to_string(number);  
8 }  
9  
10 int main() {  
11     int num = 123;  
12     string str = intToString(num);  
13     cout << "정수를 문자열로 변환한 결과: " << str << endl;  
14     return 0;  
15 }
```

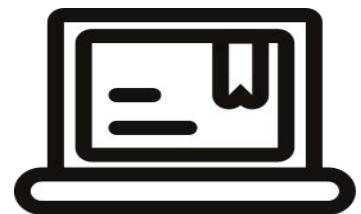
# Day 3

## 회의 자료

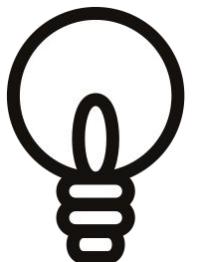


# 3가지 주요 목표 [목]

날씨데이터 + 코딩  
LED 코딩 합쳐서 작성  
이미지 파일  
디자인시안 + OPEN CV

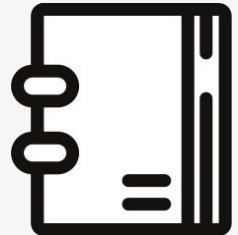


라즈베리파이 +  
아두이노 나노  
합치기

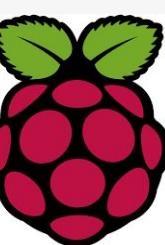
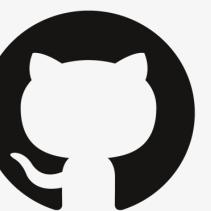


목업 재수정  
LED데코레이션



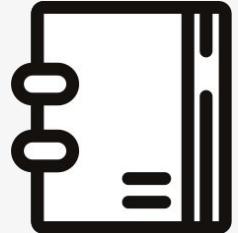


## 14일 화요일 프로젝트 진행 과정

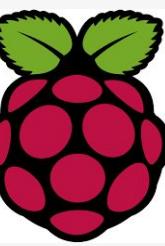


# ENCODING.CPP [UTF-8] 【인코딩 디코딩】

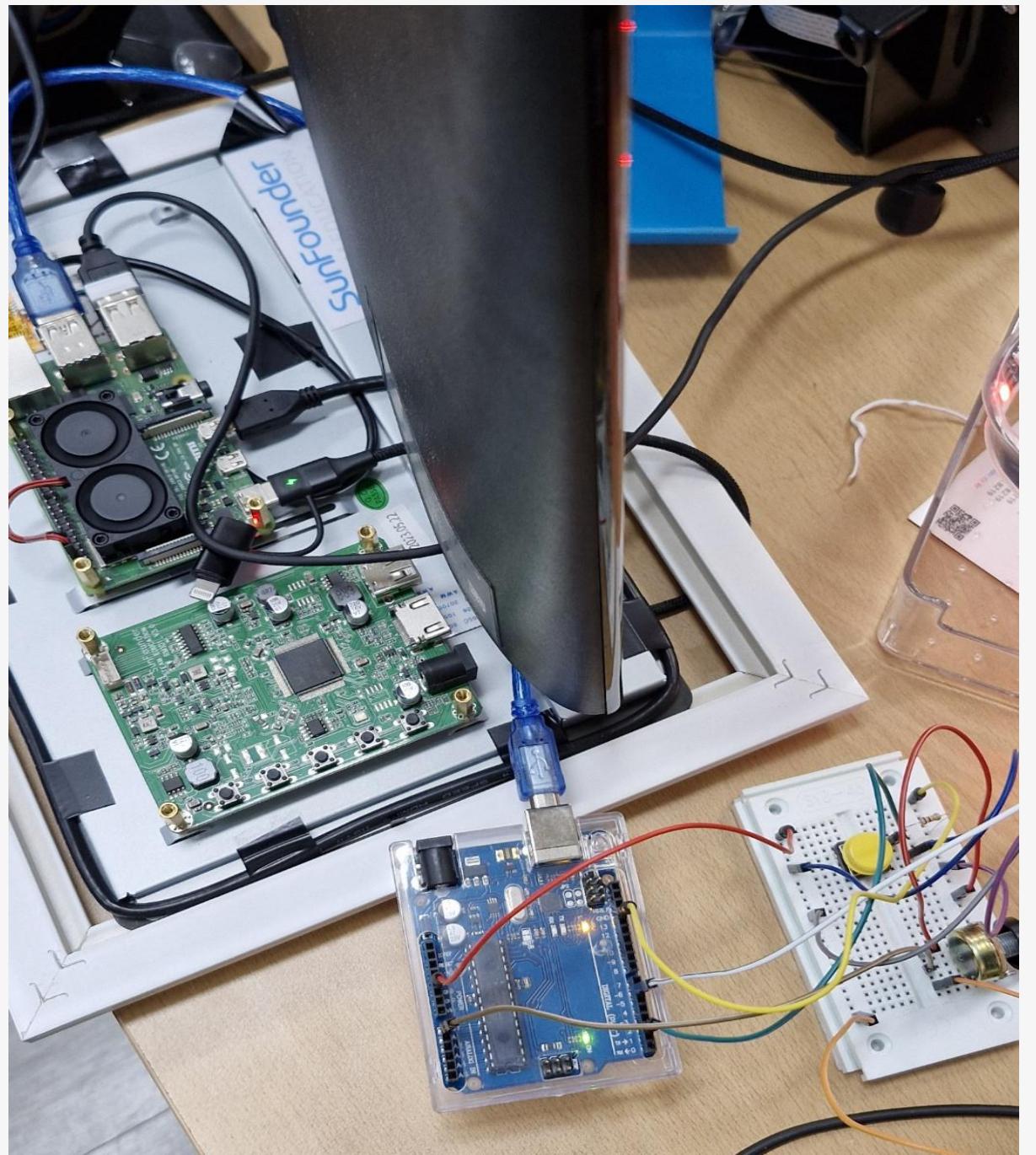
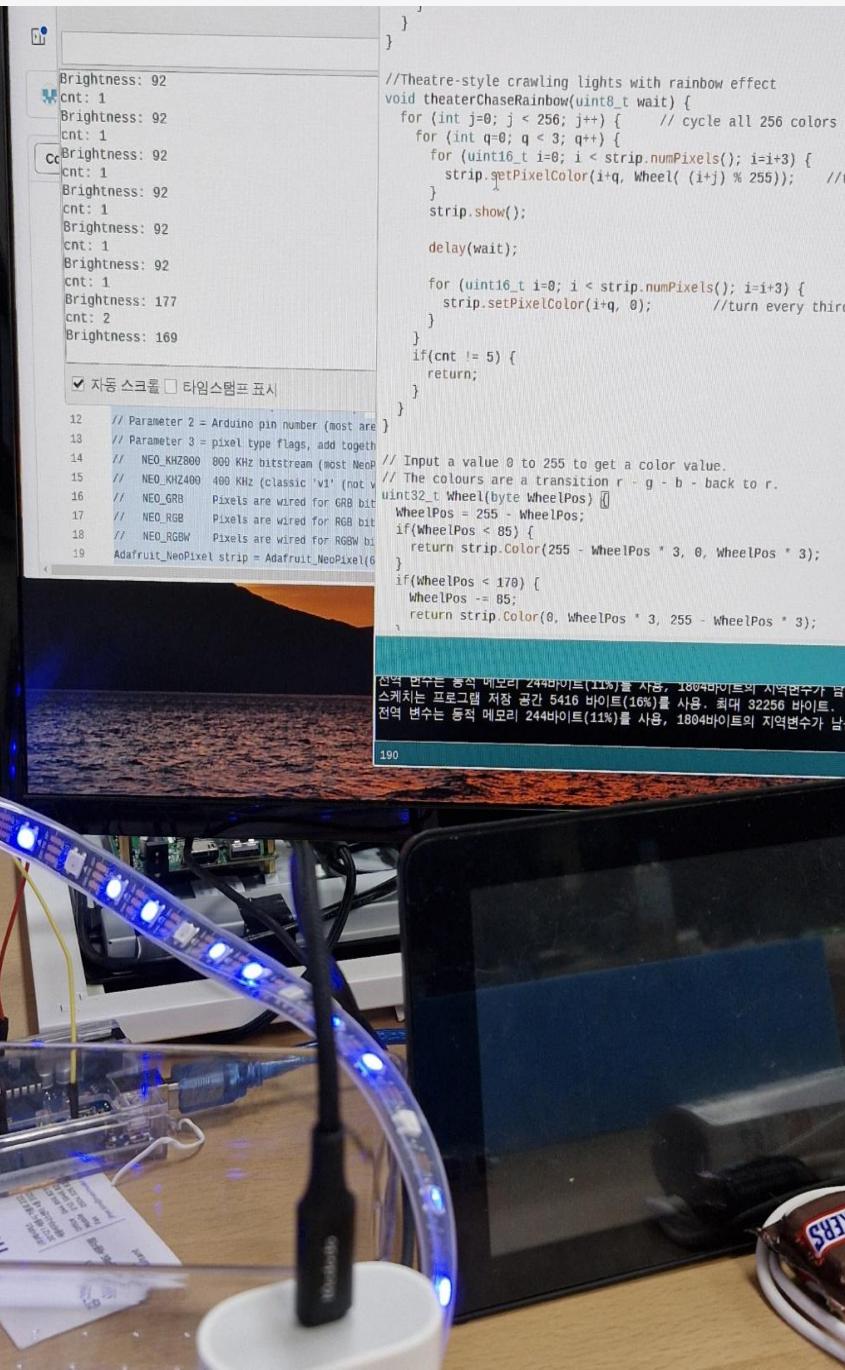
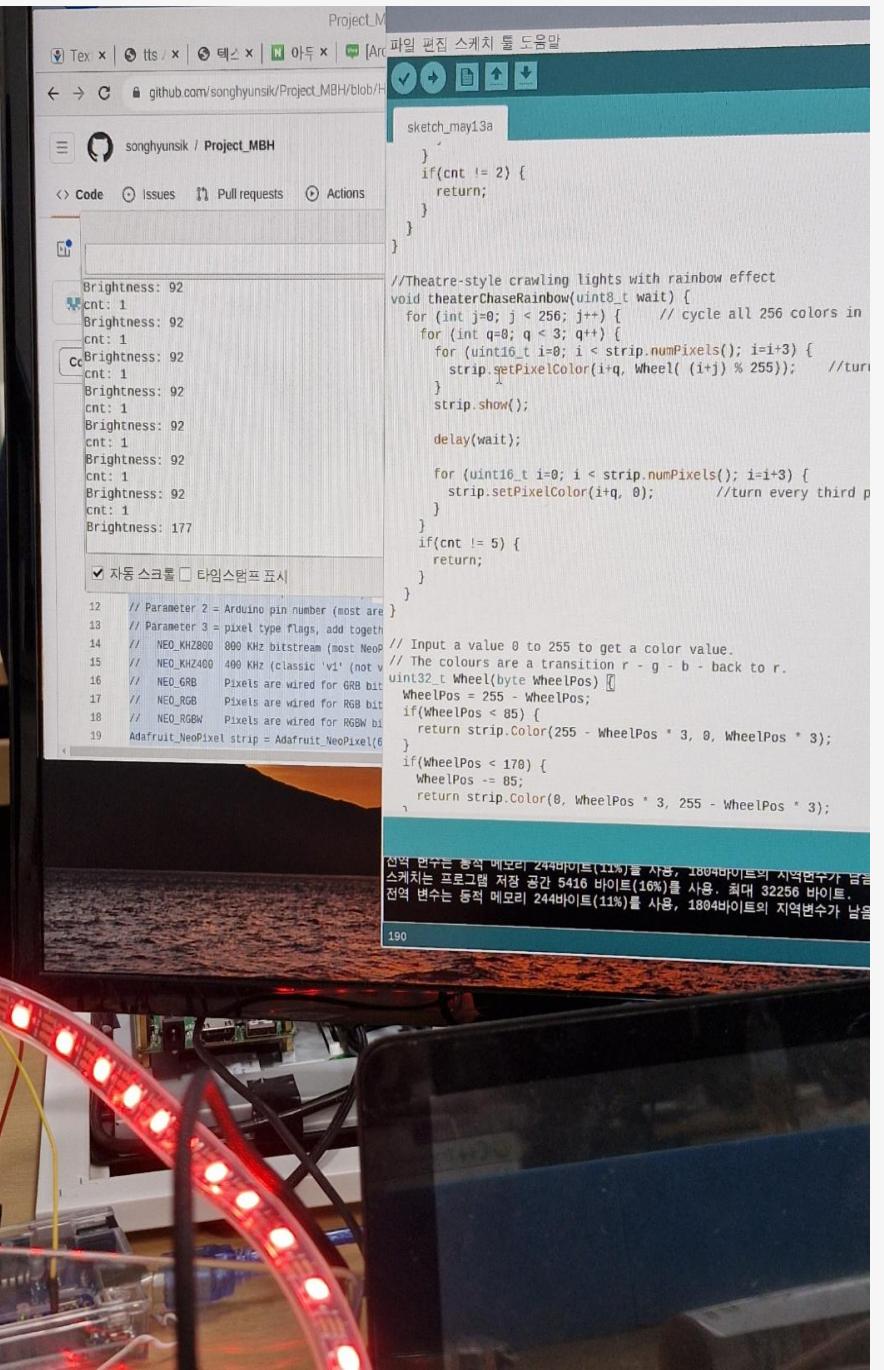
```
project > openCV > C: encoding.cpp > convertEUC_KRtoUTF8(std::string)
  1 #include <iostream>
  2 #include <iomanip>
  3 #include <sstream>
  4 #include <fstream>
  5 #include <string>
  6
  7 Codeium: Refactor | Explain | Generate Function Comment | ×
  8 std::string convertEUC_KRtoUTF8(std::string strEUCKR) {
  9     iconv_t conv = iconv_open("UTF-8", "EUC-KR");
 10     if (conv == (iconv_t)-1) {
 11         perror("iconv_open");
 12         return "";
 13     }
 14
 15     size_t inbytesleft, outbytesleft;
 16     inbytesleft = strEUCKR.size();
 17     outbytesleft = inbytesleft * 3;
 18
 19     char *inbuf = &strEUCKR[0];
 20     char *outbuf = new char[outbytesleft];
 21     char *outbufstart = outbuf;
 22
 23     if (iconv(conv, &inbuf, &inbytesleft, &outbuf, &outbytesleft) ==
 24         (size_t)-1) {
 25         perror("iconv");
 26         return "";
 27     }
 28
 29     std::string strUTF8(outbufstart, outbuf - outbufstart);
 30     delete[] outbufstart;
 31
 32     iconv_close(conv);
 33
 34     return strUTF8;
 35 }
 36 // 5번째 줄 12번, 14번, 16번에 있는 단어 추출
 37 Codeium: Refactor | Explain | ×
 38 int main() {
 39     std::ifstream inputFile("output1.txt"); // 파일 경로를 수정하세요
 40     std::string line;
 41     std::string temperature; // 추출한 단어를 저장할 변수
 42     std::string humidity; // 추출한 단어를 저장할 변수
 43     std::string precipitation; // 추출한 단어를 저장할 변수
 44
 45     int currentLine = 0;
 46     while (std::getline(inputFile, line)) {
 47         currentLine++;
 48
 49         // 5번째 줄인 경우
 50         if (currentLine == 12) {
 51             std::istringstream iss(line);
 52             std::string word;
 53             int wordCount = 0;
 54
 55             // 공백을 구분자로 하여 단어 추출
 56             while (iss >> word) {
 57                 wordCount++;
 58                 // 12번째 단어
 59                 if (wordCount == 12) {
 60                     // temperature = word;
 61                     continue;
 62                 }
 63                 // 14번째 단어
 64                 if (wordCount == 14) {
 65                     // humidity = word;
 66                     continue;
 67                 }
 68                 // 16번째 단어
 69                 if (wordCount == 5) {
 70                     precipitation = word;
 71                     break;
 72                 }
 73             }
 74             break;
 75         }
 76     }
 77
 78     // 단어를 출력
 79     // std::cout << "현재 세종시 기온은 " <<temperature << "도입니다." <<
 80     // std::endl; std::cout << "현재 세종시 습도은 " <<humidity << "도입니다."
 81 }
```



# 16일 목요일 프로젝트 진행 과정

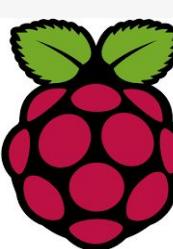
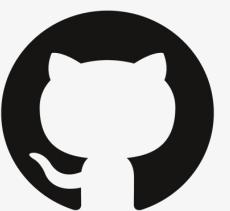


## ● 아두이노 uno LED 테스트



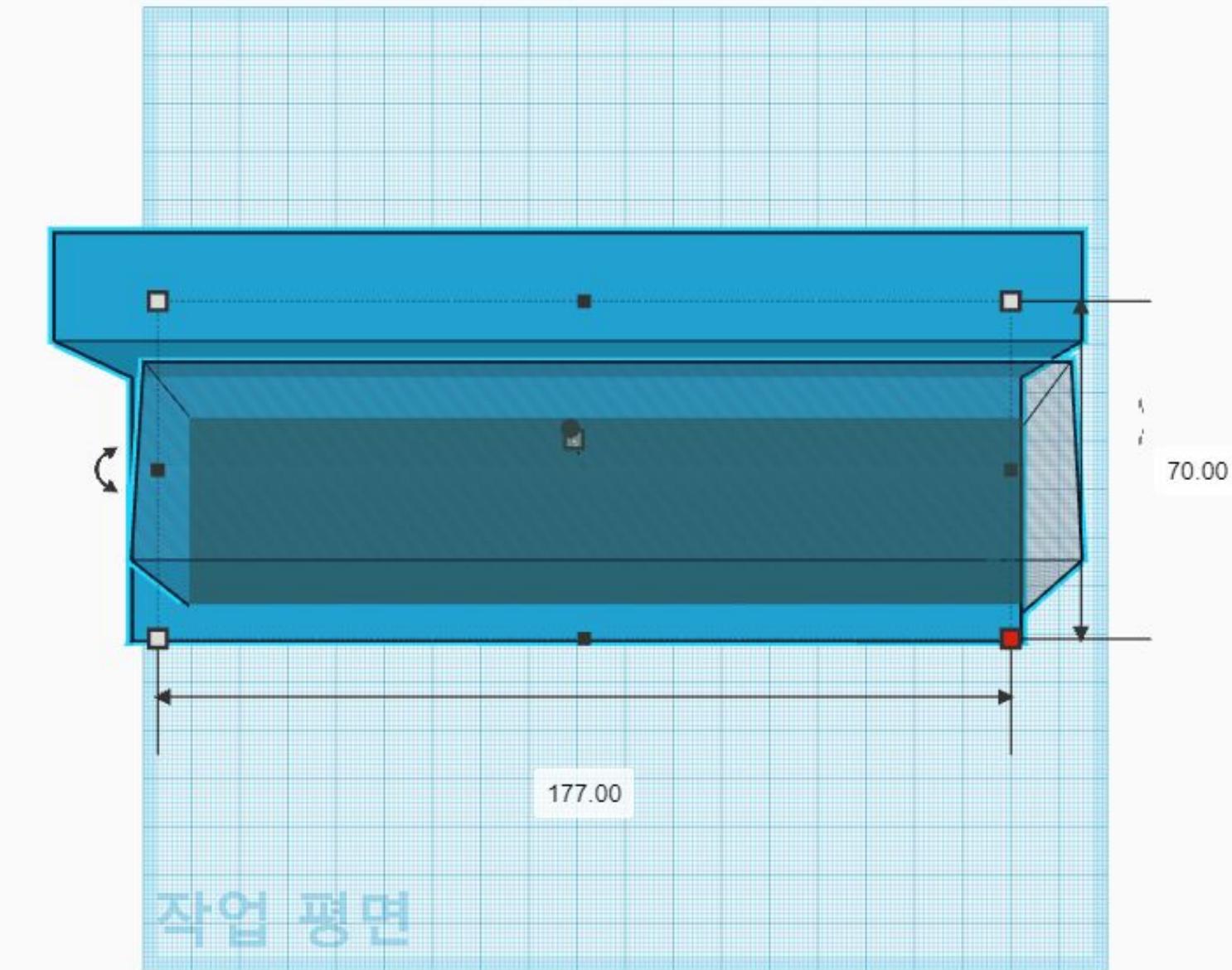
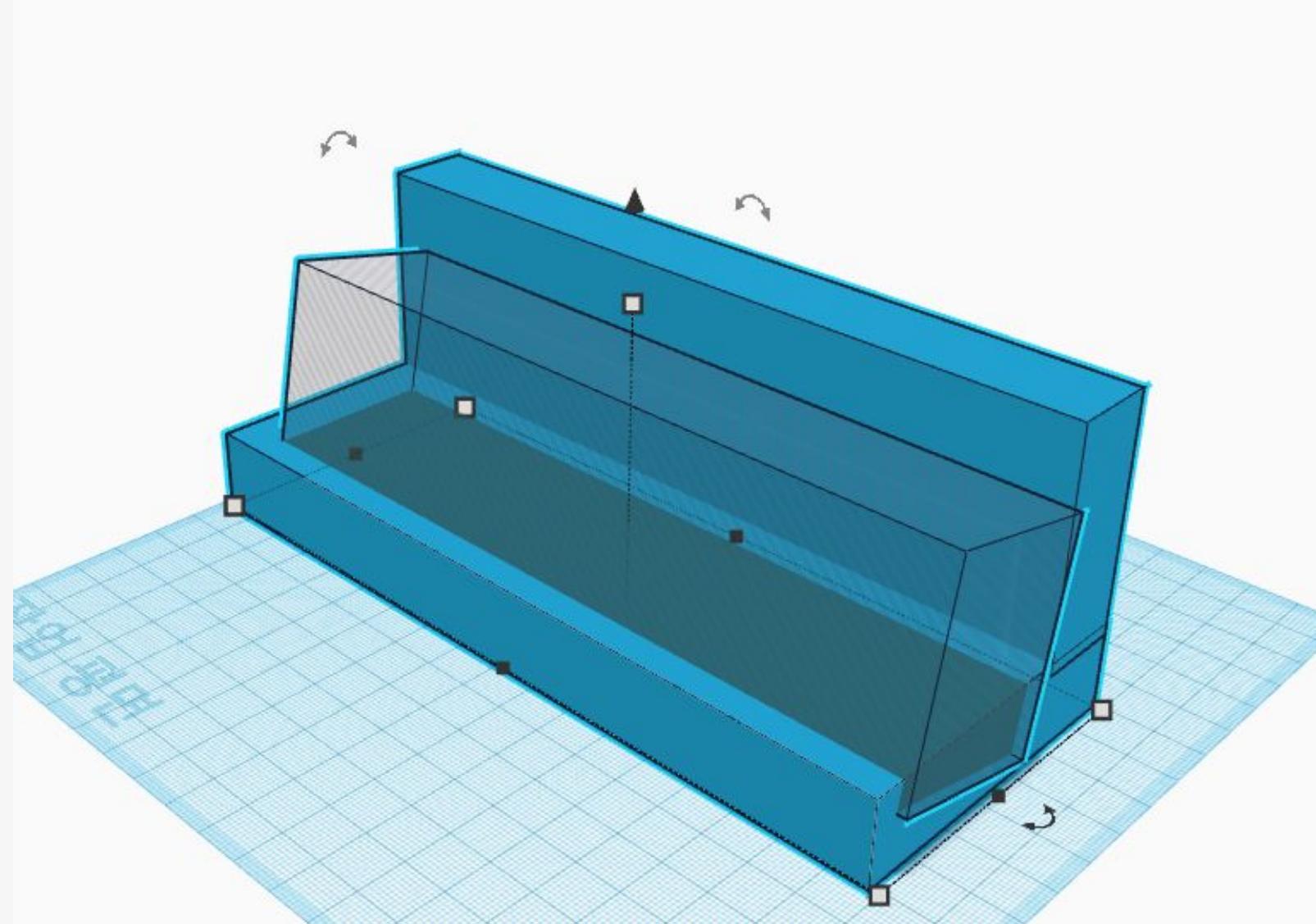


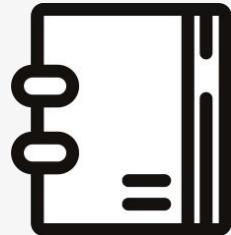
## 16일 목요일 프로젝트 진행 과정



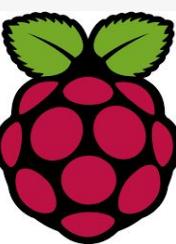
- 액자사이즈를 다시 측정해 가로 175로 늘려서 새롭게 디자인해 출력

### 3D PRINT

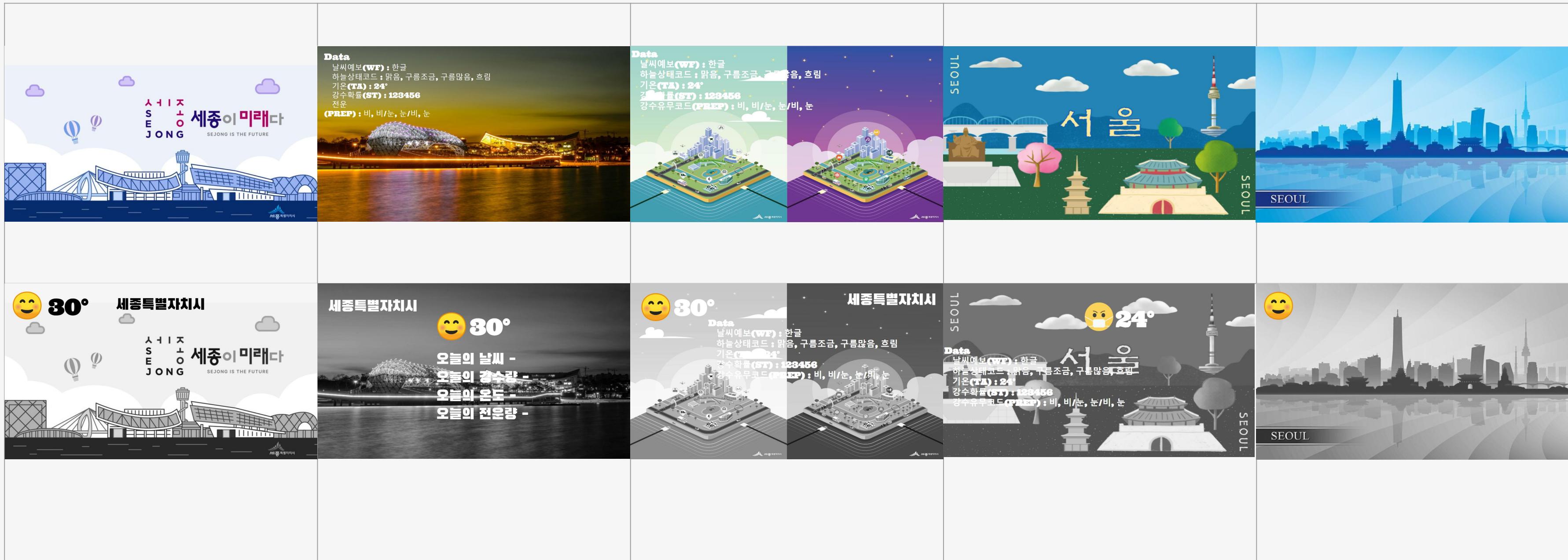


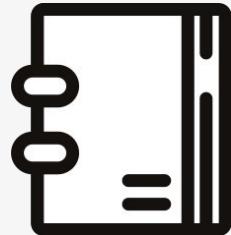


# 16일 목요일 프로젝트 진행 과정

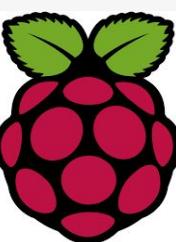
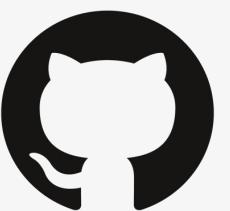


## ● 디자인 시안 제작 [ 지역4곳 + Grayscale ]

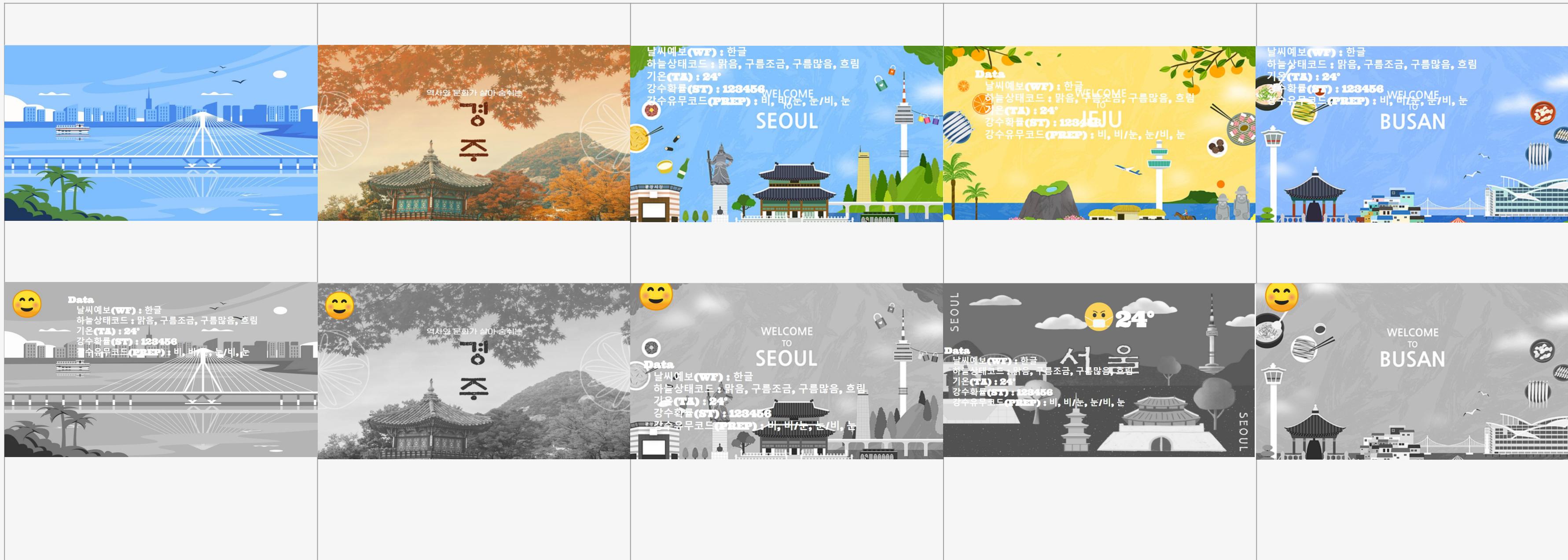


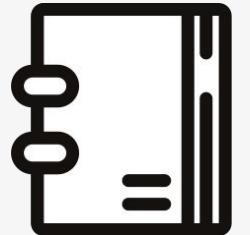


# 16일 목요일 프로젝트 진행 과정

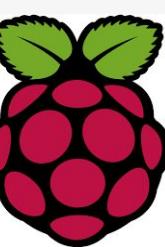
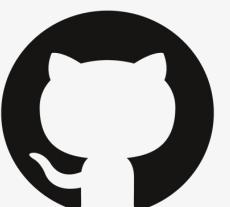


## ● 디자인 시안 제작 [ 지역4곳 + Grayscale ]





# 16일 목요일 프로젝트 진행 과정



## ● OpenCV 관련

라이브러리

전체 설치

The screenshot shows a Linux desktop environment with a dark theme. On the left, there's a dock with various icons. In the center, a terminal window is open with the following command and output:

```
matt@Rio:~$ sudo apt-get install libopencv-dev
matt@Rio:~$ sudo apt-get install tesseract-ocr
matt@Rio:~$ sudo apt-get install libtesseract-dev
```

Below the terminal, a message from ChatGPT says: "4. Tesseract OCR을 사용하여 텍스트를 추출합니다. 5. 추출된 텍스트와 텍스트 영역을 화면에 출력합니다. 이 코드를 실행하기 위해서는 OpenCV와 Tesseract OCR 라이브러리가 설치되어 있어야 합니다. 설치 방법은 다음과 같습니다:"

**OpenCV 설치:**

```
sh
sudo apt-get install libopencv-dev
```

**Tesseract 설치:**

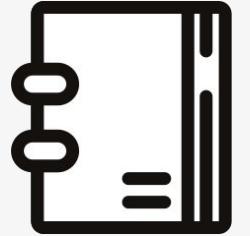
```
sh
sudo apt-get install tesseract-ocr
sudo apt-get install libtesseract-dev
```

At the bottom, a message from ChatGPT says: "이제 위의 코드를 컴파일하고 실행하면 이미지에서 중앙 텍스트를 추출할 수 있습니다."

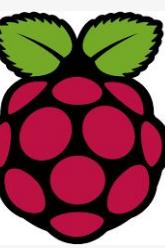
The terminal window also shows the results of the installations:

```
matt@Rio:~$ sudo apt-get install libopencv-dev
matt@Rio:~$ sudo apt-get install tesseract-ocr
matt@Rio:~$ sudo apt-get install libtesseract-dev
```

The desktop background is a purple and red gradient.



# 16일 목요일 프로젝트 진행 과정



## ● 서울이미지

텍스트 추출

코드

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Rio [실행 중] - Oracle VM VirtualBox, 5월 16일 21:29
- File Explorer:** Shows the project structure: PROJECT\_MBH, .vscode, project, data, openCV, build, and various configuration files like settings.json, launch.json, and tasks.json.
- Code Editor:** The main.cpp file contains C++ code for image processing using OpenCV and Tesseract. It includes comments in Korean explaining the steps: reading an image, converting it to grayscale, binarizing it, finding contours, and identifying the largest contour.
- Terminal:** Shows error messages related to the #include directive for opencv2/opencv.hpp.
- Status Bar:** Includes file navigation icons, Git status, and system information like date and time.

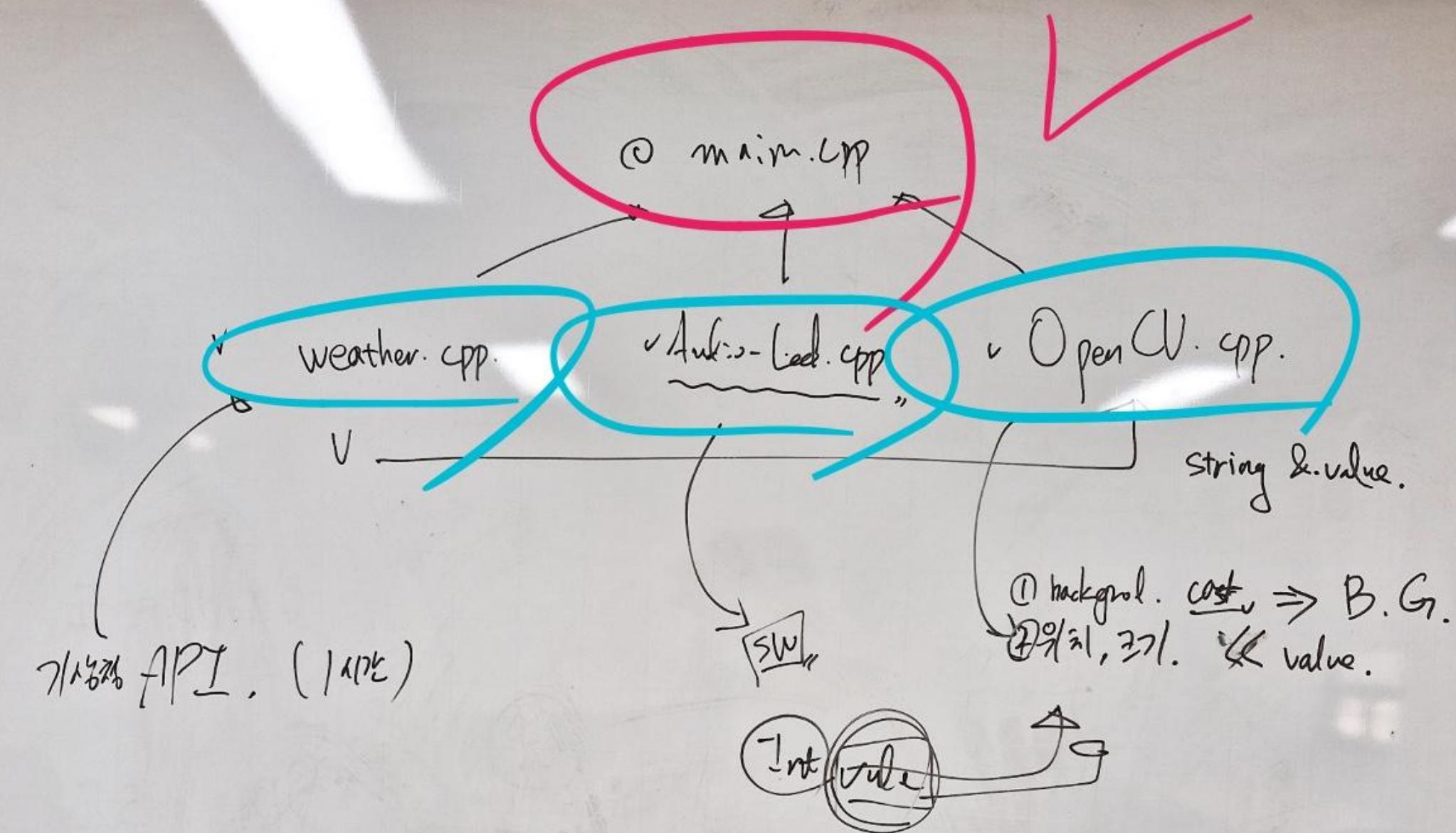
Day 4

마무리

단계 전,

회의

자료

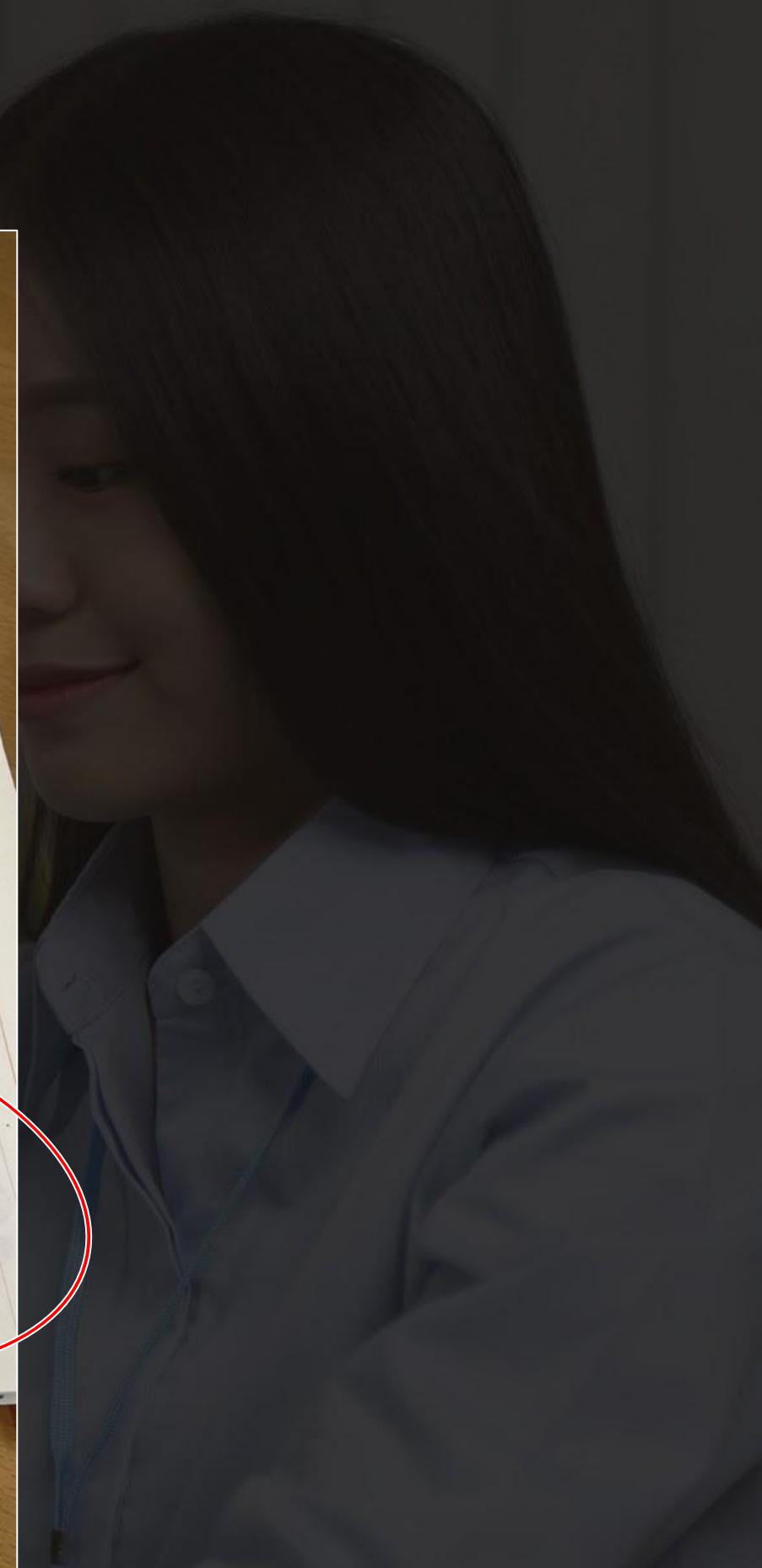
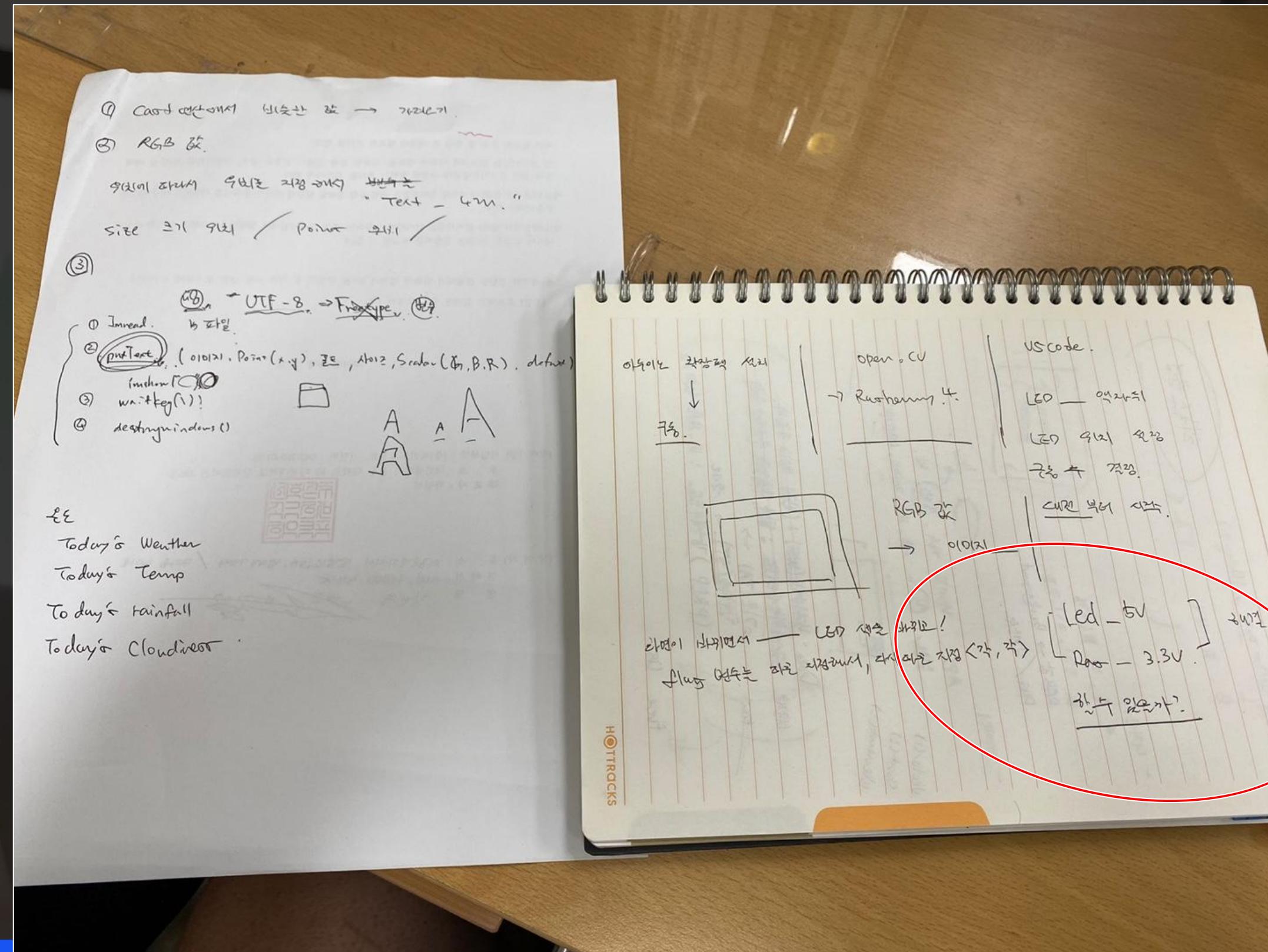


D.S.

while( $\Rightarrow$  Display.  
( if([SW]{LED, V})  
if(min == 0) )

# Day 4

## 회의 자료



# 3가지 주요 목표 [금]

날씨 데이터 실시간 받기  
코드 병합 [main.cpp]

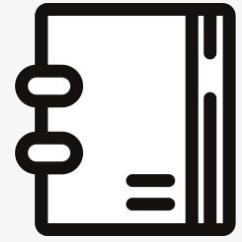


아두이노UNO  
LED 실행가능 여부  
[WS2812B LED STRIP]

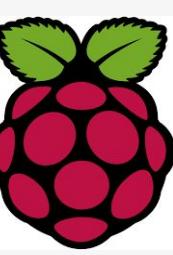
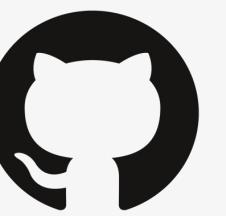


3D 목업 제작 완성  
[ w/ 우드스티커 ]



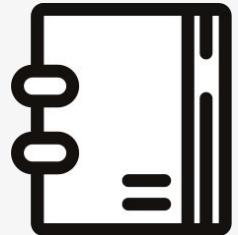


17일 금요일 프로젝트 진행 과정



## • 거치대MOCK-UP 우드처리 완료





# 17일 금요일 프로젝트 진행 과정

## MAIN.CPP



```
text.cpp U final.cpp U main.cpp 4, M encoding.cpp U README.md M
project > openCV > main.cpp > weather
1 #include <curl/curl.h>
2 #include <stdio.h>
3 #include <string>
4 #include <iostream>
5 #include <fstream>
6 #include <sstream>
7 #include <X11/Xlib.h>
8 #include "opencv2/opencv.hpp"
9 using namespace std;
10 using namespace cv;
11
12 String folder = "/home/hrd/Desktop/Project_MBH/data/";
13 string folderPath = "/home/hrd/Desktop/Project_MBH/data/";
14 int serial_num = 3;
15
16 Codeium: Refactor | Explain
17 class weather
18 {
19 private:
20     const char* spacedata;
21     const char* filename;
22     string imgfile;
23
24 public:
25     string ta;           // 기온
26     string rn;          // 강수량
27     string hm;          // 습도
28     string ca_tot;      // 전운량
29     string name;
30     weather(const char* data, const char* file, string name_value, string fileimg);
31
32     void outputtxt();    // API 데이터 입력
33     void findtxt();      // 필요 데이터 변수 저장
34     void OpenCV();
35     void init();
36 };
37
```

```
int main() {
    // set class 지역멤버(const char* url, const char* filename, string 지역이름, string 이미지파일이름)
    weather seoul("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=108&help=0&authKey=FyIoXmJzSiWiKF5icxolng", "seoul.txt", "SEOUL", "seoul.png"); // 서울: 108
    weather busan("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=159&help=0&authKey=FyIoXmJzSiWiKF5icxolng", "busan.txt", "BUSAN", "busan.png"); // 부산: 159
    weather daejeon("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=133&help=0&authKey=FyIoXmJzSiWiKF5icxolng", "daejeon.txt", "DAEJEON", "daejun.png"); // 대전: 133
    weather jeju("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=184&help=0&authKey=FyIoXmJzSiWiKF5icxolng", "jeju.txt", "JEJU", "jeju.jpg"); // 제주: 184

    // 현재시간 확인
    time_t now = time(nullptr);
    tm* current_time = localtime(&now);

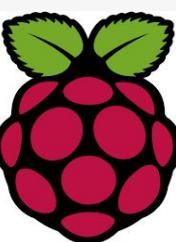
    // 시간 정수형 변환
    // string year,mon,hour,min;
    // year = to_string(current_time->tm_year + 1900);
    // mon = to_string(current_time->tm_mon + 1);
    // hour = to_string(current_time->tm_hour);
    // min = to_string(current_time->tm_min);

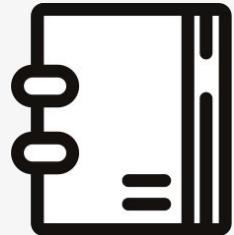
    while(1){
        // set weather value from 기상청 API
        seoul.init();
        busan.init();
        daejeon.init();
        jeju.init();

        // 정각마다 기상청 API 데이터 revalue
        if(current_time->tm_hour == 0){
            seoul.init();
            busan.init();
            daejeon.init();
            jeju.init();
        }

        switch (serial_num)
        {
        case 1:
            seoul.OpenCV();
            continue;
        case 2:
            busan.OpenCV();
            continue;
        case 3:
            daejeon.OpenCV();
            continue;
        case 4:
            jeju.OpenCV();
            continue;

        default :
            destroyAllWindows();
        }
    }
}
```





# 17일 금요일 프로젝트 진행 과정

## MAIN.CPP



```
Codeium: Refactor | Explain | Generate Function Comment | ×
weather::weather(const char* data, const char* file, string name_value, string fileimg){
    name = name_value;
    spacedata = data;
    filename = file;
    imgfile = fileimg;
}

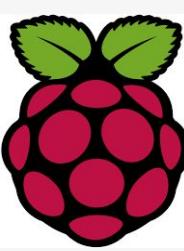
Codeium: Refactor | Explain | Generate Function Comment | ×
void weather::outputtxt()
{
    CURL* curl = curl_easy_init();
    if(curl){
        FILE* fp = fopen(filename, "wb");
        const char* url = spacedata ;
        curl_easy_setopt(curl, CURLOPT_URL, url); // URL 설정
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp); // 데이터 쓰기 설정
        CURLcode res = curl_easy_perform(curl); // 데이터 다운로드 수행
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res)); // 오류 처리
        }
        curl_easy_cleanup(curl); // CURL 정리
        fclose(fp); // 파일 닫기
    }
}
```

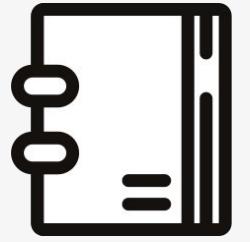
```
Codeium: Refactor | Explain | Generate Function Comment | ×
void weather::findtxt()
{
    ifstream inputFile(filename); // 파일 경로를 수정하세요
    string line;

    // 파일을 한 줄씩 읽어서 특정 줄과 특정 단어를 추출
    int currentLine = 0;
    while (std::getline(inputFile, line)) {
        currentLine++;

        // 5번째 줄인 경우
        if (currentLine == 5) {
            std::istringstream iss(line);
            std::string word;
            int wordCount = 0;

            // 공백을 구분자로 하여 단어 추출
            while (iss >> word) {
                wordCount++;
                // 12번째 단어
                if (wordCount == 12) {
                    ta = word;
                    continue;
                }
                // 14번째 단어
                if (wordCount == 14) {
                    hm = word;
                    continue;
                }
                // 16번째 단어
                if (wordCount == 16) {
                    rn = word;
                    continue;
                }
                // 28번째 단어
                if (wordCount == 26) {
                    ca_tot = word;
                    break;
                }
            }
            break;
        }
    }
}
```



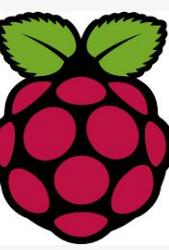
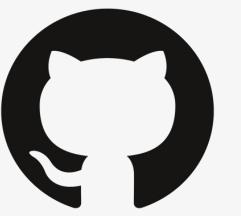


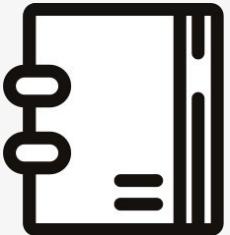
# 17일 금요일 프로젝트 진행 과정

## MAIN.CPP



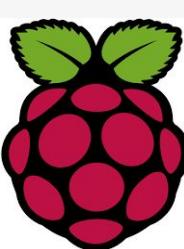
```
173     void weather::openCV() {
174         Display* display = XOpenDisplay(NULL);
175         if (display == NULL) {
176             std::cerr << "Cannot open display" << std::endl;
177         }
178
179         Screen* screen = DefaultScreenOfDisplay(display);
180         int screen_width = screen->width;
181         int screen_height = screen->height;
182
183         Mat img = imread(folderPath + imgfile);      //Scalar(220, 245, 245) : 베이지색
184         putText(img, name, Point(50, 30), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 3.0);
185         // putText(img, "Today's Weather : ", Point(50, 60), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
186         putText(img, "Today's Temperature : " + ta, Point(50, 90), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
187         putText(img, "Today's Rainfall : " + rn, Point(50, 120), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
188         putText(img, "Today's Total Cloudiness : " + ca_tot, Point(50, 150), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
189
190         // 이모지 이미지 불러오기
191         Mat emoji = imread(folderPath + "smile.png", IMREAD_COLOR);
192         if (emoji.empty()) {
193             cerr << "이모지 이미지 불러오기 실패!" << endl;
194         }
195         // 이모지 이미지 크기 조정
196         double scale = 0.04;
197         Mat resizedEmoji;
198         resize(emoji, resizedEmoji, Size(), scale, scale);
199
200         int emojiX = 340;
201         int emojiY = 30;
202
203
204         Rect roi(Point(emojiX, emojiY), resizedEmoji.size());
205         resizedEmoji.copyTo(img(roi), resizedEmoji);
206
207         if (img.cols > screen_width || img.rows > screen_height) {
208             double scale_width = static_cast<double>(screen_width) / img.cols;
209             double scale_height = static_cast<double>(screen_height) / img.rows;
210             double scale = std::min(scale_width, scale_height);
211
212             resize(img, img, Size(), scale, scale);
213         }
214
215         imshow("img", img);
216         waitKey(0);
217
218         XCLOSEDISPLAY(display);
219     }
220
221     void weather::init(){
222         outputtxt();
223         findtxt();
224     }
```





# 17일 금요일 프로젝트 진행 과정

# 아두이노 MAIN.CPP



The screenshot shows a Visual Studio Code window titled "Project\_MBH.ino - Project\_MBH - Visual Studio Code". The main area displays the following Arduino code:

```
#include <Adafruit_NeoPixel.h>
#ifndef __AVR__
#include <avr/power.h>
#endif

#define PIN 6
#define SW 2 // 버튼 연결 핀
#define BRIGHTNESS_PIN A0 // 가변 저항 연결 핀

int brightnessValue = 0; // 현재 밝기 값
int brightnessChanged = false; // 이전 밝기 값

volatile int cnt = 0; // 인터럽트에 의해 변경되므로 volatile 선언

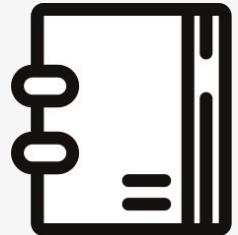
Adafruit_NeoPixel strip = Adafruit_NeoPixel(60, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    #if defined (__AVR_ATtiny85__)
    if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
    #endif

    pinMode(SW, INPUT_PULLUP); // 버튼 핀을 풀업 입력 모드로 설정

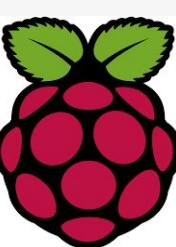
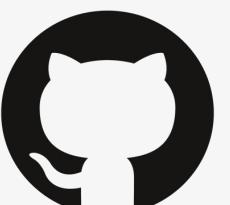
    strip.begin();
    strip.setBrightness(50);
    Off(strip.Color(0, 0, 0), 50); // Off
    strip.show(); // 모든 픽셀이 꺼져있는 상태로 출력
    Serial.begin(115200);

    // 인터럽트 설정 (버튼 핀에 FALLING 모드 설정)
    attachInterrupt(digitalPinToInterrupt(SW), buttonISR, FALLING);
}
```



17일 금요일 프로젝트 진행 과정

# 아두이노 MAIN.CPP



Ubuntu0222 - VMware Workstation 17 Player (Non-commercial use only)

Player

현재 활동 Visual Studio Code

Project\_MBH.ino - Project\_MBH - Visual Studio Code

5월 17일 16:33

Project\_MBH.ino

File 편집 선택 영역 보기 이동 실행 터미널 도움말

Project\_MBH.ino > ...

33 }

34

35 void loop() {

36 brightnessValue = map(analogRead(BRIGHTNESS\_PIN), 0, 1023, 0, 255); // 아날로그 입력 값을 밝기로 변환

37 strip.setBrightness(brightnessValue); // NeoPixel의 밝기를 설정

38 Serial.print("Mode: ");

39 Serial.println(cnt);

40 // 현재 밝기 값이 이전 값과 다르면 작업을 수행

41 if (brightnessChanged) {

42 // NeoPixel의 밝기를 설정

43 strip.setBrightness(brightnessValue);

44

45 // 이전 밝기 값을 업데이트

46 brightnessChanged = false;

47 }

48 // 변경된 밝기 값 출력

49 Serial.print("Brightness: ");

50 Serial.println(brightnessValue);

51

52 // cnt 값에 따라 LED 패턴 실행

53 switch (cnt) {

54 case 1:

55 BUSAN\_COLOR(strip.Color(0, 0, 127), 50); // Blue

56 break;

57 case 2:

58 DAEJEON\_COLOR(strip.Color(255, 0, 255), 50); // Purple

59 break;

60 case 3:

61 JEJU\_COLOR(strip.Color(255, 255, 0), 50); // Yellow

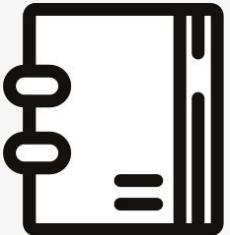
62 break;

63 case 4:

64 SEOUL\_COLOR(strip.Color(0, 0, 127), 50); // Blue

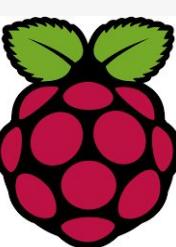
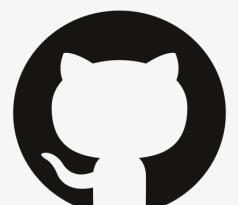
65

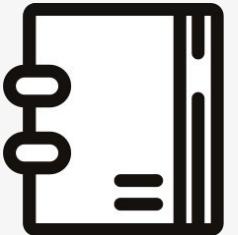
The screenshot shows the Visual Studio Code interface with the Arduino IDE extension. The left sidebar shows project files like README.md, settings.json, and arduino.json. The main editor pane displays the Arduino sketch 'Project\_MBH.ino' with the provided code. The status bar indicates the date and time as '5월 17일 16:33'.



# 17일 금요일 프로젝트 진행 과정

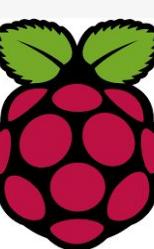
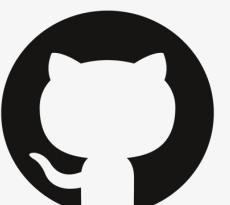
# 아두이노 MAIN.CPP





# 17일 금요일 프로젝트 진행 과정

# 아두이노 MAIN.CPP



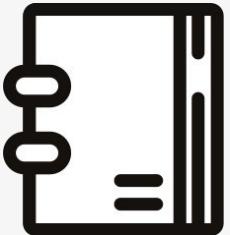
The screenshot shows a Visual Studio Code interface running on an Ubuntu 0.22.2 VM. The title bar indicates the application is running in a VMware Workstation 17 Player. The main area displays an Arduino sketch named 'Project\_MBH.ino'. The code defines four functions: 'Off', 'BUSAN\_COLOR', 'DAEJEON\_COLOR', and 'JEJU\_COLOR'. Each function iterates through all pixels of a strip and sets their color to a specified value ('c') using 'strip.setPixelColor(i, c);'. After setting the color, it calls 'strip.show()' and 'delay(wait)'. The 'Off' function also checks if 'cnt' is not equal to 0 before returning. The project structure on the left shows a folder named 'PROJECT...' containing '.vscode', 'build', and 'RaspberryPi\_Arduino...' subfolders. Inside '.vscode', there are 'arduino.json', 'c\_cpp\_properties.json', 'settings.json', and 'tasks.json' files. The status bar at the bottom shows the date and time as '5월 17일 16 : 34'.

```
90 void Off(uint32_t c, uint8_t wait) {
91     strip.show();
92     delay(wait);
93     if(cnt != 0) {
94         return;
95     }
96 }
97
98
99 }

100 void BUSAN_COLOR(uint32_t c, uint8_t wait) {
101     for(int i=0; i<strip.numPixels(); i++) {
102         strip.setPixelColor(i, c);
103     }
104     strip.show();
105     delay(wait);
106     if(cnt != 1) {
107         return;
108     }
109 }
110

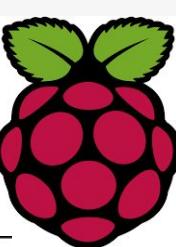
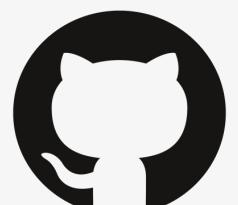
111 void DAEJEON_COLOR(uint32_t c, uint8_t wait) {
112     for(int i=0; i<strip.numPixels(); i++) {
113         strip.setPixelColor(i, c);
114     }
115     strip.show();
116     delay(wait);
117     if(cnt != 2) {
118         return;
119     }
120 }
121

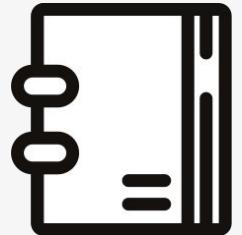
122 void JEJU_COLOR(uint32_t c, uint8_t wait) {
123     for(int i=0; i<strip.numPixels(); i++) {
124         strip.setPixelColor(i, c);
125     }
126 }
```



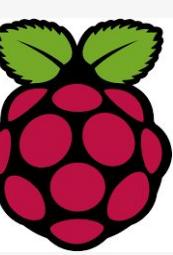
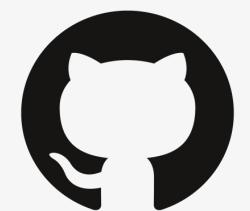
# 17일 금요일 프로젝트 진행 과정

# 아두이노 MAIN.CPP





# 17일 금요일 프로젝트 진행 과정



## • Open CV

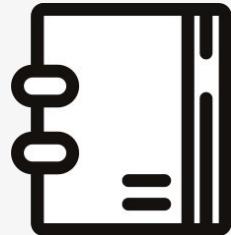
문서, 위치

## MAIN.CPP

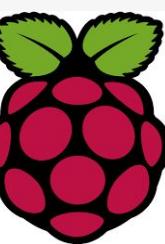
The screenshot shows a code editor with a dark theme. On the left is a file tree for a project named 'PROJECT\_MBH'. The 'main.cpp' file is selected and shown in the main pane. The code uses OpenCV to read an image, add text, and copy an emoji to it.

```
탐색기 ... main.cpp README.md
PROJECT_MBH
  .vscode
    c_cpp_properties.json
    tasks.json
  project
    data
      busan.jpg
      daejun.png
      jeju.jpg
      mask.png
      mydatajson
      output_busan.jpg
      output_daejun.png
      output_jeju.jpg
      output_seoul.jpg
      sad.png
      seoul.jpg
      smile.png
  openCV
    build
    CMakeLists.txt
    main.cpp
  README.md

main.cpp
project > openCV > main.cpp > main()
1 #include "opencv2/opencv.hpp"
2 #include <iostream>
3
4 using namespace cv;
5 using namespace std;
6 string folderPath = "/home/matt/바탕화면/Project_MBH/project/data/";
7
8 int main(){
9
10   Mat img = imread(folderPath + "daejun.png"); //Scalar(220, 245, 245) : 배이지색
11   putText(img, "DAEJEON ", Point(50, 30), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 3.0);
12   putText(img, "Today's Weather : ", Point(50, 60), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
13   putText(img, "Today's Temperature : ", Point(50, 90), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
14   putText(img, "Today's Rainfall : ", Point(50, 120), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
15   putText(img, "Today's Total Cloudiness : ", Point(50, 150), FONT_HERSHEY_SIMPLEX, 1, Scalar(220, 245, 245), 2.5);
16
17   // 이모지 이미지 불러오기
18   Mat emoji = imread(folderPath + "smile.png", IMREAD_COLOR);
19   if (emoji.empty()){
20     cerr << "이모지 이미지 불러오기 실패!" << endl;
21     return -1;
22   }
23   // 이모지 이미지 크기 조정
24   double scale = 0.04;
25   Mat resizedEmoji;
26   resize(emoji, resizedEmoji, Size(), scale, scale);
27
28   int emojiX = 340;
29   int emojiY = 30;
30
31
32   Rect roi(Point(emojiX, emojiY), resizedEmoji.size());
33   resizedEmoji.copyTo(img(roi), resizedEmoji);
34
35   imshow("img", img);
36   waitKey(0);
37
38   return 0;
39 }
```



# 17일 금요일 프로젝트 진행 과정



matt@Rio: ~/바탕화면/Project\_MBH/project/openCV/build  
matt@Rio: ~/바탕화면/Project\_MBH/project/openCV/build 10x23

```
matt@Rio:~/바탕화면/Project_MBH/project/openCV/build$ ./text
```

The terminal output shows:

```
DAEJEON
Today's Weather : 😊
Today's Temperature :
Today's Rainfall :
Today's Total Cloudiness :
```

A red circle highlights the weather information in the terminal.

text.cpp - Project\_MBH - Visual Studio Code

파일 편집 선택 영역 보기 이동 실행 터미널 도움말

text.cpp u x encoding.cpp u README.md ●

PROJECT\_MBH  
.vscode  
c\_cpp\_properties.json  
tasks.json  
project  
data  
busan.jpg  
daejun.png  
mask.png  
mydata.json  
output\_busan.jpg  
output\_daejun.png  
output\_jeju.jpg  
output\_seoul.jpg  
sad.png  
seoul.jpg  
smile.png  
openCV  
build  
CMakeLists.txt  
encoding.cpp u  
text.cpp u  
README.md

```
project > openCV > text.cpp > main()
1 #include "opencv2/opencv.hpp"
2 #include <iostream>
3
4 using namespace cv;
5 using namespace std;
6 string folderPath = "/home/matt/바탕화면/Project_MBH/project/data/";
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
```

Codeium: Refactor | Explain | Generate Function Comment | ×

```
int main(){
    Mat img = imread(folderPath + "daejun.png"); //Scalar(220, 245
    putText(img, "DAEJEON ", Point(50, 30), FONT_HERSHEY_SIMPLEX, 1, Sc
    putText(img, "Today's Weather : ", Point(50, 60), FONT_HERSHEY_SIMP
    putText(img, "Today's Temperature : ", Point(50, 90), FONT_HERSHEY_
    putText(img, "Today's Rainfall : ", Point(50, 120), FONT_HERSHEY_SI
    putText(img, "Today's Total Cloudiness : ", Point(50, 150), FONT_HE
    // 이모지 이미지 불러오기
    Mat emoji = imread(folderPath + "smile.png", IMREAD_COLOR);
    if (emoji.empty()){
        cerr << "이모지 이미지 불러오기 실패!" << endl;
        return -1;
    }
    // 이모지 이미지 크기 조정
    double scale = 0.04;
    Mat resizedEmoji;
    resize(emoji, resizedEmoji, Size(), scale, scale);

    int emojiX = 340;
    int emojiY = 30;

    Rect roi(Point(emojiX, emojiY), resizedEmoji.size());
    resizedEmoji.copyTo(img(roi), resizedEmoji);

    imshow("img", img);
    waitKey(0);

    return 0;
}
```

줄 38, 열 18 공백: 4 UTF-8 LF () C++ Codeium: [...]

Day 5  
회의  
자료

1. Full Screen C2 인터넷 넣어주기 → OK ✓
  2. LED 고정 → Display 시험을 넣기 쉬운 것 → OK ✓  
시내의 표면 + 스케치 둘 (Error) 해결
  3. 프로그램 / 1분마다 →  
/ 데이터 1시간마다 저장  
쓰기하지! 14
- 
- 기록 마수 2 + 사진 영상 (max 1분)

# 3가지 주요 목표 [월]

날씨 데이터 실시간 받기  
이미지 구축 + 텍스트 표현

코드 병합 [main.cpp]



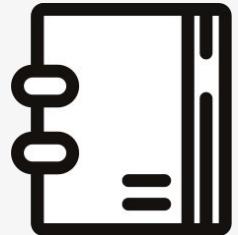
아두이노UNO  
LED 실행가능 여부

[WS2812B LED STRIP]

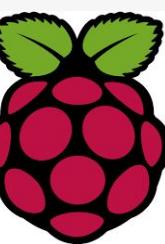
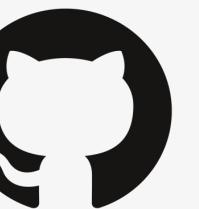


마지막 데코  
+  
사전 영상제작





## 20일 월요일 프로젝트 진행 과정



- 디스플레이 사이즈 맞게 이미지 **RESIZE** 코드 생성
- **TIME** 시간이 실시간 넘어가게 에러코드 수정

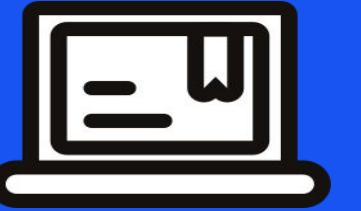
**FULL SCREEN  
RESIZE**

```
// 시간 데이터 화면 표기
putText(img, H+" : "+M, Point(400, 30), FONT_HERSHEY_SIMPLEX, 0.8, Scalar(220, 245, 245), 2.8);

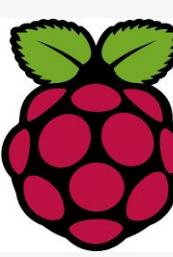
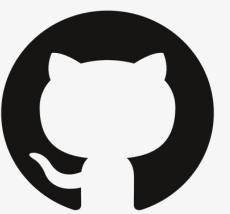
// window fullscreen resize
string windowName = "original image";
namedWindow(windowName,WINDOW_NORMAL);
setWindowProperty(windowName,WND_PROP_FULLSCREEN,WINDOW_FULLSCREEN);
resizeWindow(windowName,1280,800); // raspberry pi 4 display size = (1280,800)

// print display
imshow(windowName, img);
```

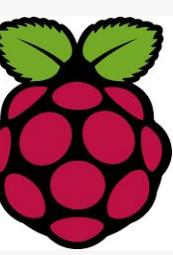
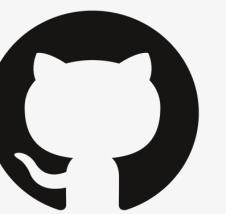
# FINAL.CPP



```
text.cpp 0 final.cpp 2,0 X
project > openCV > final.cpp > onMouse(int, int, int, int, void *)
1 #include <curl/curl.h>
2 #include <stdio.h>
3 #include <string>
4 #include <iostream>
5 #include <fstream>
6 #include <sstream>
7 #include <ctime>
8 #include <iostream>
9 #include "opencv2/opencv.hpp"
10
11 using namespace std;
12 using namespace cv;
13
14 int cnt = 0;
15 int button = 0;
16
17 void onMouse(int event, int x, int y, int flags, void* userdata) {
18     if (event == EVENT_RBUTTONDOWN) {
19         cnt++;
20         button = cnt %4;
21     }
22 }
23
24 string folderPath = "/home/pi/hrd/Desktop/Project_MBH/data/";
25
26 class weather
27 {
28 private:
29     const char* spacedata;
30     const char* filename;
31     string imgfile;
32
33 public:
34     Mat img;
35     string ta;           // 기온
36     string rn;           // 강수량
37     string hm;           // 습도
38     string ca_tot;       // 전운량
39     string name;
40     weather(const char* data, const char* file, string name_value, string fileimg);
41
42     void outputtxt();    // API 데이터 입력
43     void findtxt();      // 필요 데이터 변수 저장
44     void OpenCV(string H, string M, string window);
45     void init();
46 }
```



# FINAL.CPP



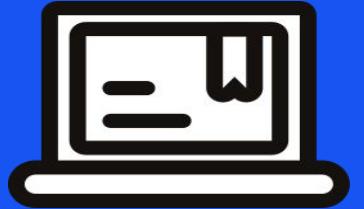
```
int main() {
    // set 지역 (const char* url, const char* filename, string 지역이름, string 이미지파일이름)
    // 서울: 108, 부산: 159, 대전: 133, 제주: 184
    weather seoul("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=108&help=0&authKey=FyIoXmJzSiWiKF5icxolng","seoul.txt","SEOUL","seoul.jpg");
    weather busan("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=159&help=0&authKey=FyIoXmJzSiWiKF5icxolng","busan.txt","BUSAN","busan.jpg");
    weather daejeon("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=133&help=0&authKey=FyIoXmJzSiWiKF5icxolng","daejeon.txt","DAEJEON","daejeon.jpg");
    weather jeju("https://apihub.kma.go.kr/api/typ01/url/kma_sfctm2.php?tm=0&stn=184&help=0&authKey=FyIoXmJzSiWiKF5icxolng","jeju.txt","JEJU","jeju.jpg");

    string hour,min;
    // set window _ mouse
    string windowName = "original image";
    namedWindow(windowName,WINDOW_NORMAL);
    setMouseCallback(windowName,onMouse);
    setWindowProperty(windowName,WND_PROP_FULLSCREEN,WINDOW_FULLSCREEN);
    resizeWindow(windowName,1280,800); // raspberry pi 4 display size = (1280,800)

    // init : set weather value from 기상청 API
    seoul.init();
    busan.init();
    daejeon.init();
    jeju.init();

    while(1){
        cout << button << endl;
        switch(button){
            case 0:{}
                time_t now = time(nullptr);
                tm* current_time = localtime(&now);
                min = to_string(current_time->tm_min);
                hour = to_string(current_time->tm_hour);
                if(current_time->tm_min == 0){
                    seoul.init();
                }
                seoul.OpenCV(hour,min,windowName);
                waitKey(1000);
                break;
        }
    }
}
```

# FINAL.CPP



```
project > openCV > C: final.cpp > main()
49 int main() {
50     seoul.init();
51
52     }
53     seoul.OpenCV(hour,min,windowName);
54     waitKey(1000);
55     break;
56 }
57 case 1:{
58     time_t now = time(nullptr);
59     tm* current_time = localtime(&now);
60     min = to_string(current_time->tm_min);
61     hour = to_string(current_time->tm_hour);
62     if(current_time->tm_min == 0){
63         busan.init();
64     }
65     busan.OpenCV(hour,min,windowName);
66     waitKey(1000);
67     break;
68 }
69 case 2:{
70     time_t now = time(nullptr);
71     tm* current_time = localtime(&now);
72     min = to_string(current_time->tm_min);
73     hour = to_string(current_time->tm_hour);
74     if(current_time->tm_min == 0){
75         daejeon.init();
76     }
77     daejeon.OpenCV(hour,min,windowName);
78     waitKey(1000);
79     break;
80 }
81
82 case 3:{
83     time_t now = time(nullptr);
84     tm* current_time = localtime(&now);
85     min = to_string(current_time->tm_min);
86     hour = to_string(current_time->tm_hour);
87     if(current_time->tm_min == 0){
88         jeju.init();
89     }
90     jeju.OpenCV(hour,min,windowName);
91     waitKey(1000);
92     break;
93 }
94 }
95
96 return 0;
```

Codeium: Refactor | Explain | Generate Function Comment | ×

```
void weather::outputtxt()
{
    CURL* curl = curl_easy_init();
    if(curl){
        FILE* fp = fopen(filename, "wb");
        const char* url = spacedata ; //
        curl_easy_setopt(curl, CURLOPT_URL, url); // URL 설정
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp); // 데이터 쓰기 설정
        CURLcode res = curl_easy_perform(curl); // 데이터 다운로드 수행
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res)); // 오류 처리
        }
        curl_easy_cleanup(curl); // CURL 정리
        fclose(fp); // 파일 닫기
    }
}
```

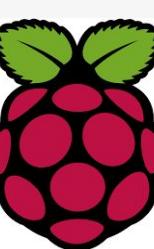
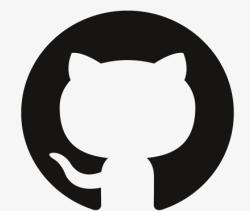
Codeium: Refactor | Explain | Generate Function Comment | ×

```
void weather::init(){
    outputtxt();
    findtxt();
}
```

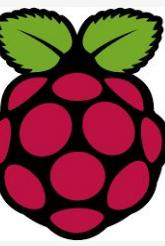
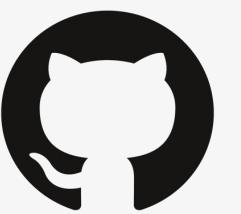
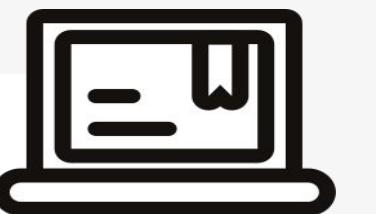
Codeium: Refactor | Explain | Generate Function Comment | ×

```
void weather::findtxt(){
    ifstream inputFile(filename); // 파일 경로를 수정하세요
    string line;

    // 파일을 한 줄씩 읽어서 특정 줄과 특정 단어를 추출
    int currentLine = 0;
    while (std::getline(inputFile, line)) {
        currentLine++;
```



# FINAL.CPP



```
Codeium: Refactor | Explain | Generate Function Comment | ×
void weather::findtxt(){
    ifstream inputFile(filename); // 파일 경로를 수정하세요
    string line;

    // 파일을 한 줄씩 읽어서 특정 줄과 특정 단어를 추출
    int currentLine = 0;
    while (std::getline(inputFile, line)) {
        currentLine++;

        // 5번째 줄인 경우
        if (currentLine == 5) {
            std::istringstream iss(line);
            std::string word;
            int wordCount = 0;

            // 공백을 구분자로 하여 단어 추출
            while (iss >> word) {
                wordCount++;
                //12번째 단어
                if (wordCount == 12) {
                    ta = word;
                    continue;
                }
                // 14번째 단어
                if (wordCount == 14) {
                    hm = word;
                    continue;
                }
                //16번째 단어
                if (wordCount == 16) {
                    rn = word;
                    continue;
                }
                //28번째 단어
                if (wordCount == 26) {
                    ca_tot = word;
                    break;
                }
            }
            break;
        }
    }
    inputFile.close();
}
```

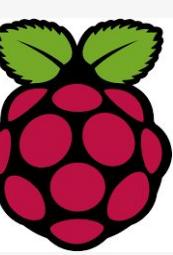
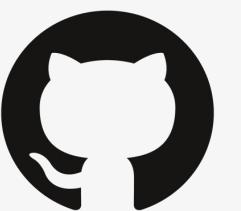
```
Codeium: Refactor | Explain | Generate Function Comment | ×
void weather::OpenCV(string H,string M,string window){

    Mat img = imread(folderPath + imgfile);      //Scalar(220, 245, 245) : 베이지색
    // 날씨 데이터 화면 표기
    putText(img, "Today's Temperature : " + ta, Point(50, 30), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(220, 245, 245), 1.8);
    putText(img, "Today's Rainfall : " + rn, Point(50,50), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(220, 245, 245), 1.8);
    putText(img, "Today's Total Cloudiness : " + ca_tot, Point(50, 70), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(220, 245, 245), 1.8);

    // 시간 데이터 화면 표기
    putText(img, H+ " : "+M, Point(400, 30), FONT_HERSHEY_SIMPLEX, 0.8, Scalar(220, 245, 245), 2.8);

    // print display
    imshow(window, img);
}
```

# FINAL.CPP



```
    inputFile.close();
}

Codeium: Refactor | Explain | Generate Function Comment | ×
void weather::OpenCV(string H,string M){

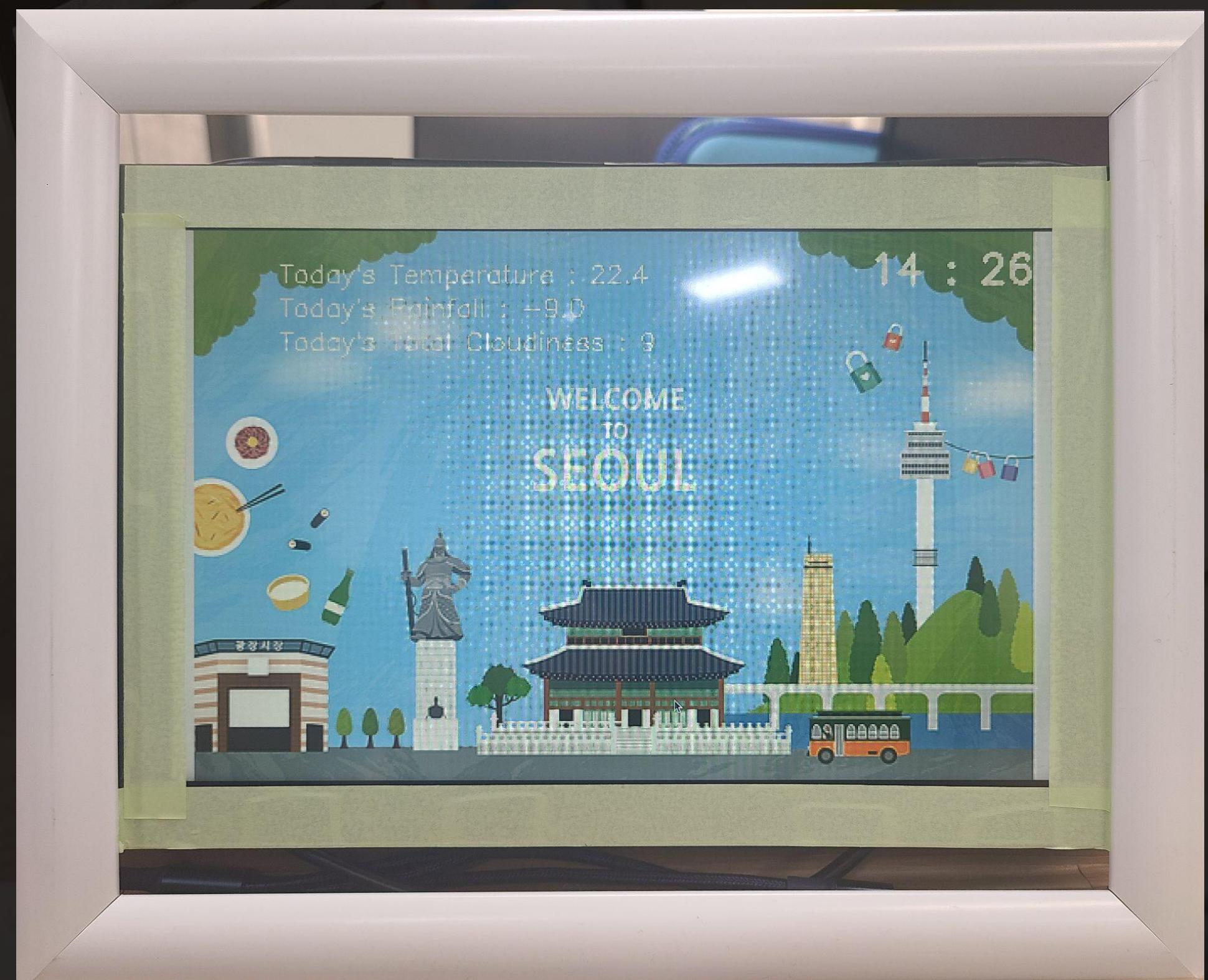
    Mat img = imread(folderPath + imgfile);      //Scalar(220, 245, 245) : 베이지색
    // 날씨 데이터 화면 표기
    putText(img, "Today's Temperature : " + ta, Point(50, 30), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(220, 245, 245), 1.8);
    putText(img, "Today's Rainfall : " + rn, Point(50,50), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(220, 245, 245), 1.8);
    putText(img, "Today's Total Cloudiness : " + ca_tot, Point(50, 70), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(220, 245, 245), 1.8);

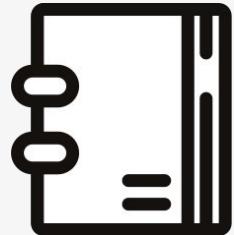
    // 시간 데이터 화면 표기
    putText(img, H+ " : " +M, Point(400, 30), FONT_HERSHEY_SIMPLEX, 0.8, Scalar(220, 245, 245), 2.8);

    // window fullscreen resize
    string windowName = "original image";
    namedWindow(windowName,WINDOW_NORMAL);
    setWindowProperty(windowName,WND_PROP_FULLSCREEN,WINDOW_FULLSCREEN);
    resizeWindow(windowName,1280,800); // raspberry pi 4 display size = (1280,800)

    // print display
    imshow(windowName, img);
}
```

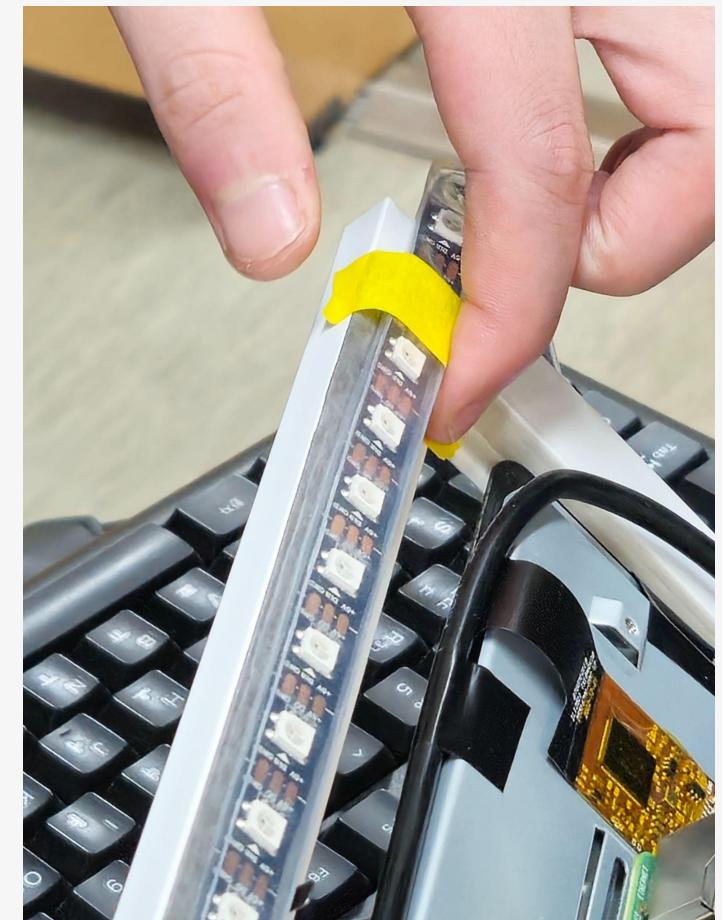
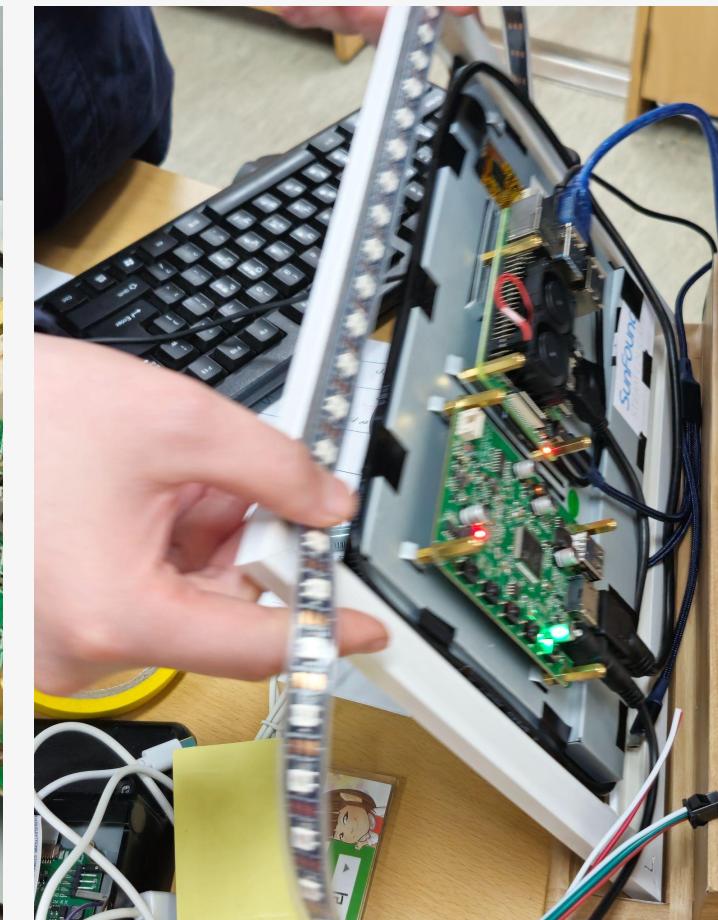
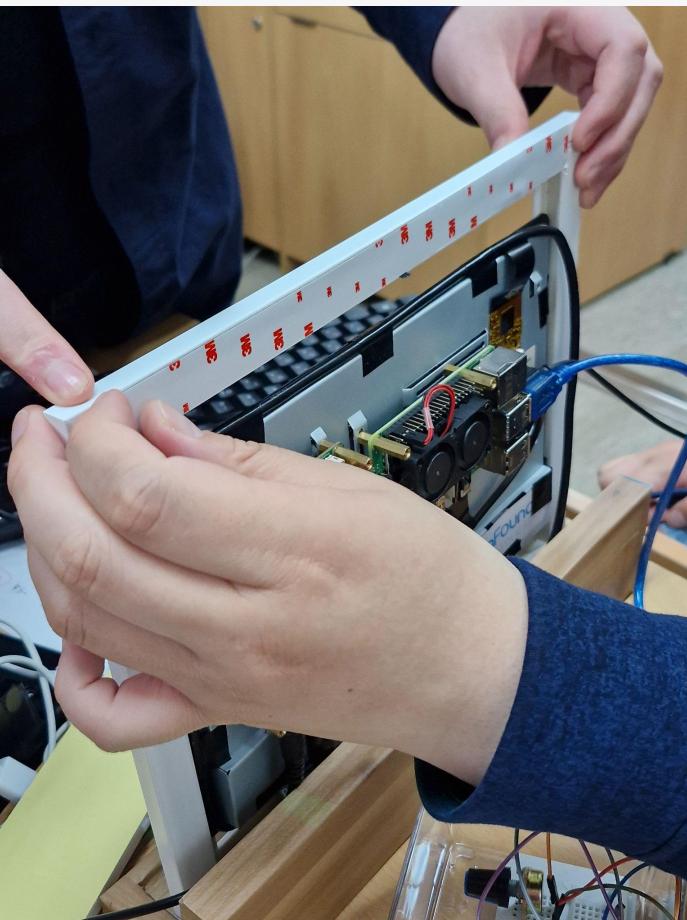
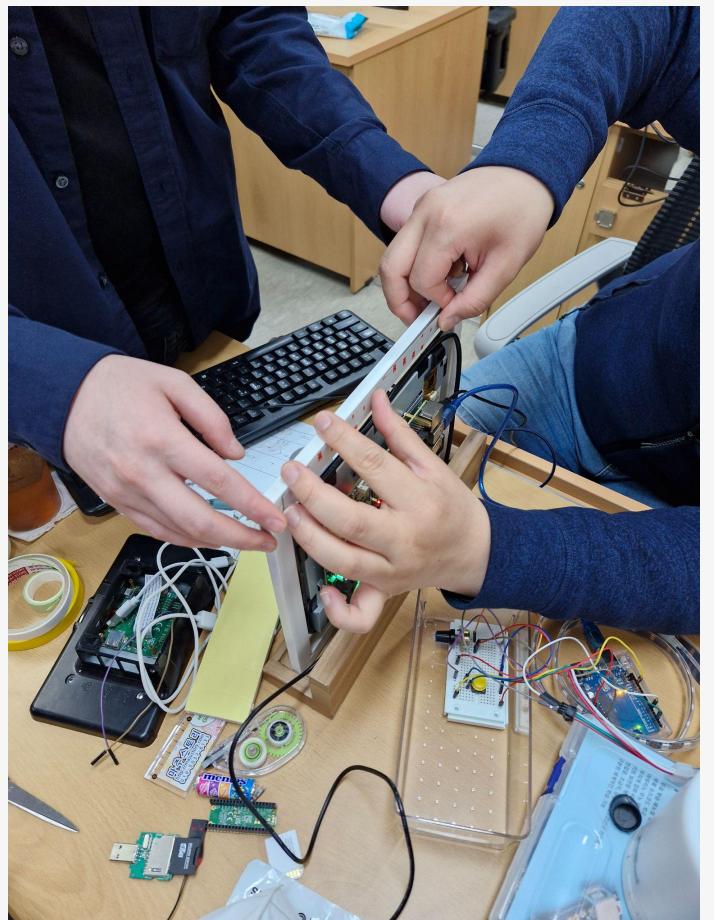
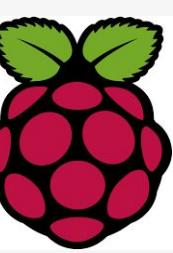
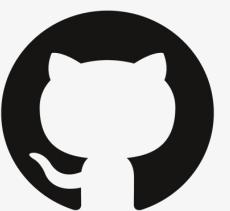
# FINAL DISPLAY

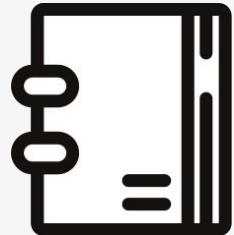




20일 월요일 프로젝트 진행 과정

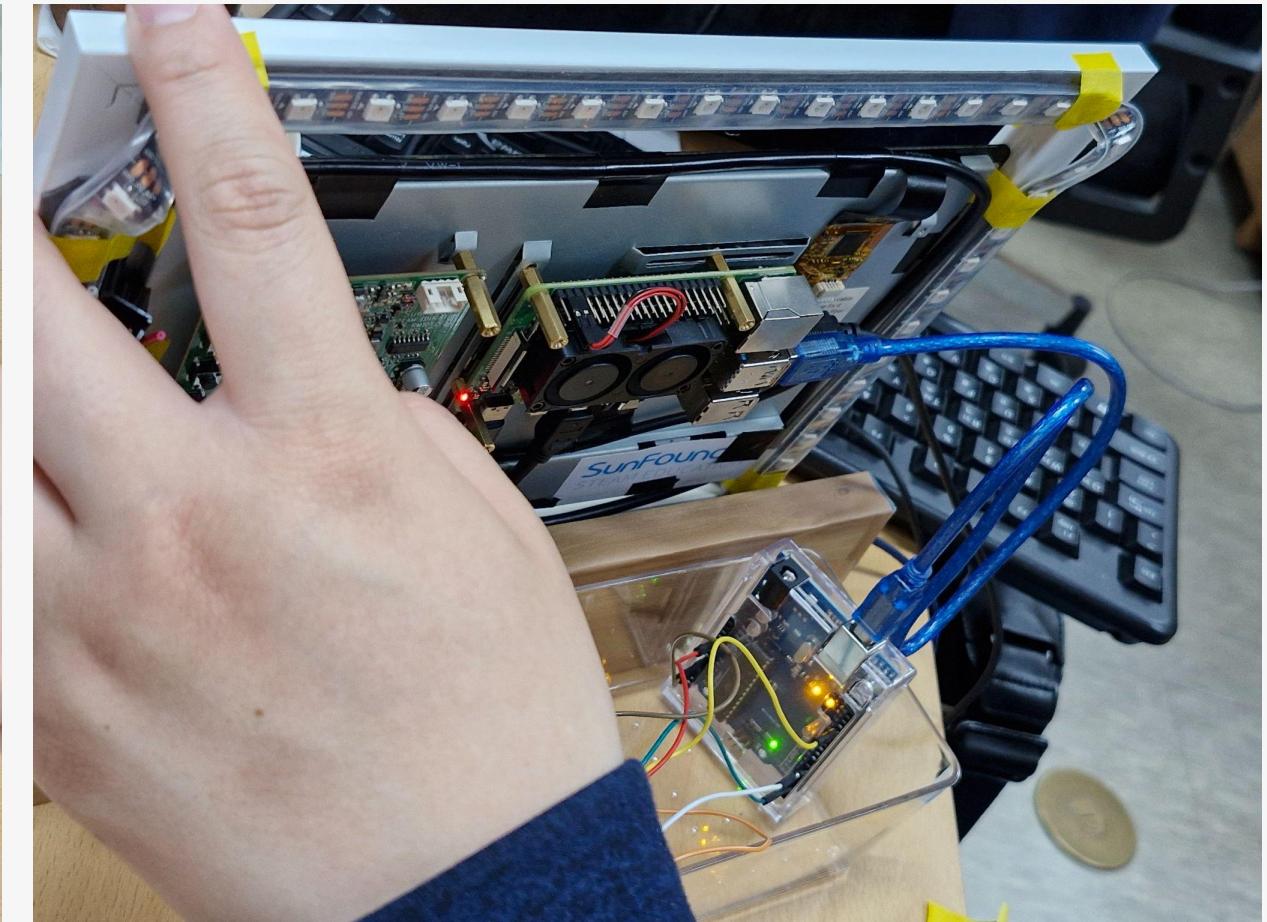
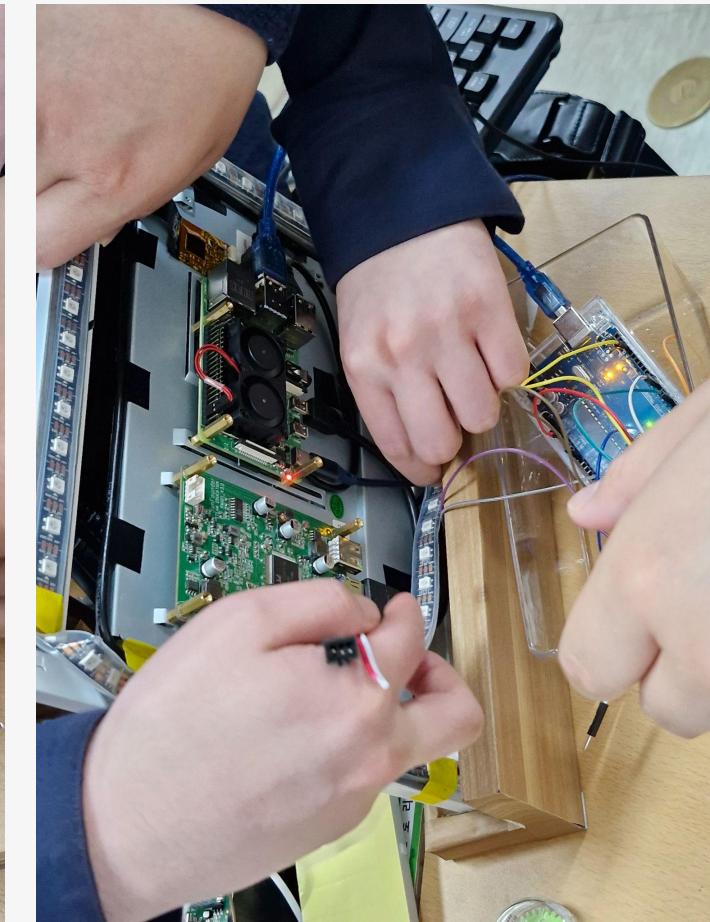
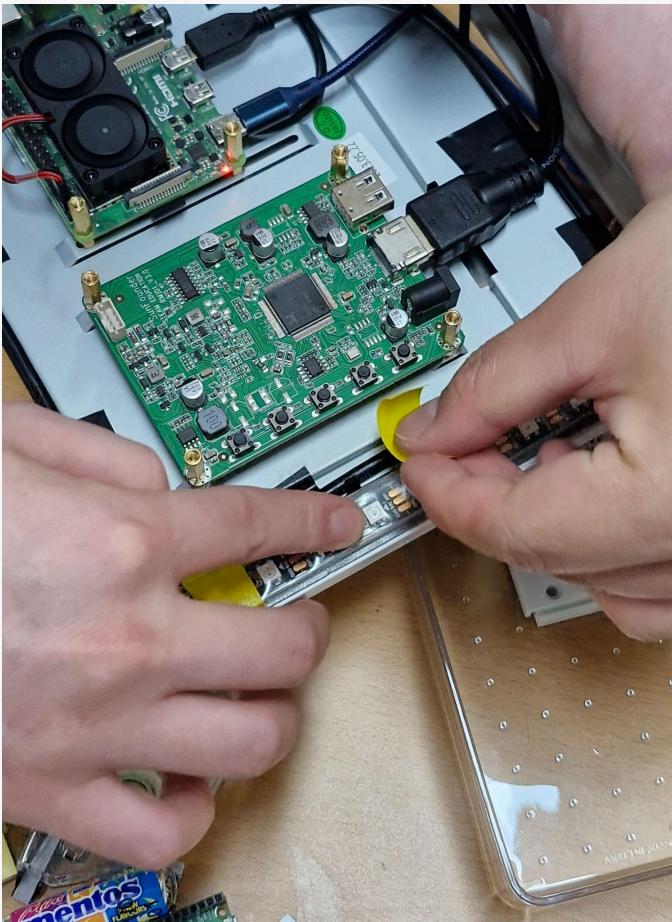
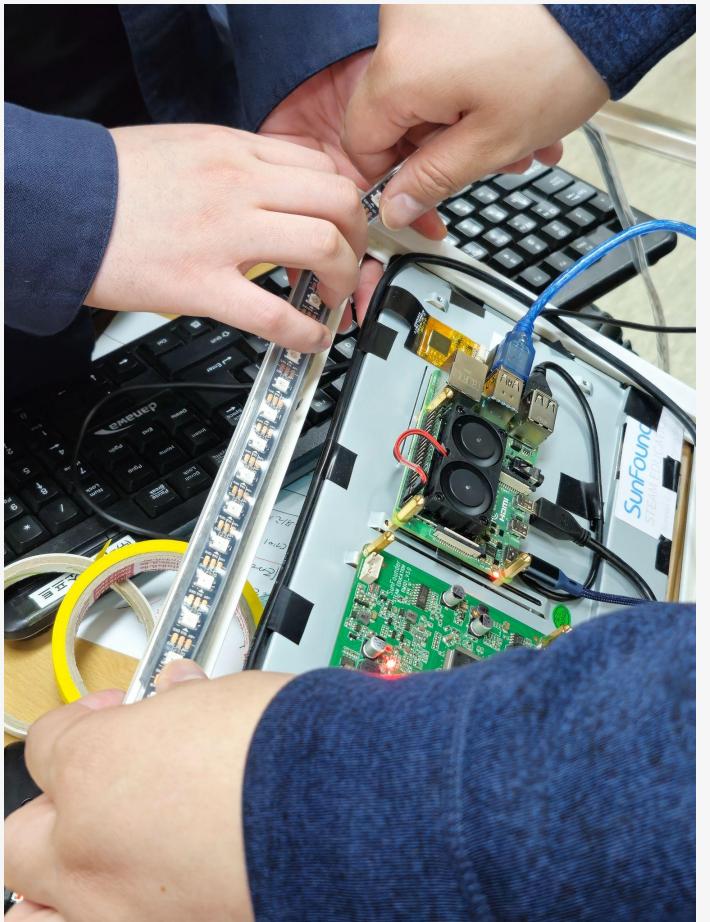
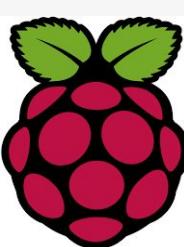
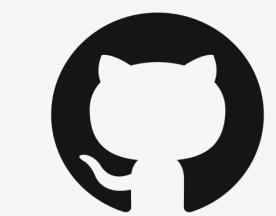
## FINAL DISPLAY

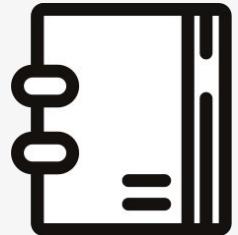




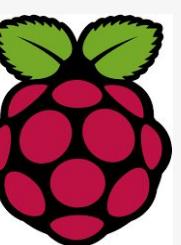
20일 월요일 프로젝트 진행 과정

## FINAL DISPLAY

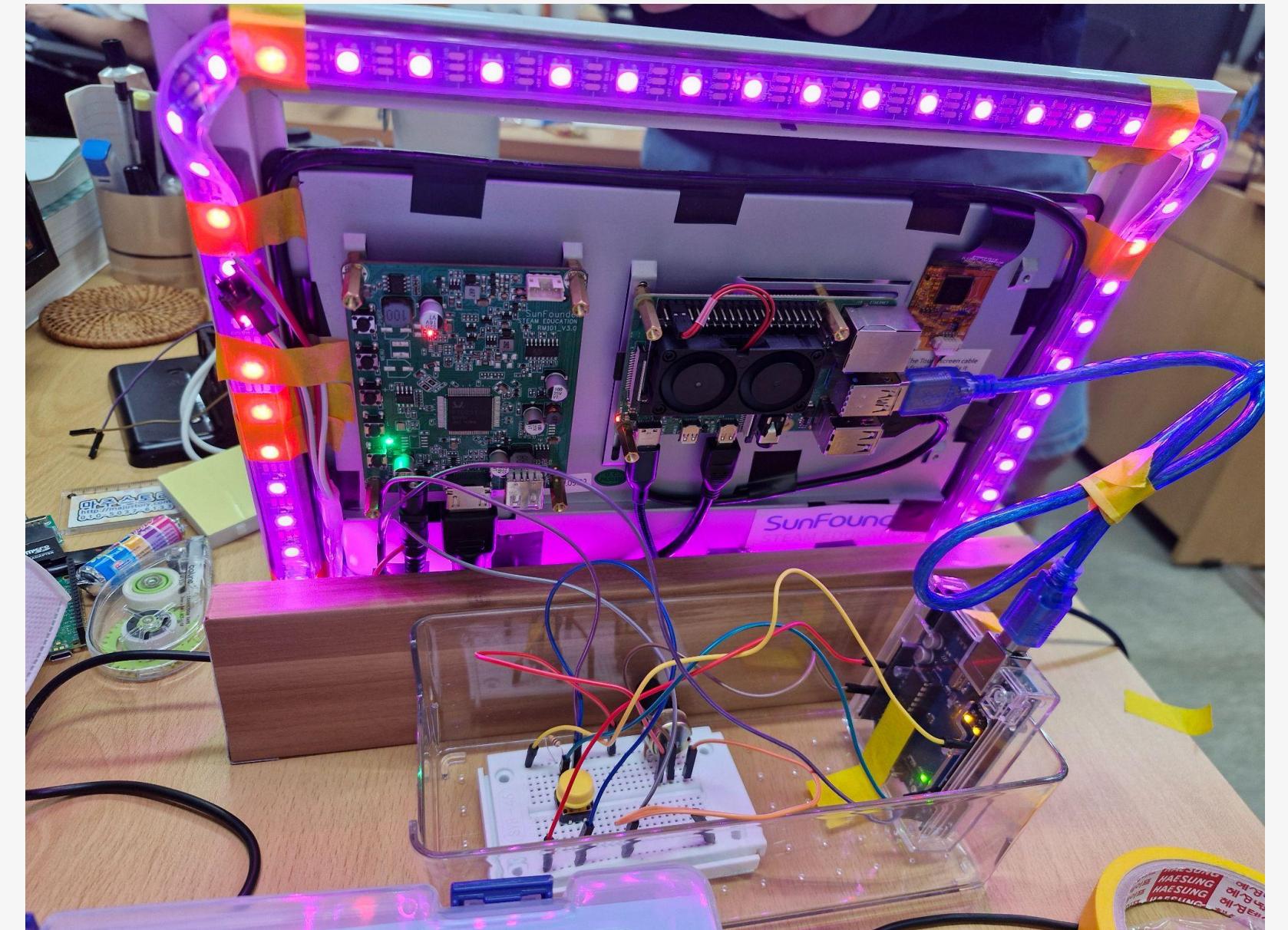




20일 월요일 프로젝트 진행 과정



# FINAL DISPLAY





# 활동방안 추후 목표

활동 방안 그리고 추후 목표?

# 스마트 미러 구현





# 코드 리뷰 및 해결방안



출처

# 기상청데이터 인용

The left screenshot shows the '气象API' (气象API) page on the Korean Meteorological Administration's website. It displays various API endpoints such as 'AWS 기상문서', '계절관측', '자료관측', and '지역관측'. The right screenshot shows a blog post titled '기상청 API허브로 기상기후데이터 알차게 활용하기' (Using the Korea Meteorological Administration API Hub to Intelligently Utilize Weather and Climate Data). The post includes a diagram illustrating the use of the API for weather data, and a sidebar with news articles related to the API.

### WS2812B

Intelligent control LED integrated light source

**Features and Benefits**

- Intelligent reverse connect protection, the power supply reverse connection does not damage the IC.
- The control circuit and the LED share the only power source.
- Control circuit and RGB chip are integrated in a package of 5050 components, form a complete control of pixel point.
- Built-in signal reshaping circuit, after wave reshaping to the next driver, ensure wave-form distortion not accumulate.
- Built-in electric reset circuit and power lost reset circuit.
- Each pixel of the three primary color can achieve 256 brightness display, completed 16777216 color full color display, and scan frequency less than 400Hz.
- Cascading port transmission signal by single line.
- Any two point the distance more than 5m transmission signal without any increase circuit.
- When the refresh rate is 30fps, cascade number are not less than 1024 points.
- Send data at speeds of 800Kbps.
- The color of the light were highly consistent, cost-effective..

**Applications**

- Full-color module, Full color soft lights a lamp strip.
- LED decorative lighting, Indoor/outdoor LED video irregular screen.

**General description**

WS2812B is an intelligent control LED light source that the control circuit and RGB chip are integrated in a package of 5050 components. It internal include intelligent digital port data latch and signal reshaping amplification drive circuit. Also include a precision internal oscillator and a 12V voltage programmable constant current control part, effectively ensuring the pixel point light color height consistent.

The data transfer protocol use single NRZ communication mode. After the pixel power-on reset, the DIN port receive data from controller, the first pixel collect initial 24bit data then sent to the internal data latch, the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade pixel through the DO port. After transmission for each pixel, the signal to reduce 24bit pixel adopt auto reshaping transmit technology, making the pixel cascade number is not limited the signal transmission, only depend on the speed of signal transmission.

LED with low driving voltage, environmental protection and energy saving, high brightness, scattering angle is large, good consistency, low power, long life and other advantages. The control chip integrated in LED above becoming more simple circuit, small volume, convenient installation.

**Absolute Maximum Ratings**

Parameter	Symbol	Ratings	Unit
Power supply voltage	$V_{DD}$	$+3.5\sim+5.3$	V
Input voltage	$V_I$	$-0.5\sim V_{DD}+0.5$	V
Operation junction temperature	$T_{Op}$	$-25\sim+80$	°C
Storage temperature range	$T_{Stg}$	$-40\sim+105$	°C

**Electrical Characteristics** ( $T_A=-20\sim+70^\circ\text{C}$ ,  $V_{DD}=4.5\sim5.5\text{V}$ ,  $V_{SS}=0\text{V}$ , unless otherwise specified)

**Data transfer time** ( $TH-TL=1.25\mu\text{s}\pm600\text{ns}$ )

Symbol	Condition	Min	T <sub>Typ</sub>	Max	Unit
$t_{TH}$	CL=15pFDIN→DOUT, RL=10KΩ	—	—	300	ns
$t_{TL}$	CL=300pF, OUTR/OU TG/OUTB	—	—	120	μs
$t_{RES}$	low voltage time	Above 50μs	—	—	—

**Sequence chart:**

# LED STRIP DATA SHEET

# 기상청 + 지역코드 출처

Code Blame 28 lines (22 loc) · 972 Bytes ⚡ Code 55% faster with GitHub Copilot

```
1 대한민국 기상청 대표 블로그: 생기발랄 : https://m.blog.naver.com/kma\_131/223006786447
2 공공데이터포털 : https://www.data.go.kr/index.do
3 기상청 API허브 : https://apihub.kma.go.kr/
4 현재시간추출(chat gpt) : https://chatgpt.com/c/f2fc2710-e3e4-4c31-93ad-5ac22ee91705
5 텍스트 단어추출(chat gpt) : https://chatgpt.com/c/5b10d7ad-d4aa-4a6b-9436-84256dd64a64
6
7 예보구역코드 정리 : (BLOG)소엔코딩 : https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=soencoding&logNo=221329313666
8
9 11B00000 서울, 인천, 경기도
10 11D10000 강원도영서
11 11D20000 강원도영동
12 11C20000 대전, 세종, 충청남도
13 11C10000 충청북도
14 11F20000 광주, 전라남도
15 11F10000 전라북도
16 11H10000 대구, 경상북도
17 11H20000 부산, 울산, 경상남도
18 11G0000 제주도
19
20 예보관서번호 정리 : https://data.kma.go.kr/tmeta/stn/selectStnList.do?pgmNo=123
21 서울: 108
22 세종: 29
23 제주: 184
24 대전: 133
25 부산: 159
```

# 지출내역

## □ 예산

연번	내용	금액(원)	수량	총액(원)	비고
1	받침대 상자	-	1개		3D프린터
2	라즈베리파이	-	1개		HRD 지원
3	디스플레이	-	1개		강사 지원
4	LED Strip	-	1개		HRD 지원
총액				-	

## □ 의의

- 제한된 예산 안에서 표현하고자 하여 기존 방향보다 축소되었음
- 차후 디스플레이 기술을 활용하여 다양한 산업으로 나아갈 수 있음을 확인함
- 이전 내용 복기하여 부족한 점을 알고 이를 차후에 공부할 수 있는 계기가 되었음.



감사합니다!