

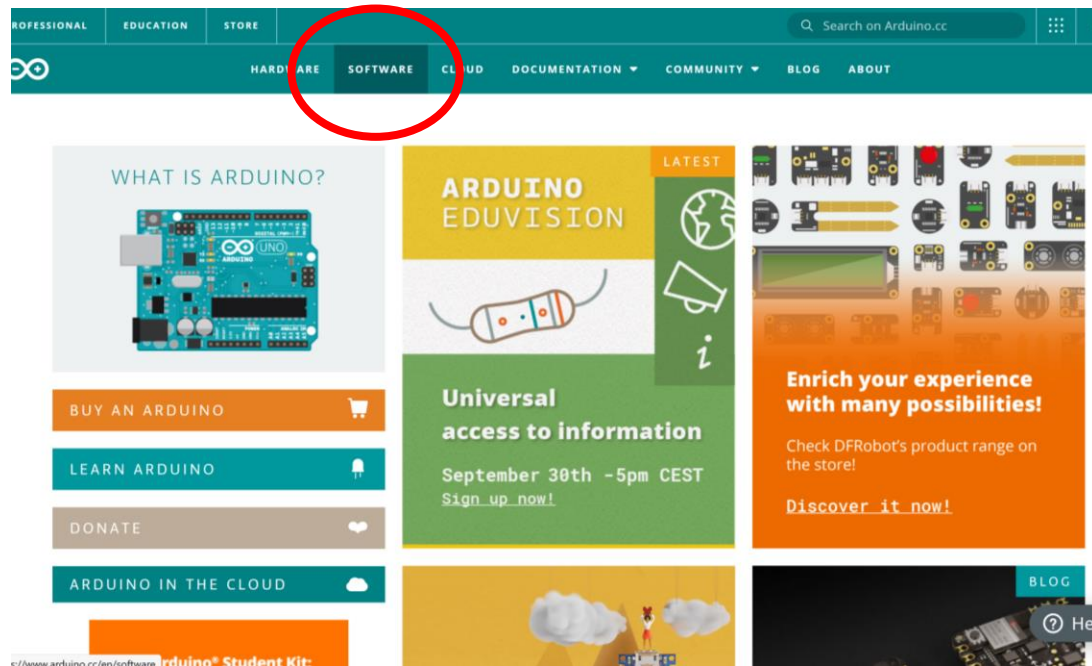
아두이노 R1 D1보드 연결 튜토리얼

아두이노 개발 환경 만들기

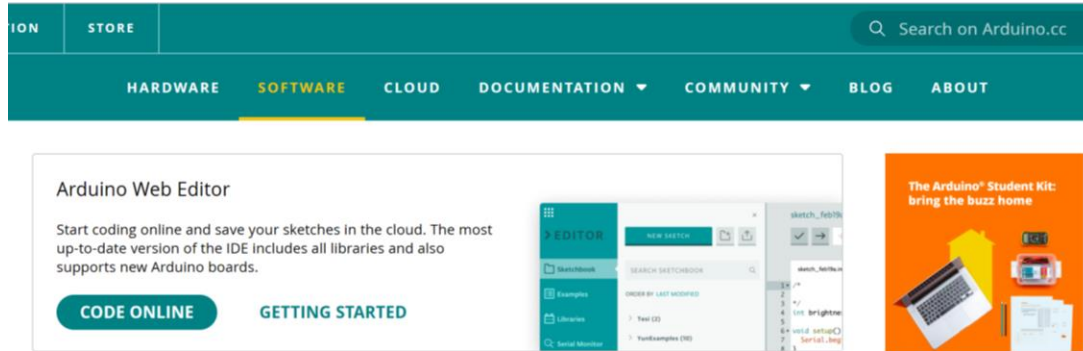
강사: 차형준

1. 아두이노 통합 개발환경 다운받기

- <https://www.Arduino.cc>에 접속합니다.
- SOFTWARE 메뉴를 클릭합니다.

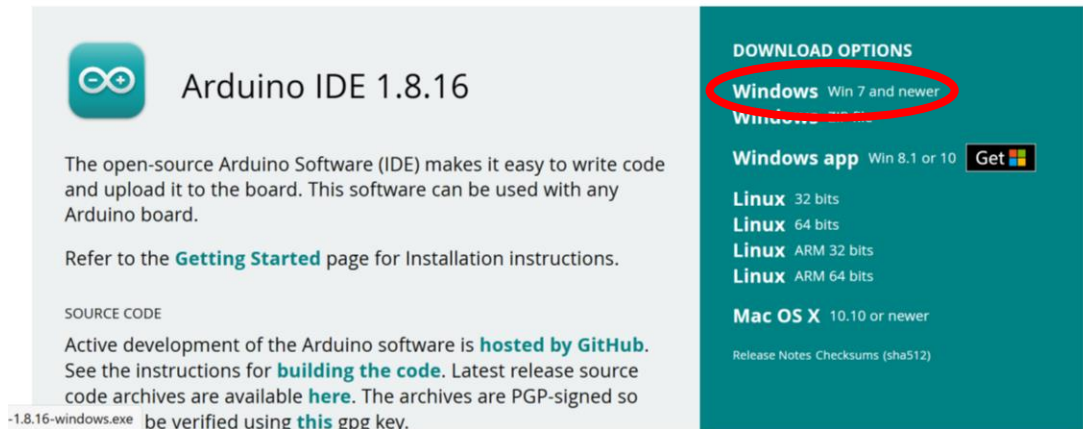


1. 아두이노 통합 개발환경 다운받기

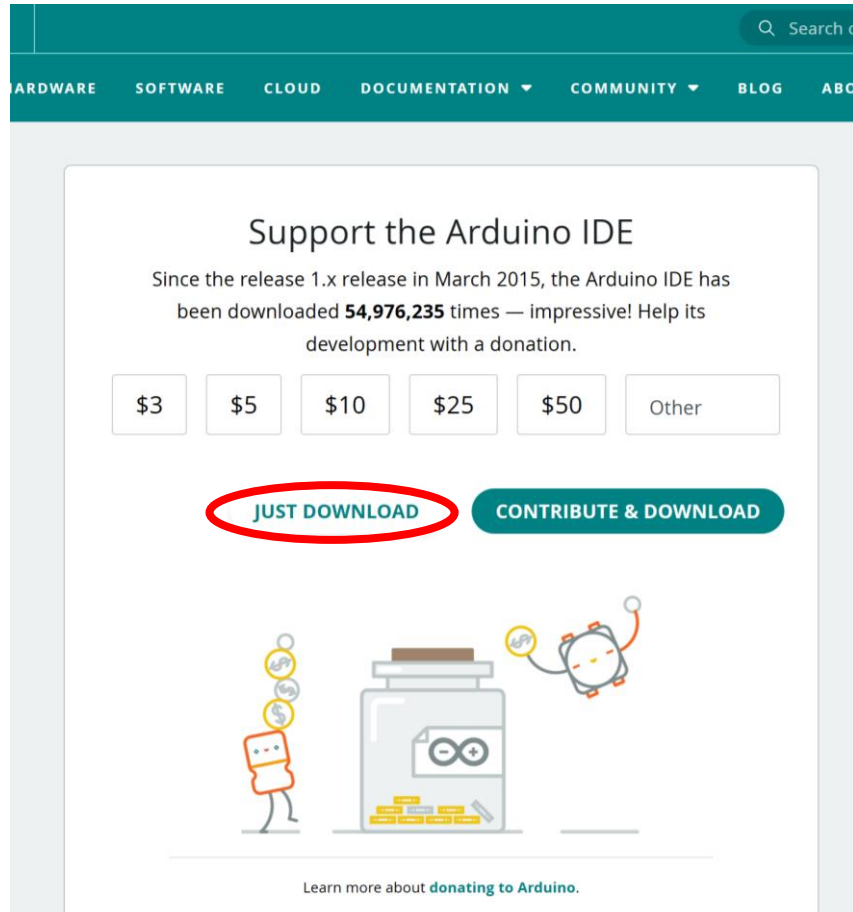


- Windows Win 7 and newer
를 클릭해주세요.

Downloads



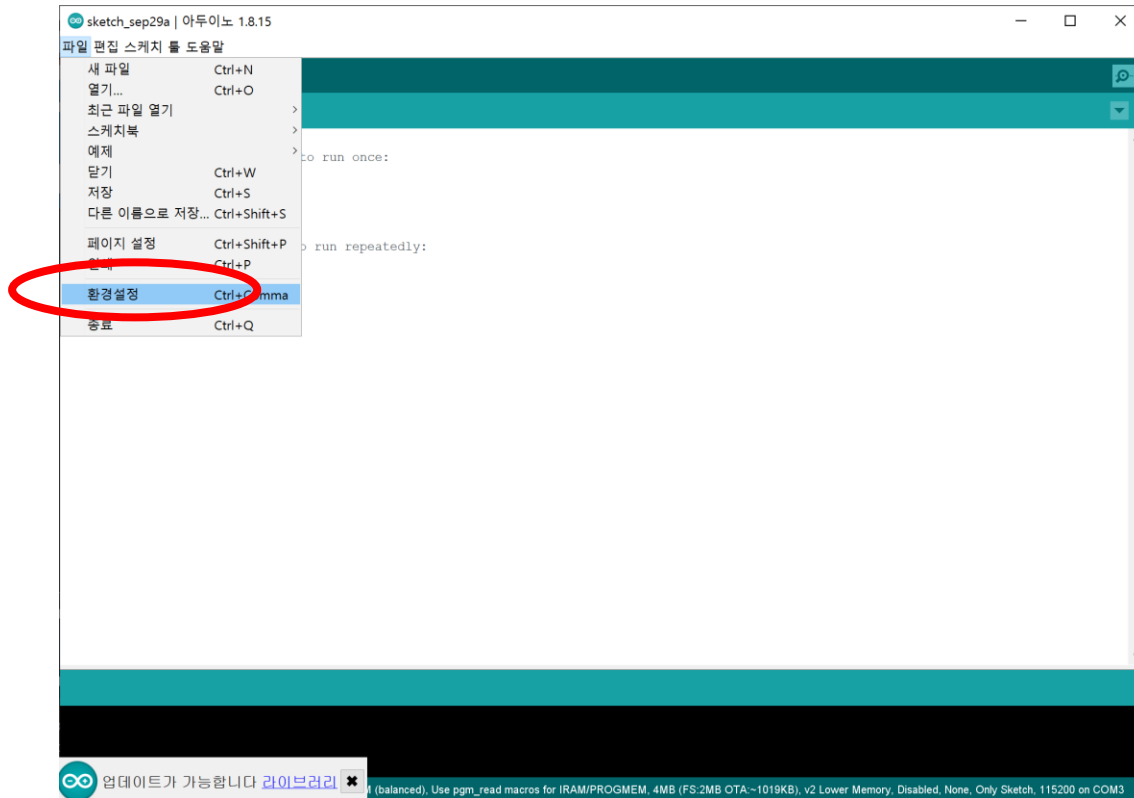
2. 아두이노 통합 개발환경 설치하기



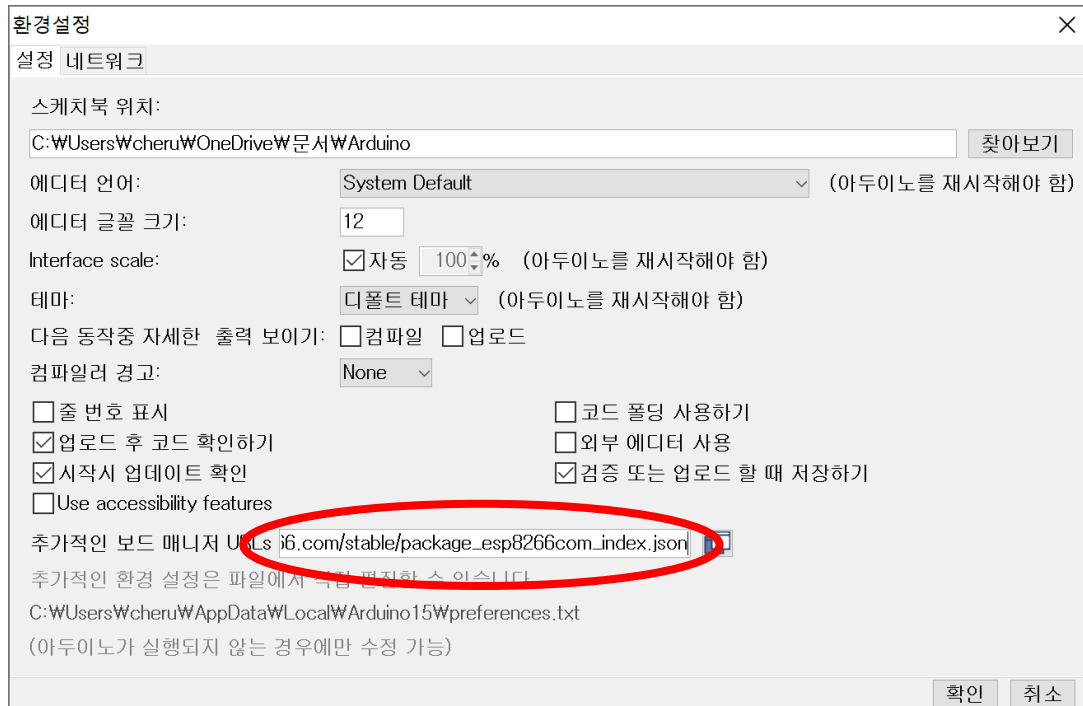
- JUST DOWNLOAD를 눌러 받은 뒤 실행시켜 설치합니다.

3. 아두이노 WiFi보드 드라이버 설치

1. 설치된 아두이노를 열어줍니다.
2. 메뉴 > 환경설정을 클릭합니다.



3. 아두이노 WiFi보드 드라이버 설치



1. 환경 설정 창이 뜨면 **추가적인 보드 매니저 URLs**에

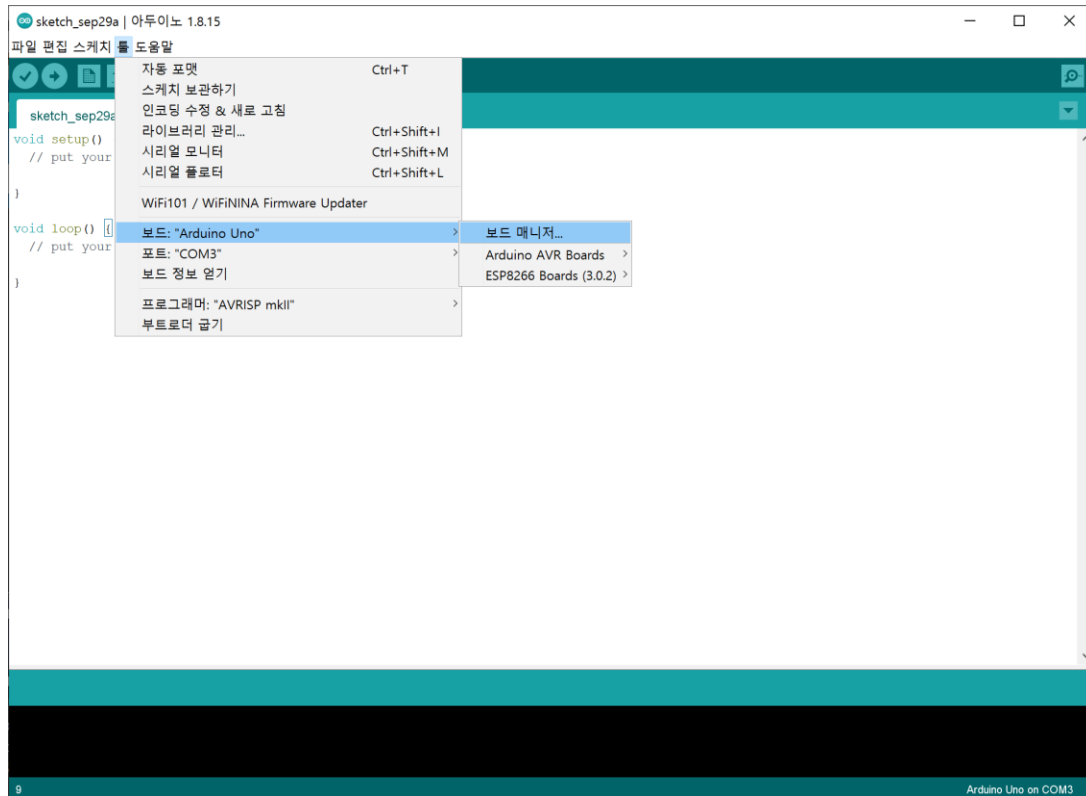
2. 아래의 값을 그대로 **복사해 붙여 넣어주세요.**

https://Arduino.esp8266.com/stable/package_esp8266com_index.json

3. 확인을 눌러줍니다.

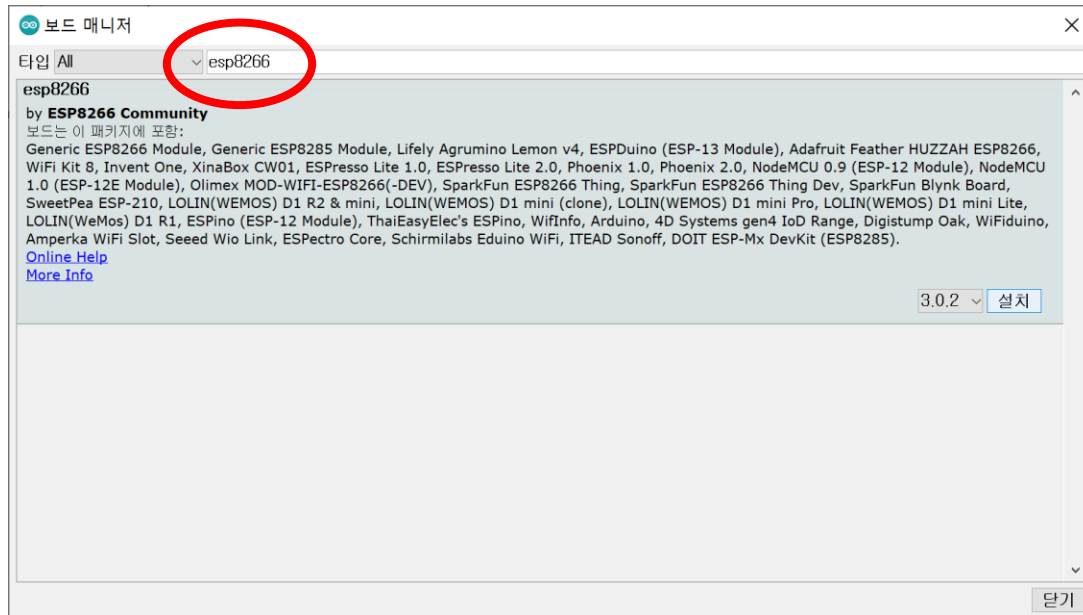
3. 아두이노 WiFi보드 드라이버 설치

- 툴 > 보드 > 보드 매니저를 클릭하세요.

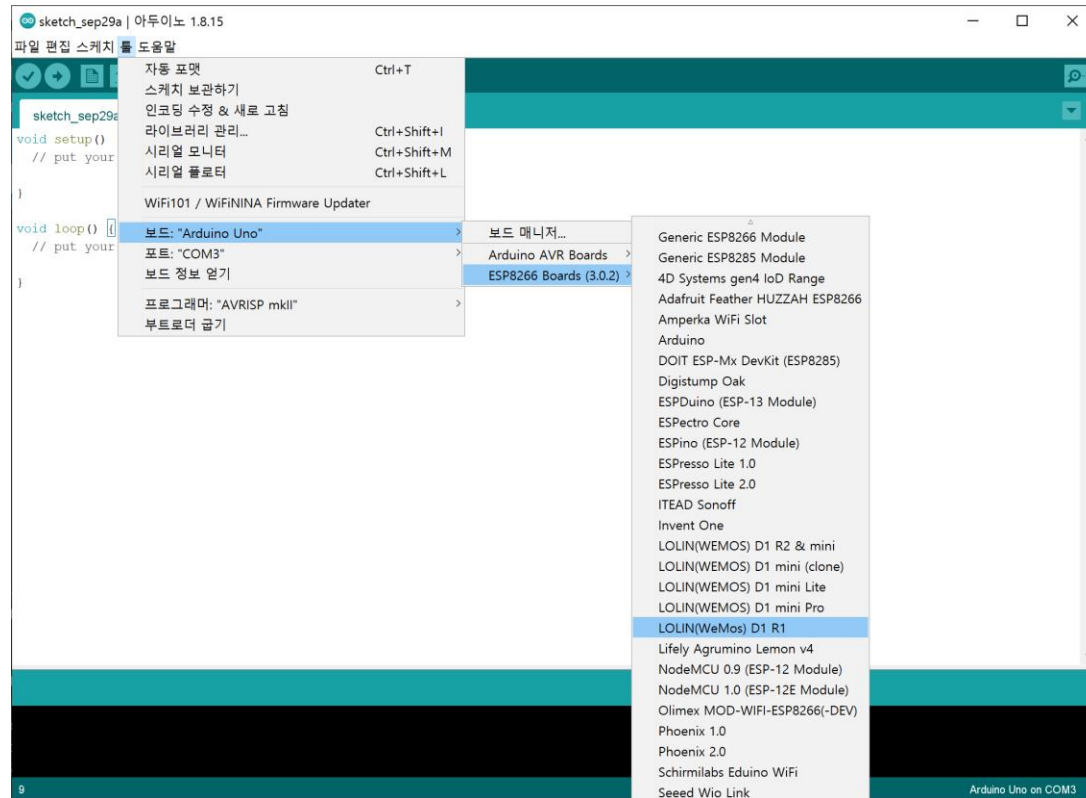


3. 아두이노 WiFi보드 드라이버 설치

1. 왼쪽과 같은 창이 뜨면
2. esp8266을 검색하여
3. ESP8266 Community를 설치합니다.



4. 아두이노 WiFi보드 설정



- 설치가 완료되면

- 툴

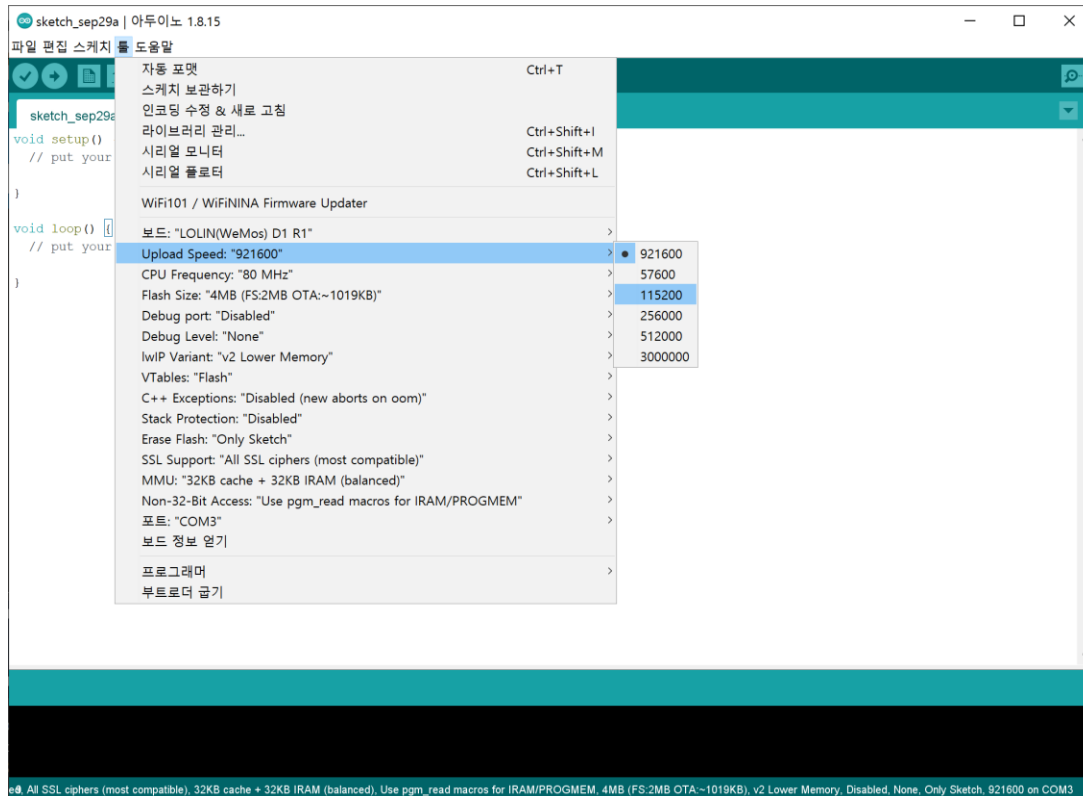
- 보드

- ESP8266 Board(3.0.2)

- LOUN(WeMos) D1 R1

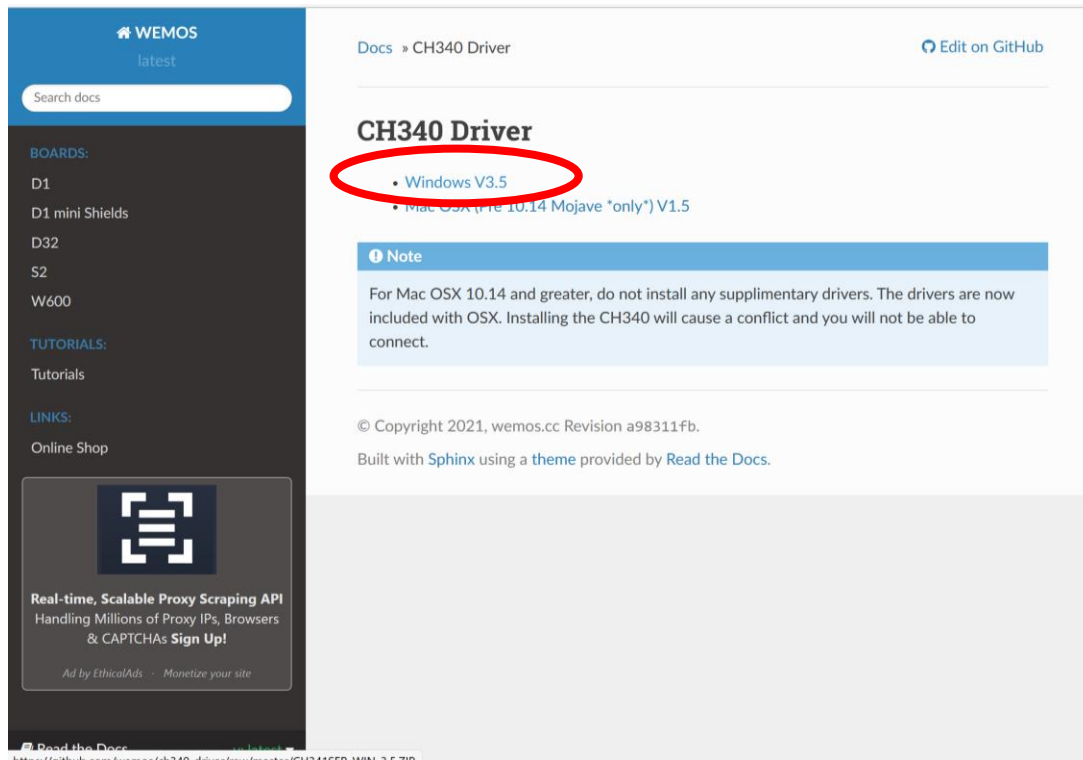
- 을 클릭합니다.

5. 아두이노 WiFi보드 상세 설정하기



1. 툴 메뉴에서
2. Upload Speed를 115200으로 바꿔주세요.

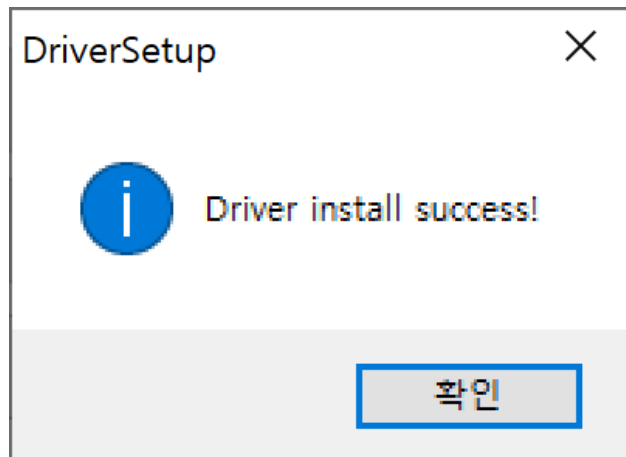
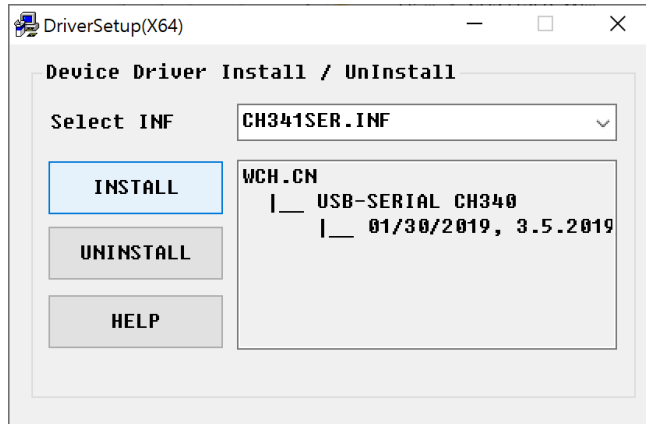
6. 호환보드 드라이버 다운로드



1. [CH340 Driver](#) 링크로 들어간 뒤 Windows V3.5를 클릭해 파일을 다운 받아주세요.

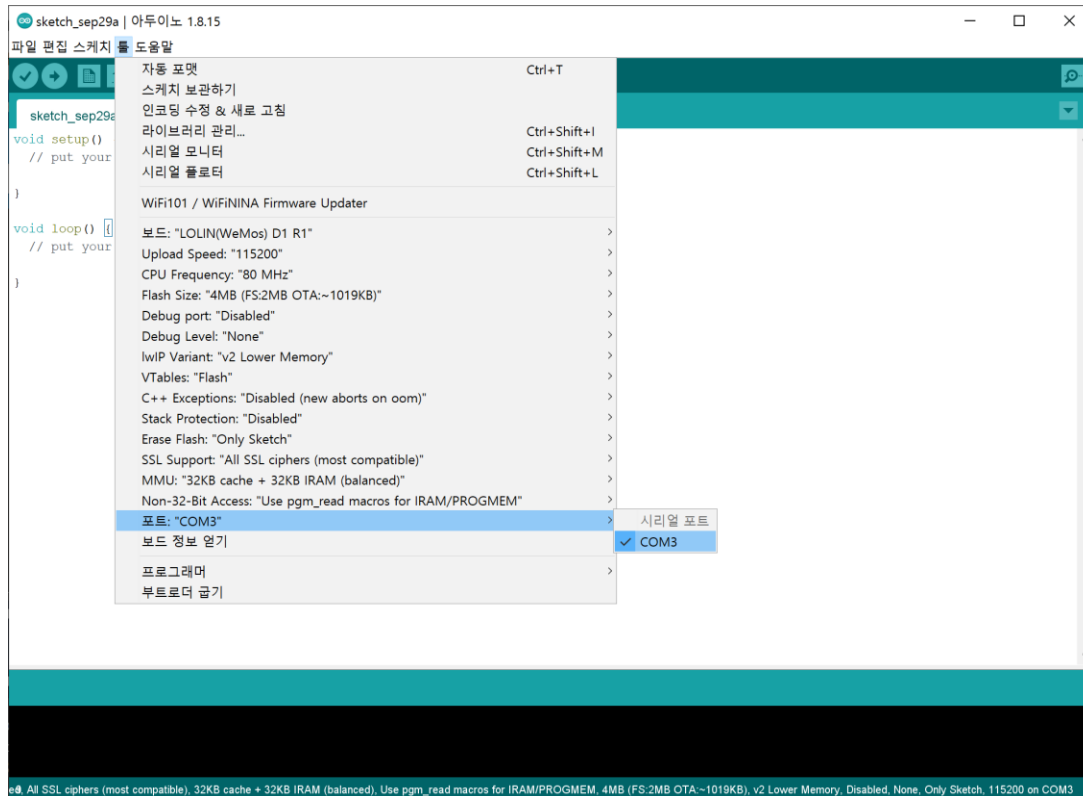
2. 다운 받은 압축파일의 압축을 모두 풀어 준 뒤 Setup.exe를 실행 시켜줍니다.

7. 호환보드 드라이버 설치



1. 왼쪽 위의 사진과 같은 창이 뜨면
2. 아두이노 기기를 USB로 연결해주세요.
3. 연결 후에 INSTALL 버튼을 눌러서 드라이브를 설치합니다.
4. 왼쪽 아래와 같이 성공 메시지가 뜨면 제대로 되었습니다.

8. 아두이노 WiFi보드 기기 연결 확인하기



1. 아두이노 프로그램에서
2. 툴 → 포트에서
3. 적절한 것으로 선택해주세요.

- COM(숫자) 형태입니다.

- 여러 개가 뜬다면 제일 아래 것으로 해보고
업로드시 에러가 난다면 다른 포트로 바꿔보세요.

모든 준비가 완료되었습니다.

아두이노 온도 센서 튜토리얼

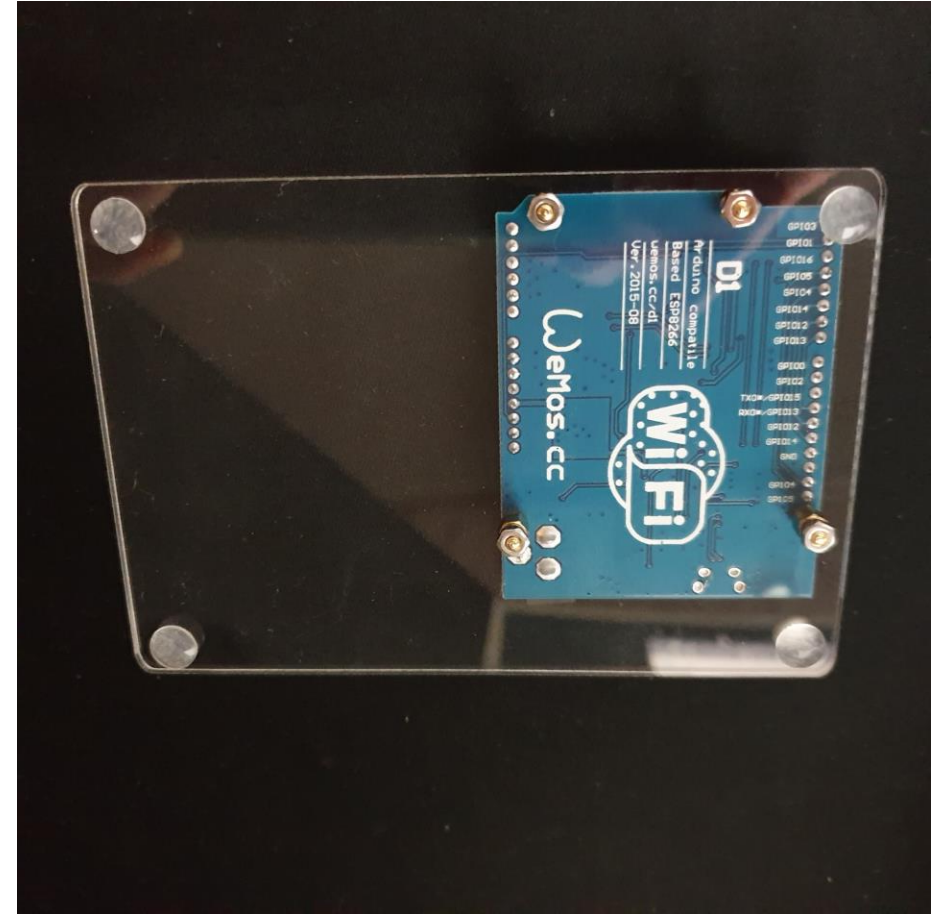
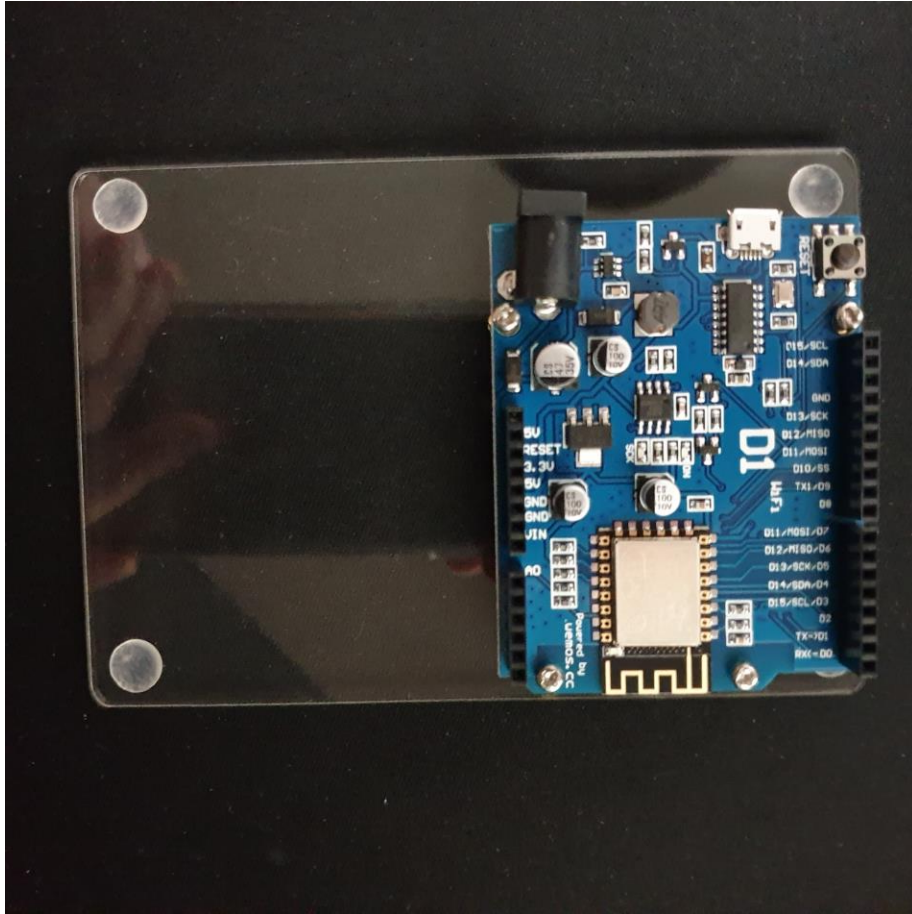
온도 센서로 실시간 측정값 받기

1. 아두이노 아크릴판에 조립하기

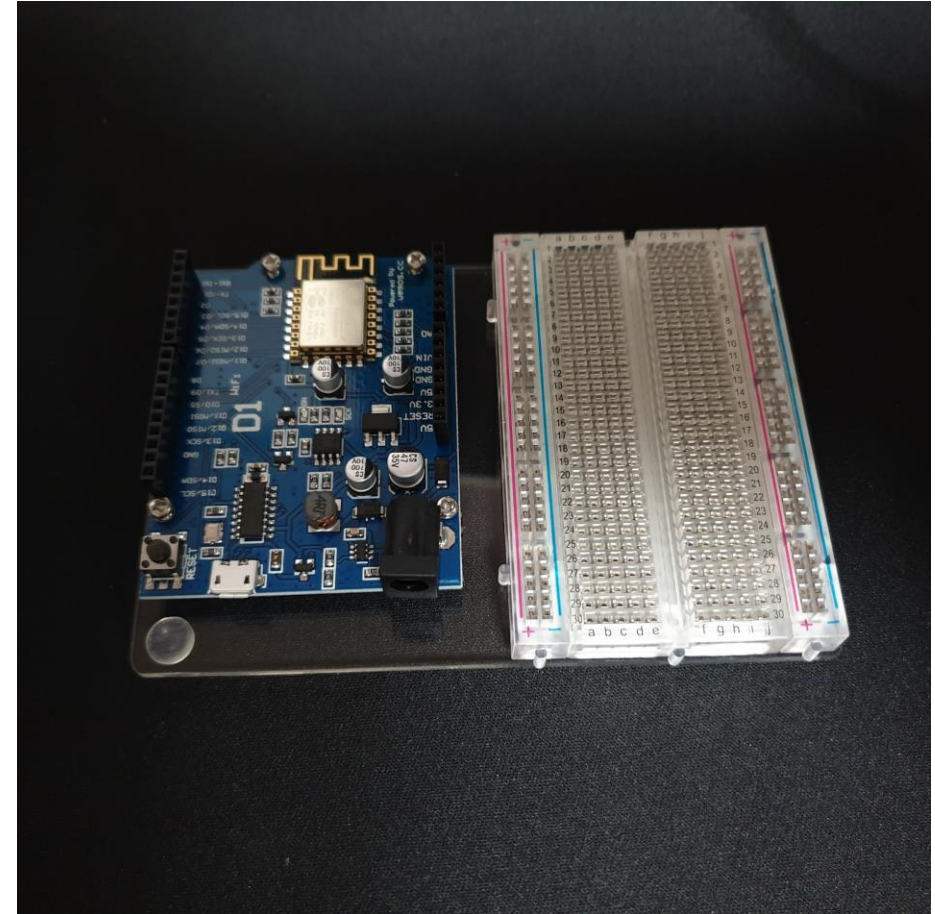
- 이 과정은 생략해도 좋습니다.

1. 이어지는 다음의 사진처럼 완성해 주세요.
2. 나사는 손으로만 조여줘도 충분합니다.
3. 아크릴판 우측의 빈공간에 브레드보드를 부착해줍니다.
4. 아크릴판 뒷면에 미끄럼 방지 엠보싱을 부착해주세요.

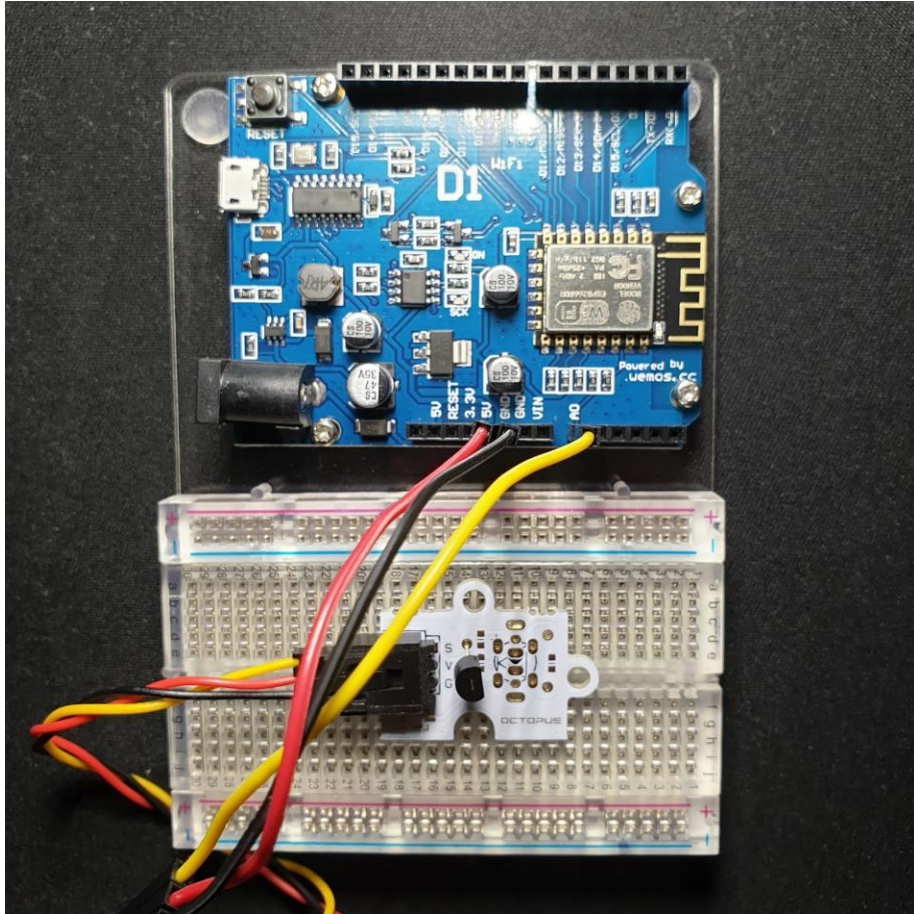
아두이노 아크릴판 조립 사진 1



아두이노 아크릴판 조립 사진 2



2. 회로 연결하기 - 온도 센서

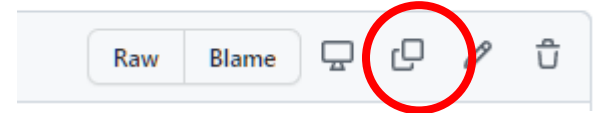


1. 온도센서 안에 함께 있는 케이블을 연결합니다.
2. 반대쪽 케이블을 수-수 점프선으로 사진과 같이 연결합니다.

G-까만색 선-GND
V-빨간색 선-5V
S-노란색 선-A0

- Tip : 되도록 같은 색깔의 선을 사용하는 것이 좋습니다.

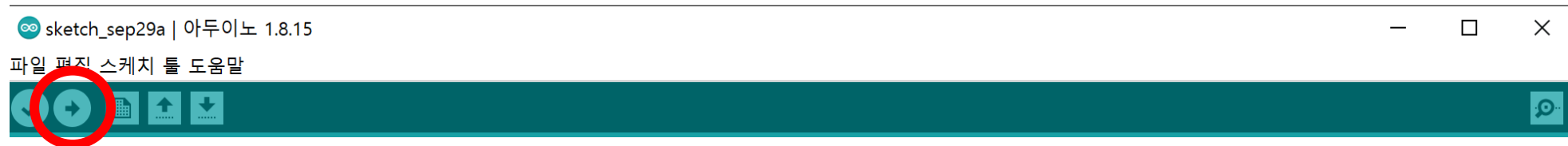
3. 코드를 작성하여 업로드하기



1. 아래의 링크에 들어가서 모든 코드를 복사하여 그대로 붙여 넣어주세요. (그림과 같은 버튼을 누르면 한 번에 복사됩니다.)

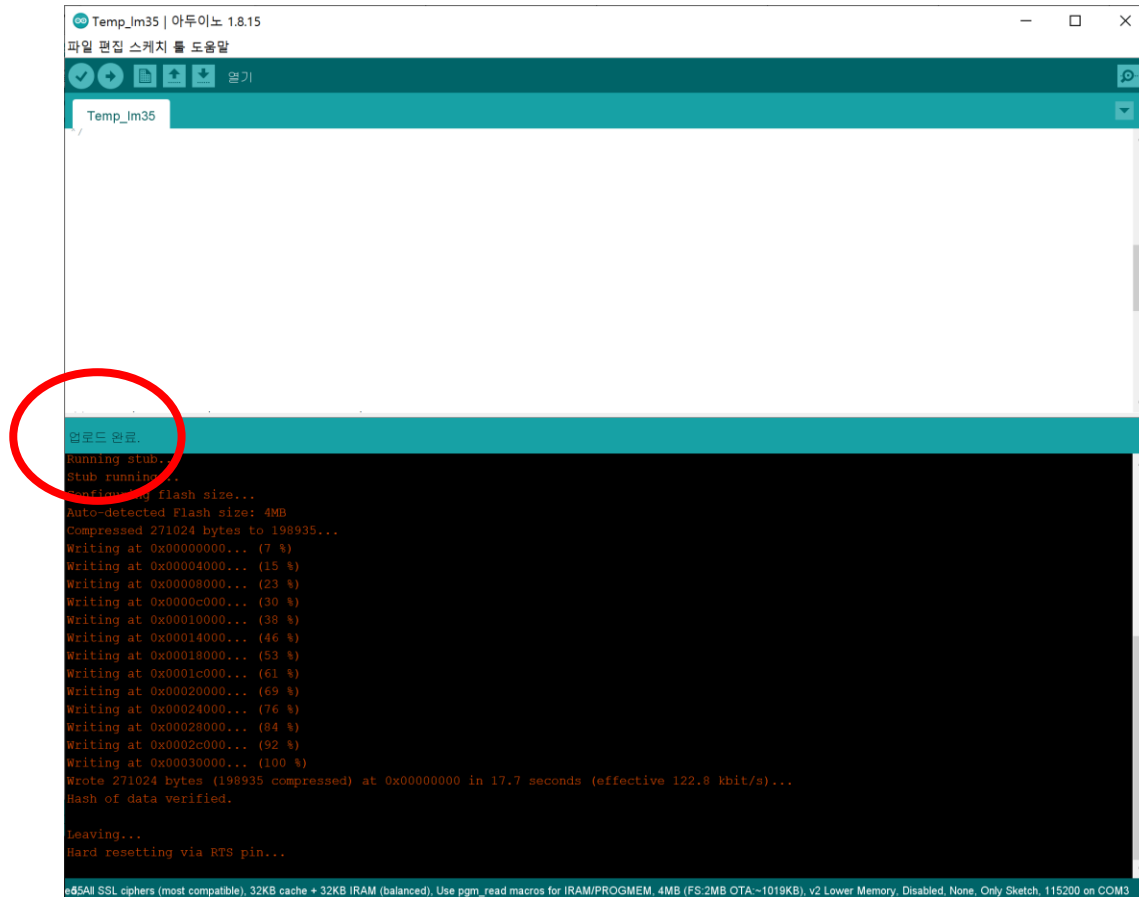
<https://github.com/cherub8128/Lecture/blob/master/temperature.ino>

2. 아두이노의 업로드 버튼을 누릅니다. (빨간색 동그라미)



- 저장메시지가 뜰 시 아무 곳이나 저장하면 됩니다.

4. 업로드 확인하기



The screenshot shows the Arduino IDE interface. At the top, the title bar reads "Temp_lm35 | 아두이노 1.8.15". Below it, the menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar has icons for opening, saving, and running. The main workspace shows a file named "Temp_lm35". A red circle highlights the "업로드 완료" (Upload Complete) message in the status bar. The terminal window at the bottom displays the following output:

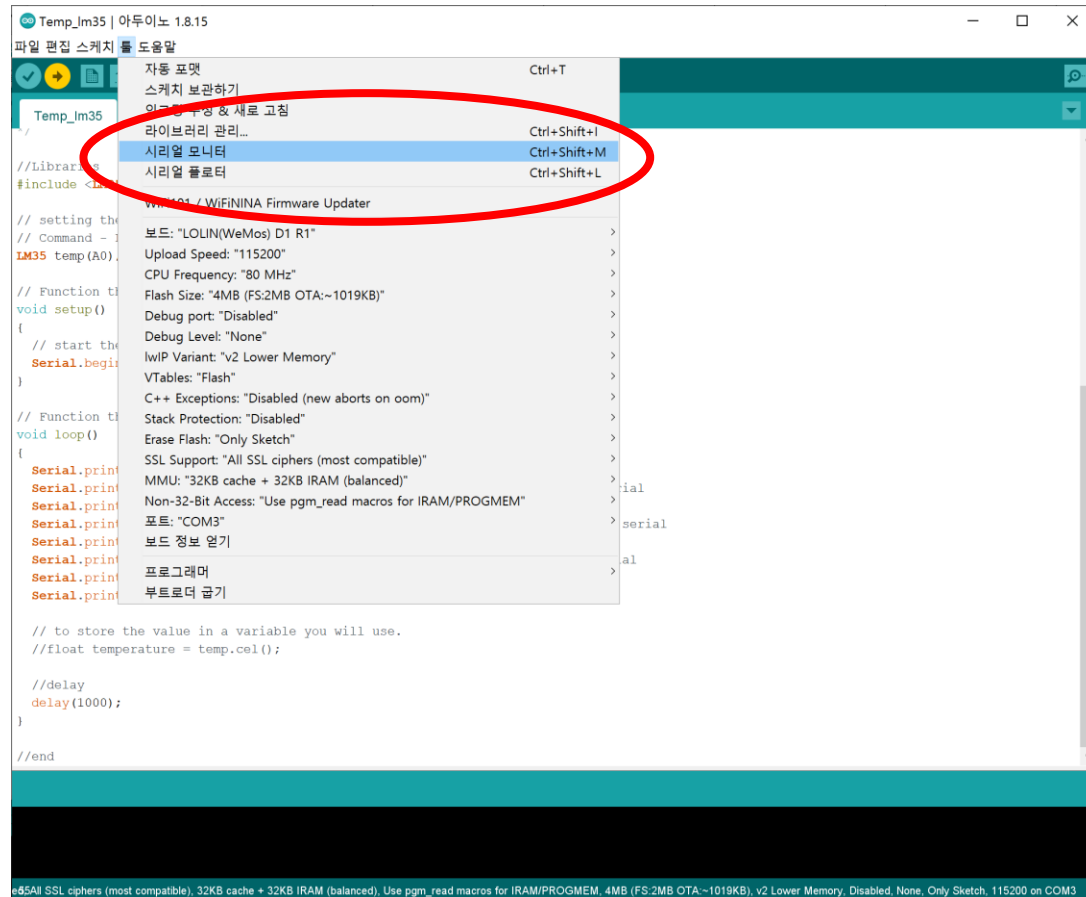
```
Running stub...
Stub running...
Detecting flash size...
Auto-detected Flash size: 4MB
Compressed 271024 bytes to 198935...
Writing at 0x00000000... (7 %)
Writing at 0x00004000... (15 %)
Writing at 0x00008000... (23 %)
Writing at 0x0000c000... (30 %)
Writing at 0x00010000... (38 %)
Writing at 0x00014000... (46 %)
Writing at 0x00018000... (53 %)
Writing at 0x0001c000... (61 %)
Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 271024 bytes (198935 compressed) at 0x00000000 in 17.7 seconds (effective 122.8 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

At the bottom of the terminal, the following information is displayed: "e85AII SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA~1019KB), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3".

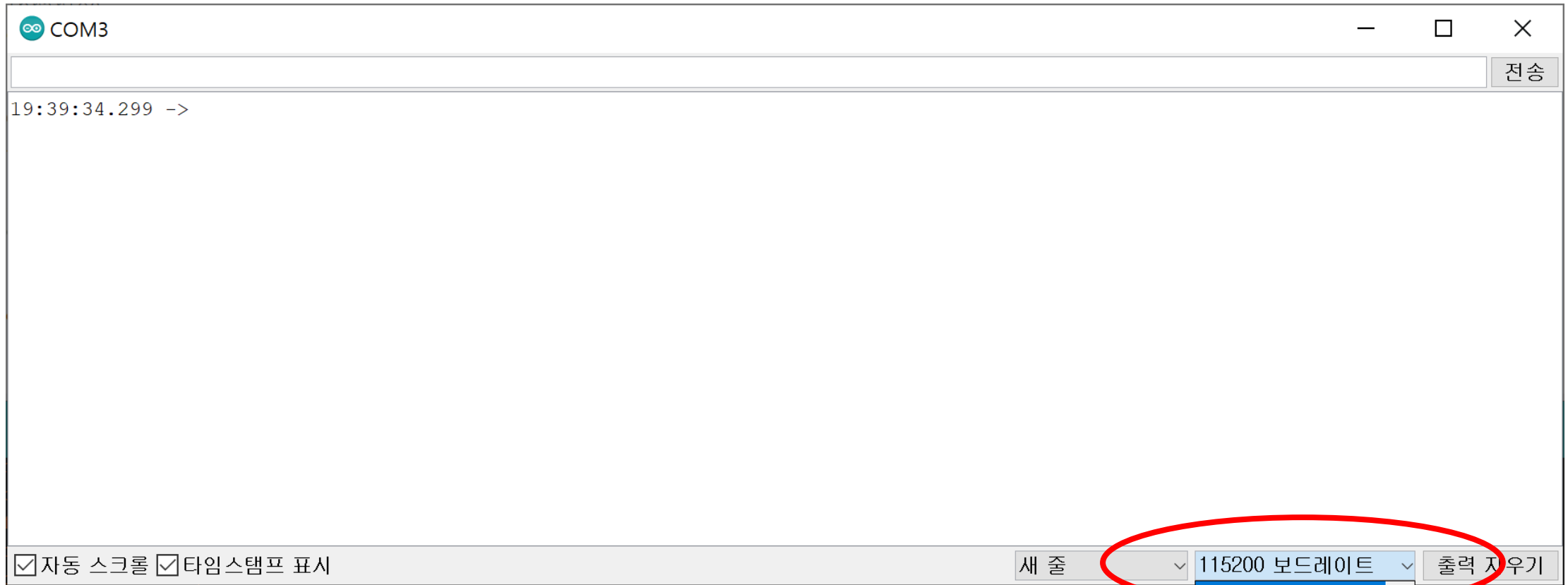
- 다음과 같이 업로드 퍼센트가 나오고 **업로드 완료**가 뜨면 성공입니다.

5. 시리얼 모니터 확인하기



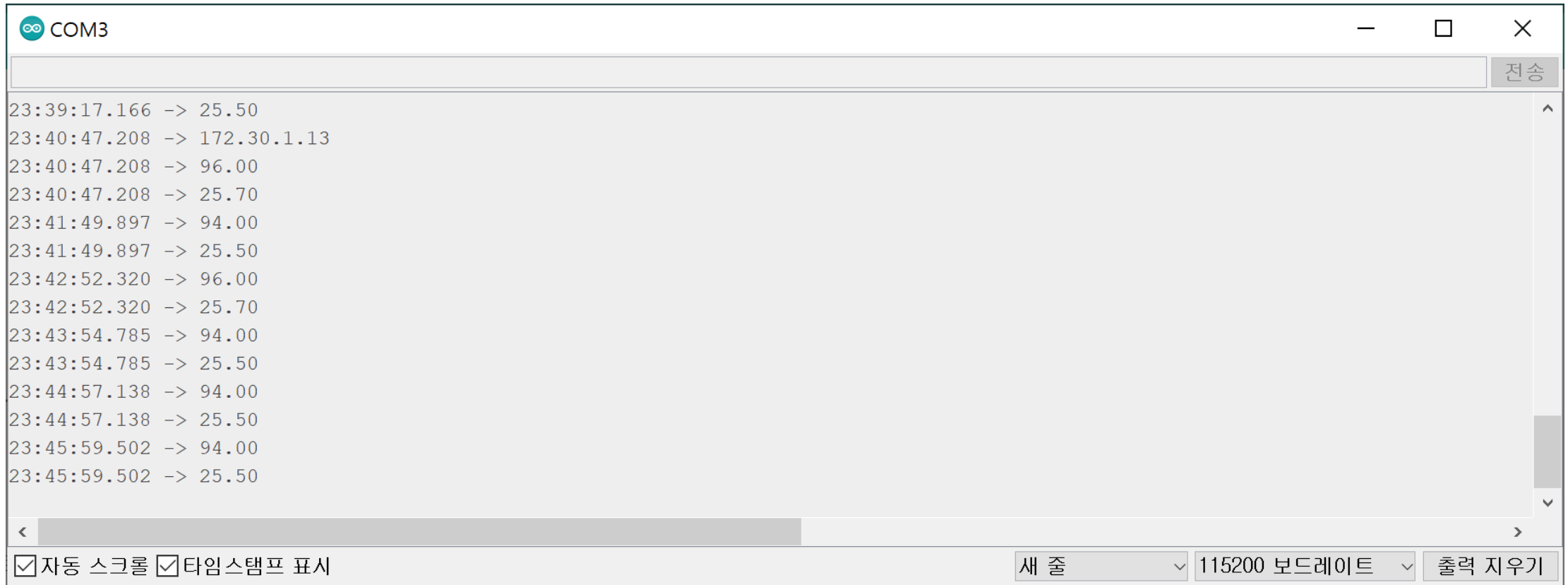
- 업로드 완료가 됐다면
툴 > 시리얼 모니터를 클릭

5. 시리얼 모니터 확인하기



창이 뜨면 오른쪽 아래 보드레이트를 115200로 맞춥니다.

5. 시리얼 모니터 확인하기

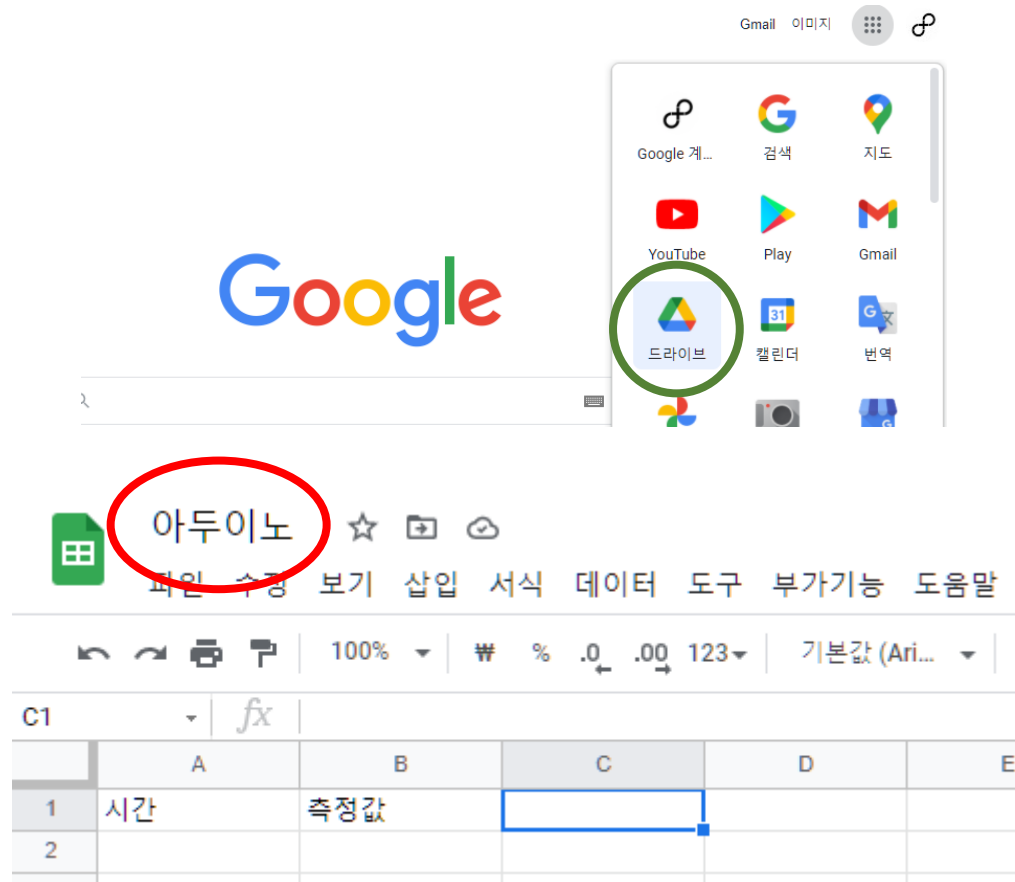


정상적으로 값이 나온다면 성공

FAQ. 실제 실내온도와 차이가 납니다.

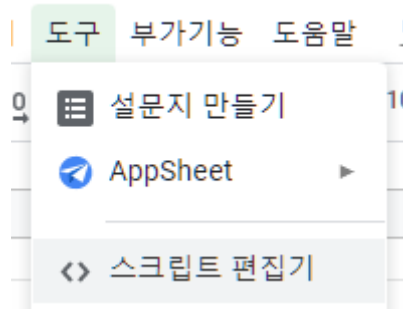
1. 다른 실내온도 측정기를 이용해 theta의 값을 고친 뒤 다시 업로드 해주세요.
 - 예) 실제실내온도 26.1, 측정값 25.5 일 경우. Theta의 값에 0.6을 더합니다.
2. 기본 측정값이 100에 가까운 수로 나오는지 확인해주세요.
 - 마지막에 나온 수가 온도, 그 직전 수가 기본 측정값입니다.
3. 만일 기본 측정 수치가 0이라면 센서의 선들이 정확하게 연결돼 있는지 확인해주세요.

6. 구글 스프레드시트로 센서 값 실시간 받기



1. 구글에 로그인 해주세요.
2. 구글 드라이브로 들어갑니다.(사진 참고)
3. 새로 만들기 > 구글 스프레드시트
4. 빨간색 동그라미 부분을 눌러 파일 이름을 정해주세요.
5. 셀 첫 번째 칸은 시간 두 번째는 측정값이라고 적습니다.

6. 구글 스프레드시트로 센서 값 실시간 받기



1. 도구 > 스크립트 편집기를 눌러 스크립트 편집기로 들어갑니다.
2. 아래의 코드를 복사해 붙여 넣어주세요.(사진 참고)

<http://colourscripiter.com/s/TmdR0y5>



 <https://docs.google.com/spreadsheets/d/1hEJAo8Crbbk1VvnnkojtxfFCEJMejKlsdqGAuoPY9tg/edit#gid=0>

1. 구글 스프레드시트로 돌아가서
2. 주소창의 드래그 된 부분을 복사해주세요.
 - 드래그 된 부분은 고유주소로 사진과 다른 것이 정상입니다.

Apps Script 아두이노

배포

이전 편집기 사용

파일 +

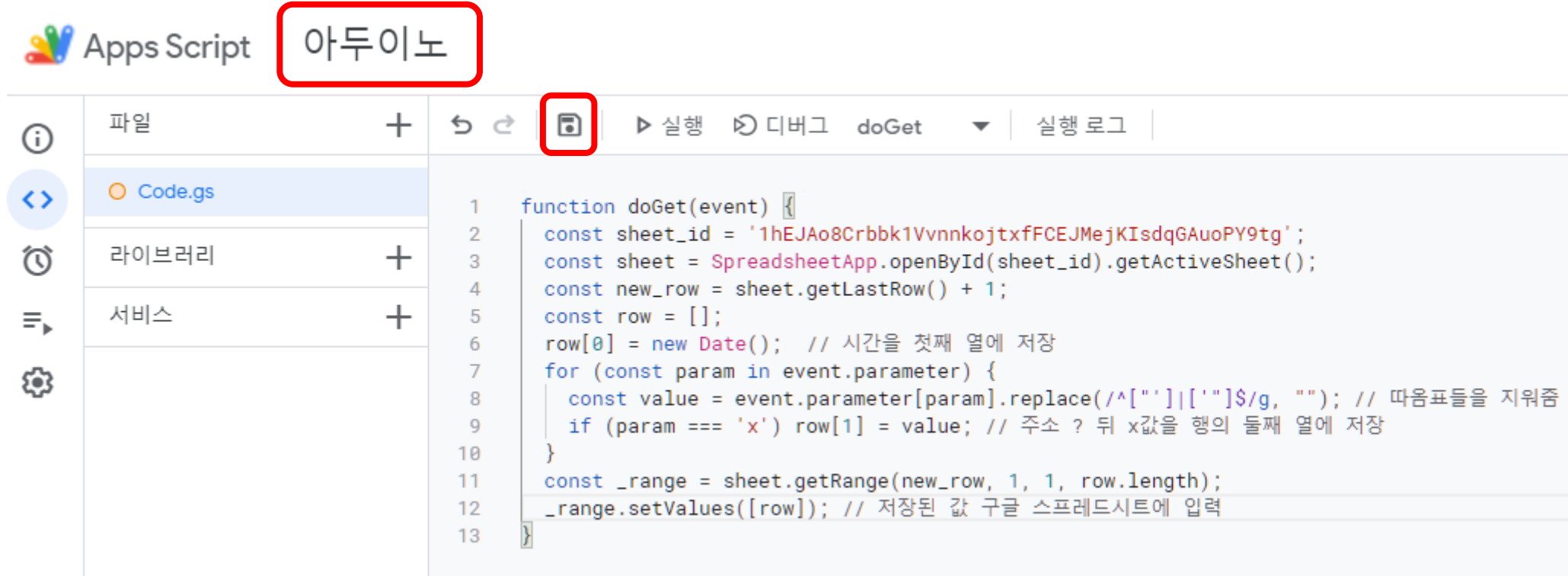
Code.gs

라이브러리 +

서비스 +

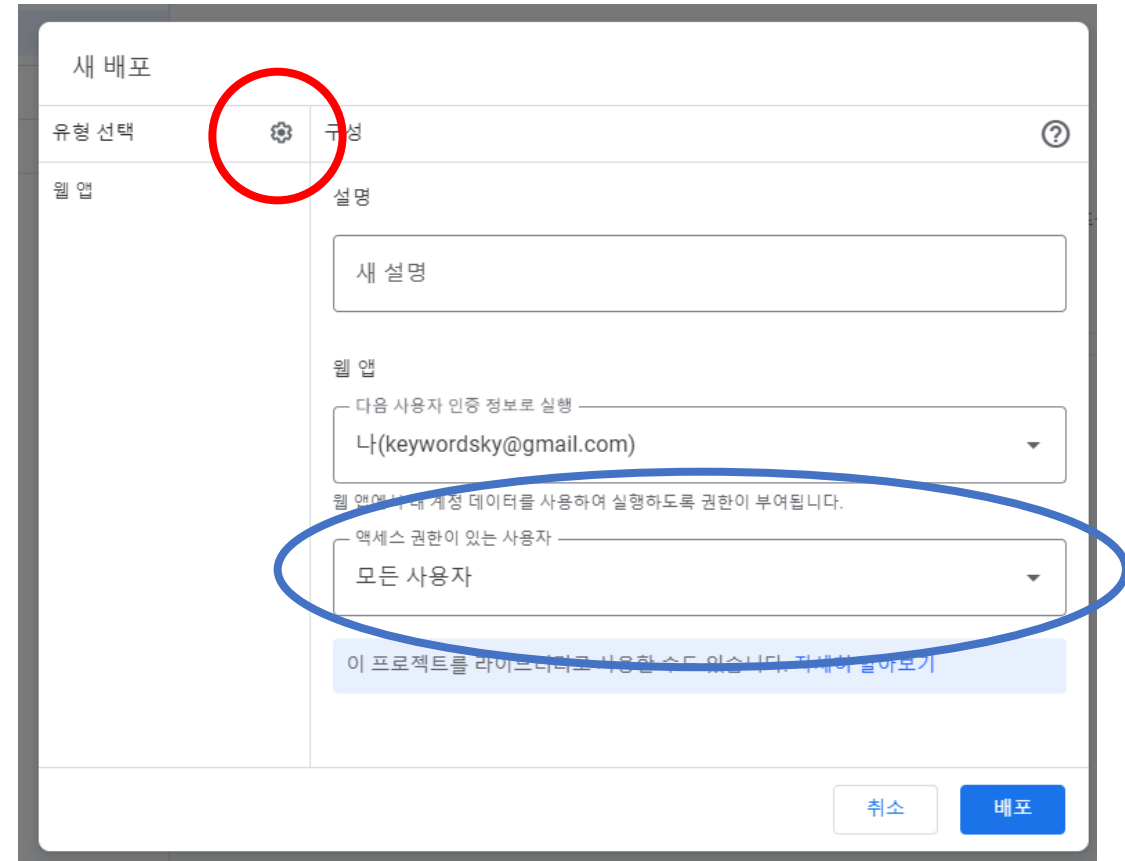
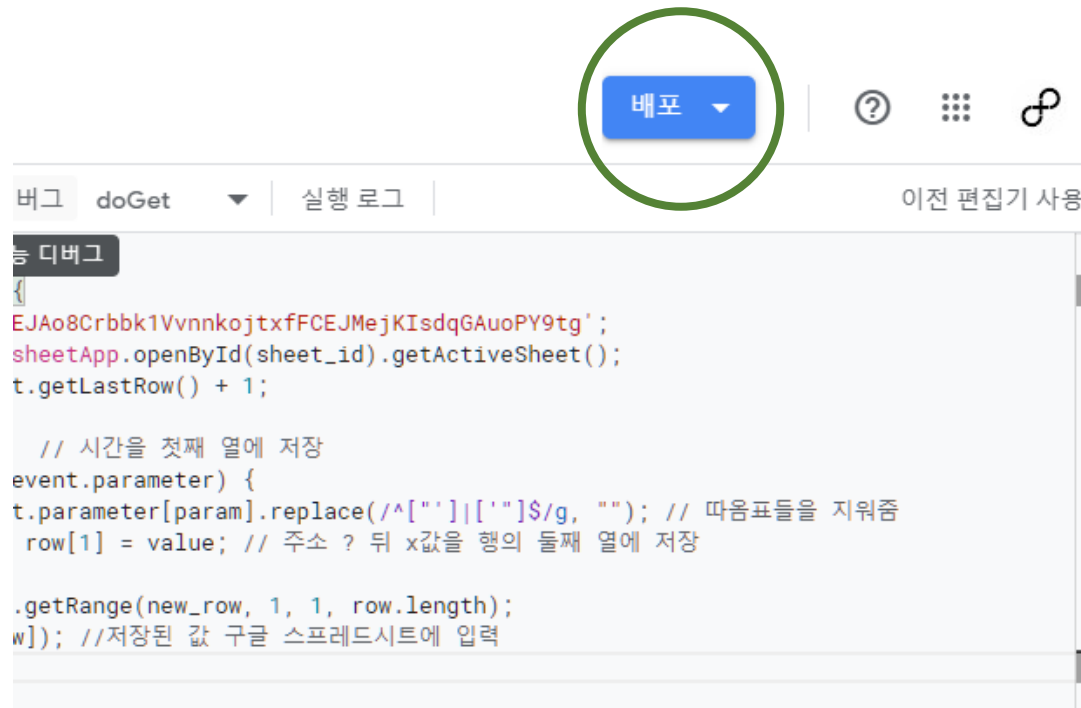
```
1 function doGet(event) {
2   const sheet_id = '1hEJAo8Crbbk1VvnnkojtxfFCEJMejKIsdqGAuoPY9tg';
3   const sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
4   const new_row = sheet.getLastRow() + 1;
5   const row = [];
6   row[0] = new Date(); // 시간을 첫째 열에 저장
7   for (const param in event.parameter) {
8     const value = event.parameter[param].replace(/^["]|["]$ /g, ""); // 따옴표들을 지워줌
9     if (param === 'x') row[1] = value; // 주소 ? 뒤 x값을 행의 둘째 열에 저장
10  }
11  const _range = sheet.getRange(new_row, 1, 1, row.length);
12  _range.setValues([row]); //저장된 값 구글 스프레드시트에 입력
13 }
```

방금 복사했던 값을 따옴표 안쪽에 정확히 넣어주세요.
따옴표가 지워지면 안됩니다.



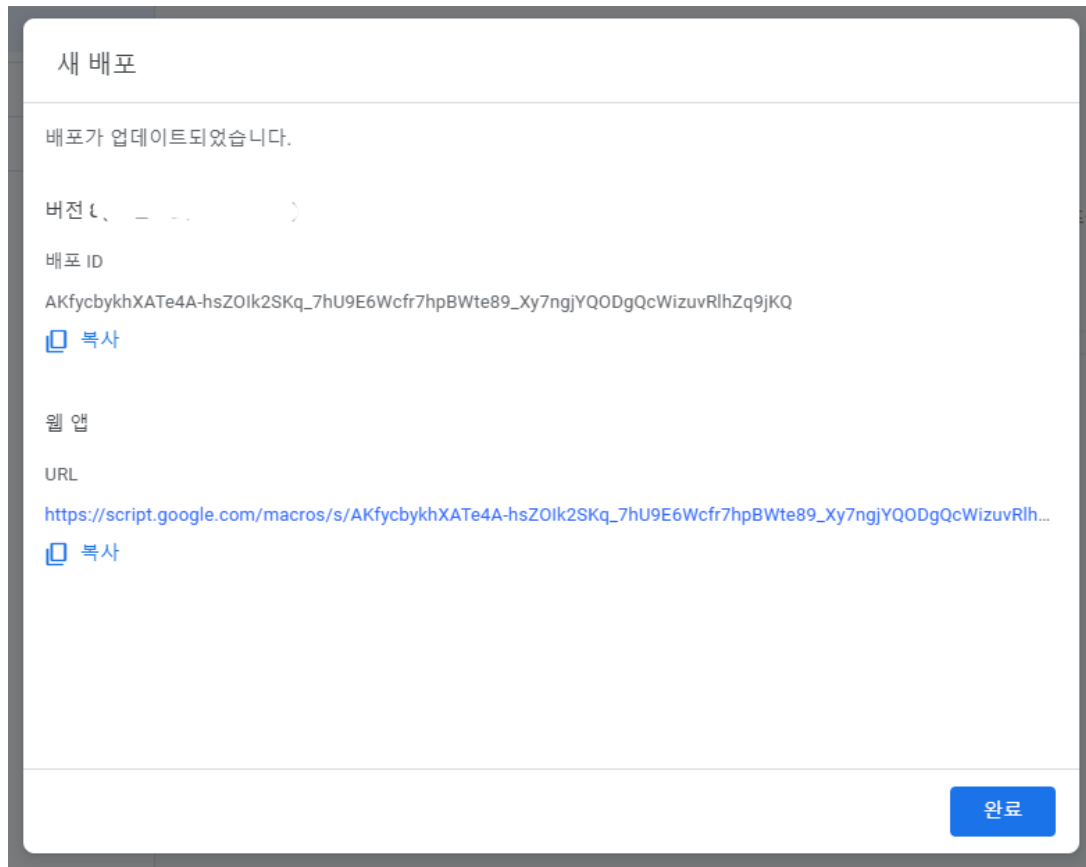
복사 완료 후 상단에 이름을 정한 뒤 저장 버튼을 꼭 눌러주세요.

6. 구글 스프레드시트로 센서 값 실시간 받기



배포 버튼 클릭 > 새 배포 > **톱니바퀴** > 웹 앱 클릭
액세스 권한이 있는 사용자 > 모든 사용자 로 맞춰주세요.

6. 구글 스프레드시트로 센서 값 실시간 받기



The image shows a '새 매크로' (New Macro) dialog box in Google Sheets. It contains the following information:

- Header: 새 매크로
- Status: 매크로가 업데이트되었습니다.
- Version: 버전 1.0.0
- ID: 매크로 ID: AKfycbykhXATe4A-hsZOIk2SKq_7hU9E6Wcfr7hpBWte89_Xy7ngjYQODgQcWizuvRlhZq9jKQ
- Copy button: 복사
- Web App: 웹 앱
- URL: URL: https://script.google.com/macros/s/AKfycbykhXATe4A-hsZOIk2SKq_7hU9E6Wcfr7hpBWte89_Xy7ngjYQODgQcWizuvRlhZq9jKQ
- Copy button: 복사
- Finish button: 완료

- 매크로가 완료되면 URL을 복사합니다.
- 매크로 과정에서 인증이 나오면 인증해주세요.
- 안전하지 않은 접근이라는 경고가 나오면 ▽버튼을 눌러서 허용해주세요.

7. 스프레드시트 입력 확인해보기

1. 복사한 URL 뒷 부분에 ?x=10 을 입력해봅시다. (아래는 예시)

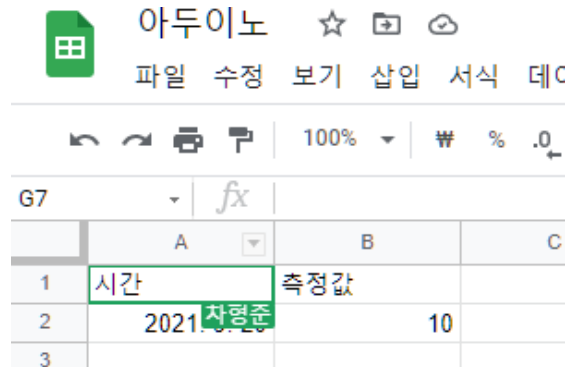
<https://script.google.com/macros/s/AKfycbyG3K4sireewChPI6gqTgt0EQvU6Rw83tKH0tNN0qIKx7SqBFCRi31gmI20PUwVDgxSug/exec?x=10>

2. 다음처럼 뜬다면 성공

Google Apps Script

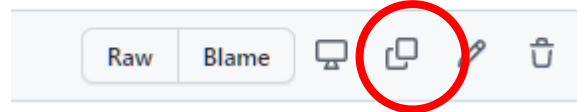
스크립트는 완료했지만 아무것도 반환하지 않았습니다.

3. 스프레드시트에서 입력시간과 10이 생겼는지 확인해보세요.



아두이노		
파일	수정	보기
100% 100%		
G7 fx		
A	B	C
1 시간	측정값	
2 2021-01-01	10	
3		

8. 아두이노 코드를 바꿔 업로드하기



1. 아래의 링크로 들어가 코드를 복사하여 아두이노에 붙여 넣어주세요.
(그림과 같은 버튼을 누르면 한 번에 복사됩니다.)

https://github.com/cherub8128/Lecture/blob/master/temperature_Wifi.ino

2. 복사 후 다음 부분의 **따옴표 안**을 내용에 맞게 적절히 수정해주세요.

```
char* ssid = "와이파이 이름";  
char* pw = "와이파이 비번";  
String url = "구글 스크립트 편집기 배포 웹 앱 URL";|
```

3. 다시 업로드 버튼을 눌러 업로드 후 시리얼 모니터를 열어 확인합니다. 정상적으로 접속 됐다면 ip주소가 모니터에 출력됩니다.
- Url은 6번 단계에서 복사한 주소입니다.

8. 아두이노 코드를 바꿔 업로드하기

sketch_sep29b | 아두이노 1.8.15

파일 편집 스케치 툴 도움말



sketch_sep29b \$

```
#include <Arduino.h>
//와이파이 라이브러리
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>

ESP8266WiFiMulti WiFiMulti; //WiFi 관련 설정
int minute = 60000;
float theta = 16.1;

//아래의 두 줄에서 따옴표 안쪽 부분만 수정해주세요.
char* ssid = "와이파이 이름";
char* pw = "와이파이 비밀번호";
String url = "구글 스크립트 편집기 배포 웹 앱 URL";

void setup() {
    //시리얼 통신 시작(보 레이트 - 1초에 몇 바이트 속도로 통신할 것인지)
    Serial.begin(115200);
    Serial.print("모니터링 시작");

    //와이파이 접속 부분
    WiFi.mode(WIFI_STA);
    WiFiMulti.addAP(ssid, pw);
    while ((WiFiMulti.run() != WL_CONNECTED))
    {
```

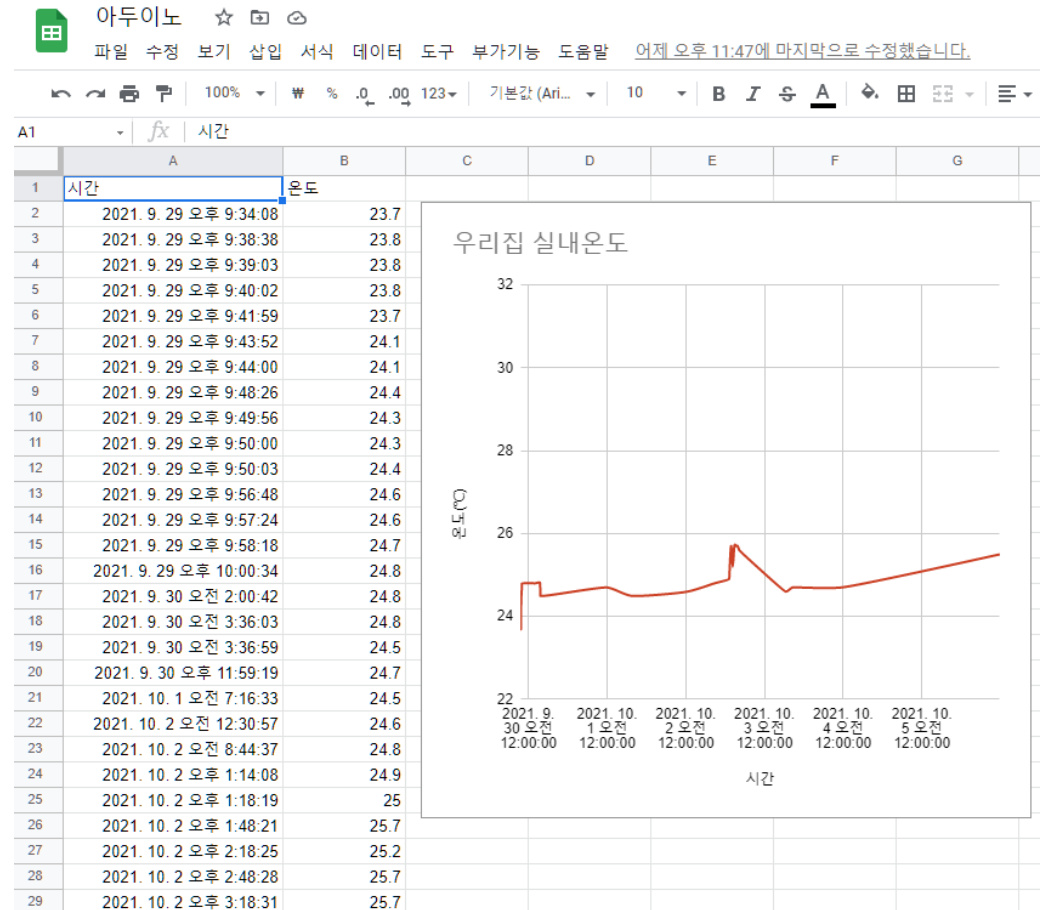
위와 같이 붙여넣고 빨간 동그라미 부분의 **따옴표 안**을 고친 후 업로드 버튼을 눌러주세요.

9. 스프레드시트에 업로드 되는지 확인하기

- 아두이노 기기가 켜진 상태에서 정상적으로 작동될 시 다음과 같이 측정값으로 셀이 추가됩니다.
- 업로드 속도가 너무 빠르다면 코드의 아래 부분의 1을 원하는 간격으로 고쳐주세요.
예) 20분일 땐, 20*minute

```
56 // 값 업로드 성공하면 1분 간 기다리기  
57 delay(1*minute);
```

- 그래프는 삽입 > 차트로 그릴 수 있습니다.



별첨: 온도 센서 수식

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±1/4°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

- 이 과정은 건너뛰셔도 됩니다.

- 제품페이지에서 설명을 읽어보면
섭씨 1도당 10mV의 값으로 변화
한다는 것을 알 수 있습니다.

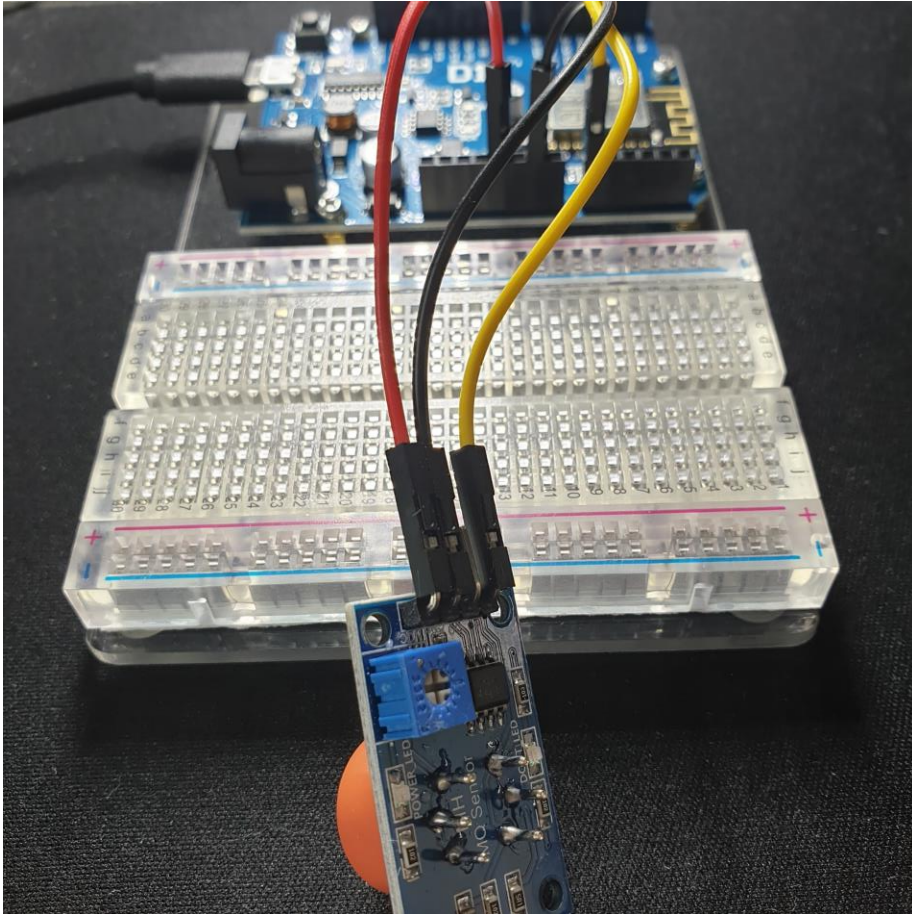
- 이를 이용해 다음과 같은 수식으로 코딩하였습니다.

$$temperature = \frac{A0 \text{ 입력값}}{10} + \theta$$

아두이노 알코올 센서 튜토리얼

알코올 센서로 실시간 측정값 받기

1. 알코올 센서 연결하기

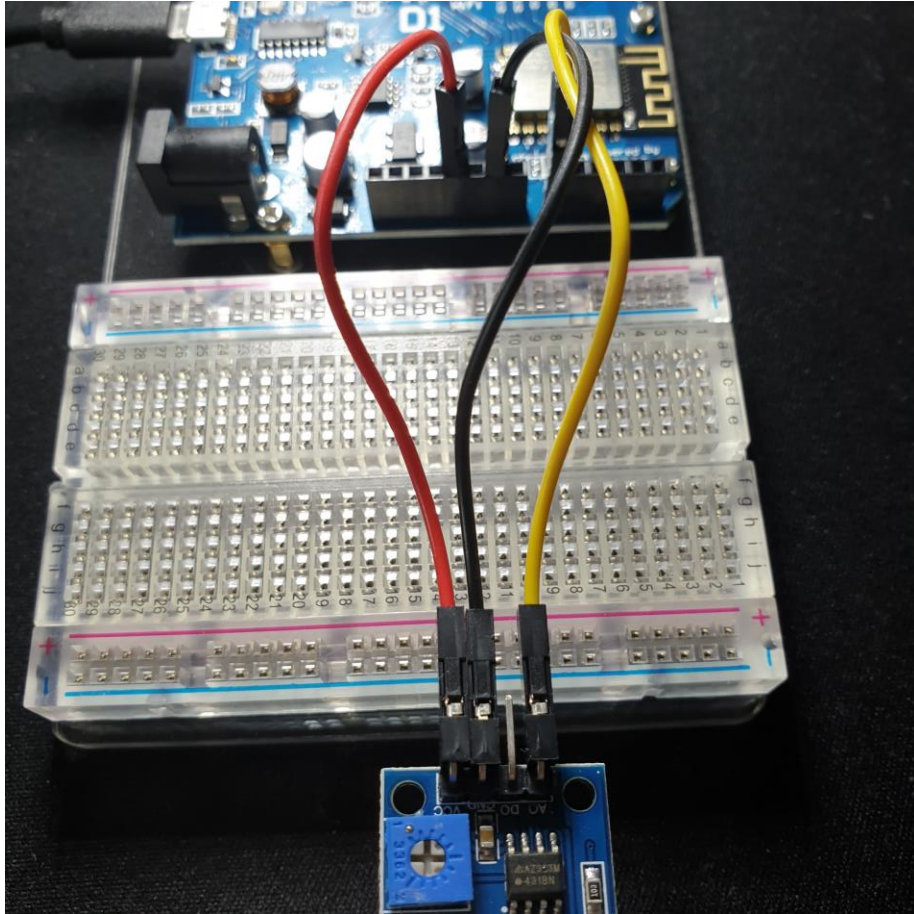


1. 주황색MQ-3 센서입니다.
2. 암-수 점프선으로 사진과 같이 연결합니다.

GND-까만색 선-GND
VCC-빨간색 선-5V
A0-노란색 선-A0
D0-연결 안함

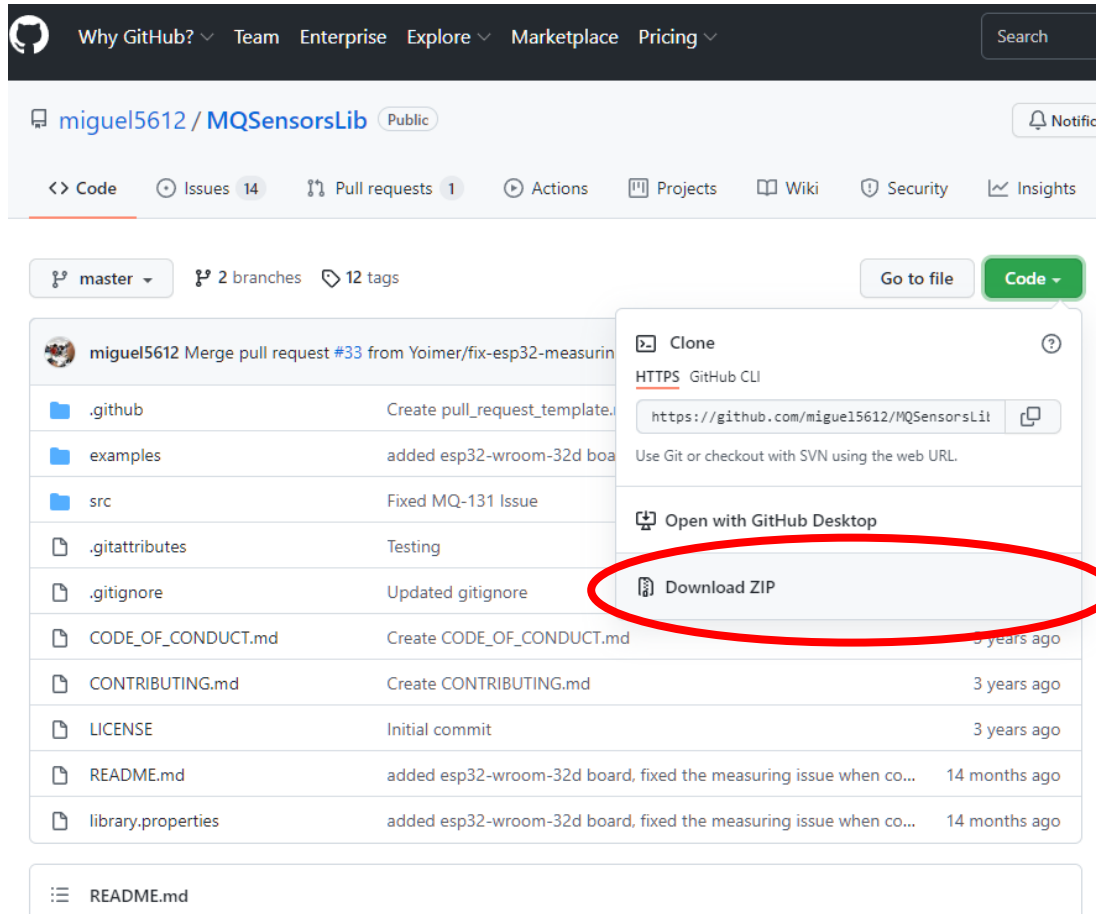
- Tip : 되도록 같은 색깔의 선을 사용하는 것이 좋습니다.

1. 알코올 센서 연결하기



• 연결 된 사진입니다.

2. 알코올 센서 코드 라이브러리 다운로드



- MQ 시리즈 센서 링크
- 위의 링크 접속 후
- Code → Download ZIP 으로 다운로드합니다.
- 다운만 받고 압축은 풀지 말아주세요.

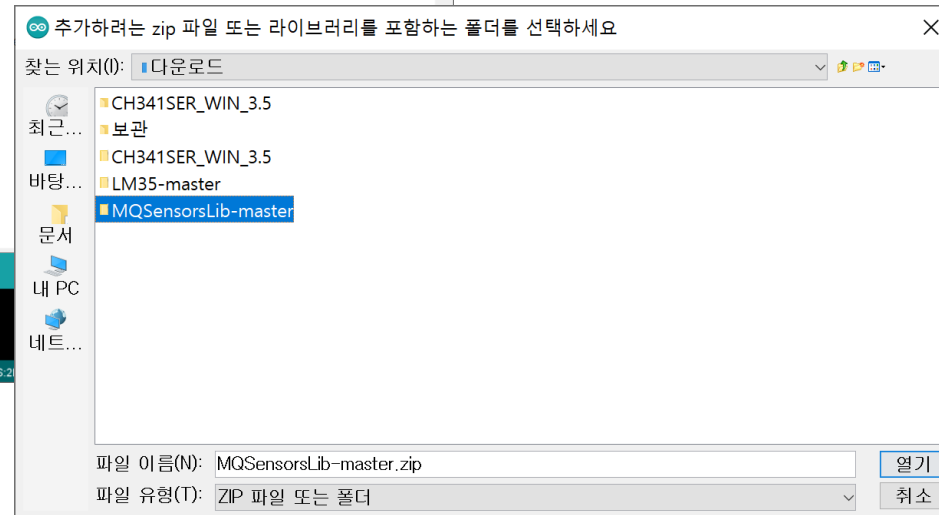
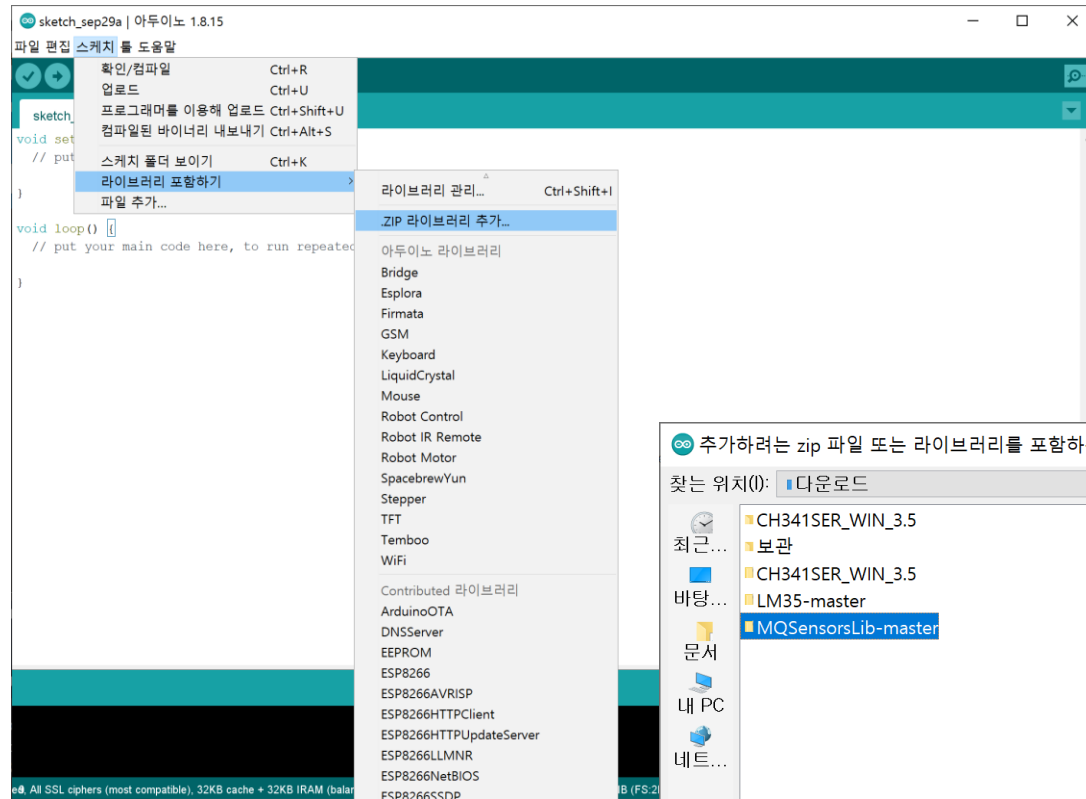
3. 알코올 센서 코드 라이브러리 설치

1. 스케치

- 라이브러리 포함하기
- .ZIP 라이브러리 추가

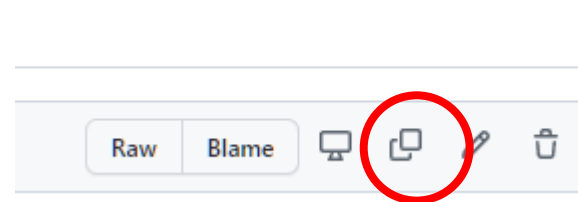
2. 앞에서 받은 파일을 선택한 뒤

3. 열기를 눌러 추가해 주세요.



2. 알코올 센서 코드 복사 붙여넣기

1. 아두이노 프로그램을 열어
2. 파일 → 새파일을 누르고
3. 아래의 링크로 들어가 코드를 복사 붙여넣기 해주세요.
https://github.com/cherub8128/Lecture/blob/master/MQ3_Wifi.ino
4. 앞서서 했던 것과 같이 다음 3줄의 따옴표 안을 수정해주세요.

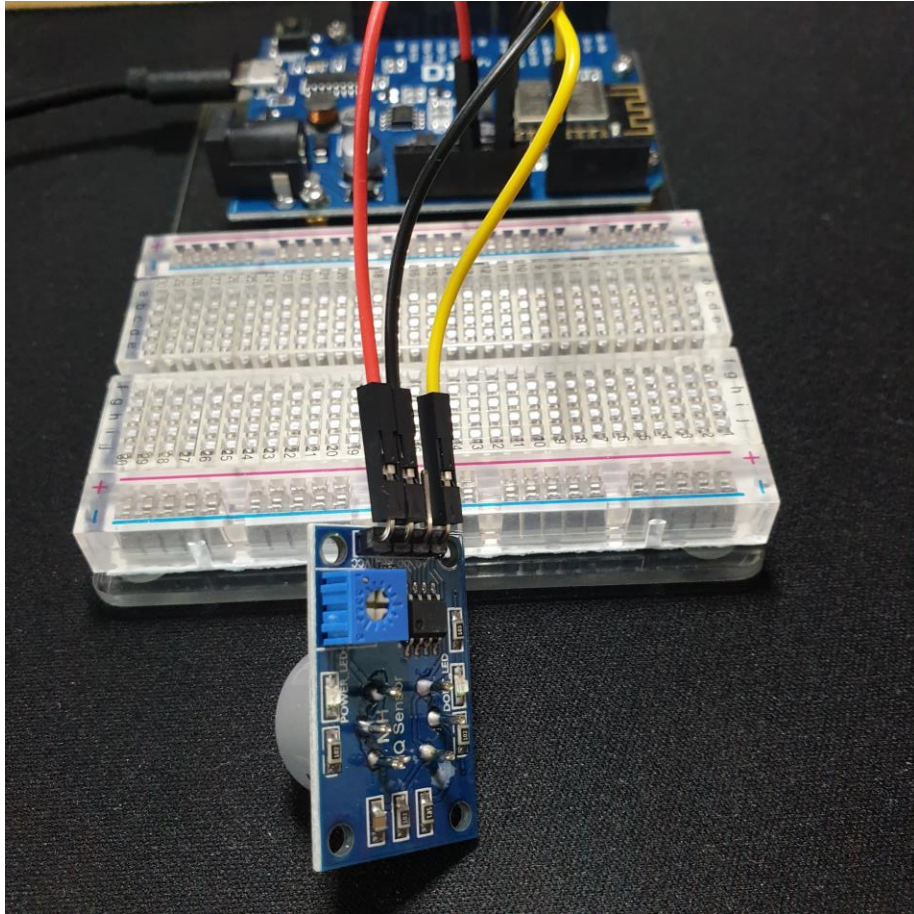


```
char* ssid = "와이파이 이름";  
char* pw = "와이파이 비번";  
String url = "구글 스크립트 편집기 배포 웹 앱 URL";
```

3. 시리얼 모니터와 스프레드시트로 확인하기

- 1. 시리얼 모니터를 열어서 측정값을 확인해보세요.**
 - 2. 스프레드 시트에도 정상적으로 업로드 되는지 확인해보세요.**
- 측정 간격은 5초로 되어 있습니다.**

4. 일산화탄소 센서로 응용



- 왼쪽과 같이 센서를 연결합니다. 알코올센서와 동일합니다.
- 다음 링크의 코드를

https://github.com/cherub8128/Lecture/blob/master/MQ7_Wifi.ino

- 복사 붙여넣기 후
- 앞선 과정과 동일하게 수정해야 하는 부분만 수정해서 업로드해 주세요.

```
char* ssid = "와이파이 이름";  
char* pw = "와이파이 비번";  
String url = "구글 스크립트 편집기 배포 웹 앱 URL";
```

5. 이산화탄소 센서

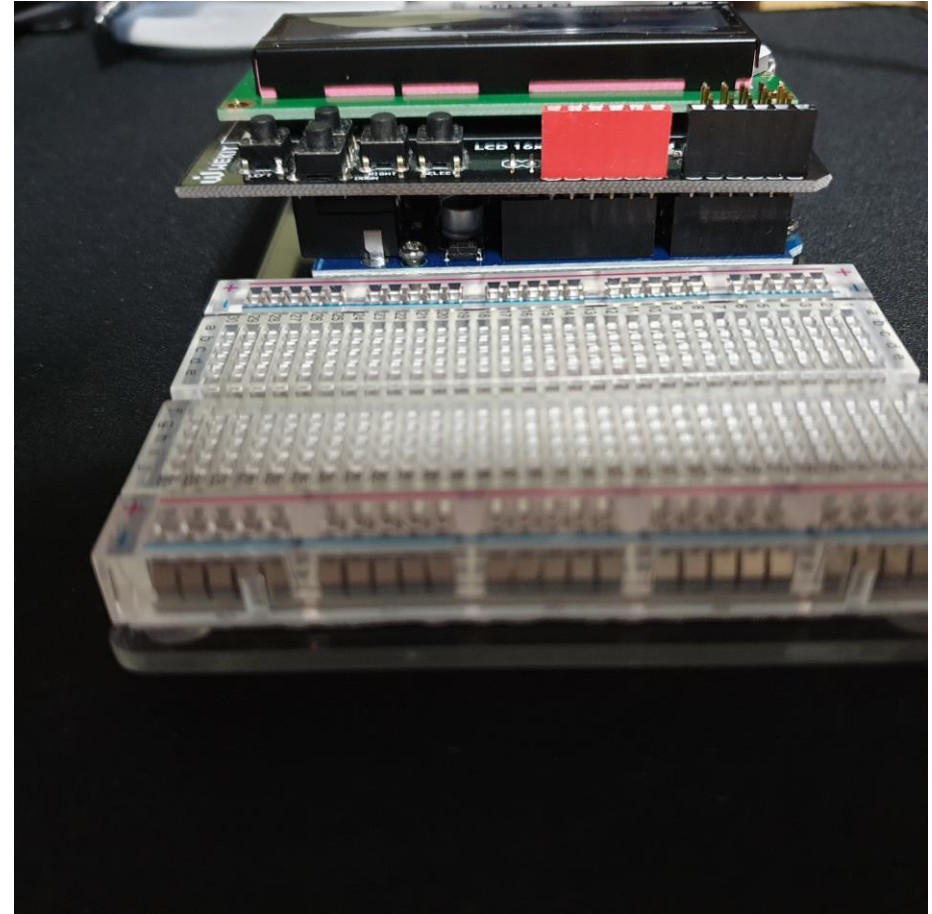
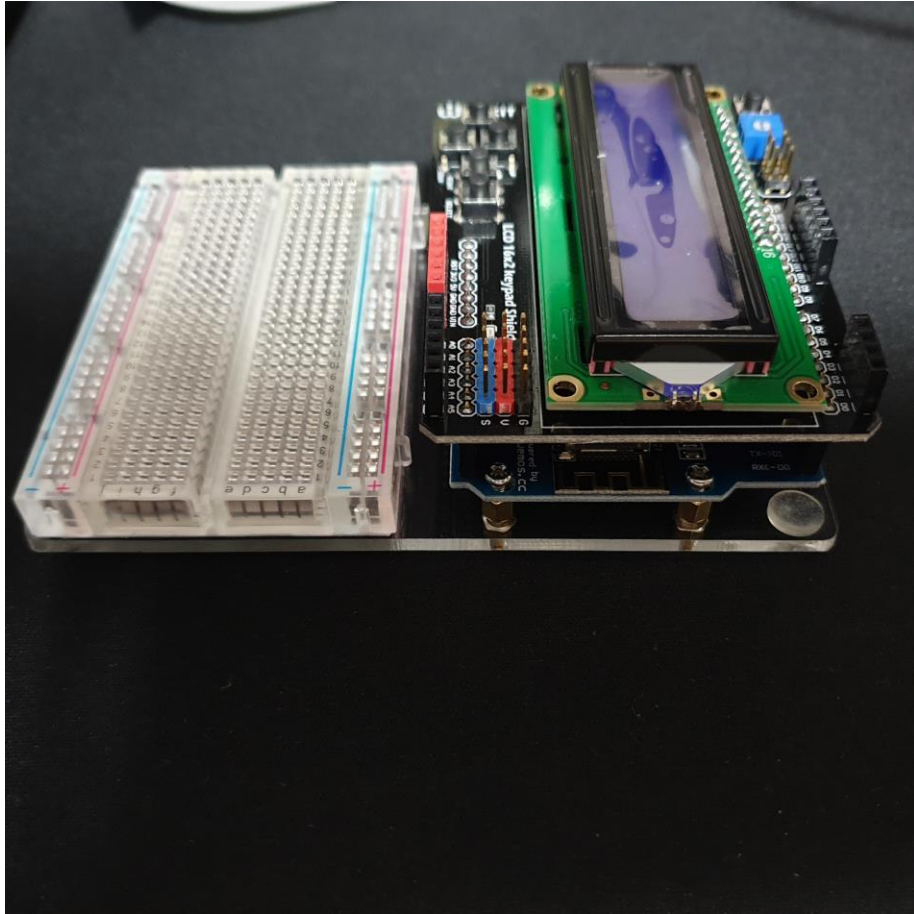
- 현재 센서 단자에 맞는 선이 준비되지 않아 테스트를 못했습니다.
- 구해지게 되면 전달해드리겠습니다.
- 아래는 예제 코드입니다. 앞선 과정과 완전히 동일합니다.
https://github.com/cherub8128/Lecture/blob/master/CO2_Wifi.ino
- 복사 붙여넣기 후 앞서서 했던 것처럼 **아래의 부분만** 수정해주세요.

```
char* ssid = "와이파이 이름";  
char* pw = "와이파이 비번";  
String url = "구글 스크립트 편집기 배포 웹 앱 URL";|
```

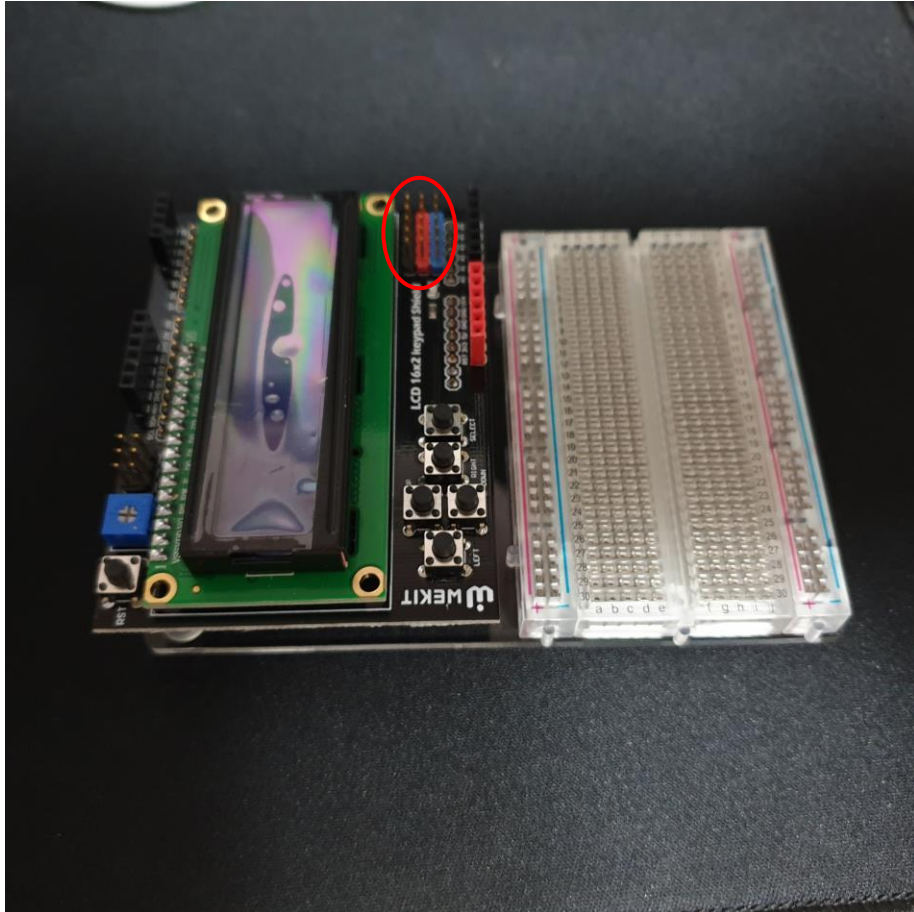
번외: 아두이노 키패드 실드

이 과정은 넘어가도 좋습니다.

아두이노 키패드 쉴드 조립 사진



아두이노 키패드 실드 조립



- 키패드 실드를 다음과 같이 꽂아주세요.
- 핀의 모양이 정확히 맞는지 확인하고 꽂습니다.
- TIP
 - 키패드 실드의 남은 핀에 아날로그, 디지털, 전원 연결이 가능합니다.
 - 센서를 쉽게 꽂을 수 있는 핀이 있습니다. **(빨간색 동그라미)**

아두이노 키패드 쉴드 사용법

- 키패드 쉴드는 아두이노 UNO(일반적인 아두이노)에 사용시 정상적으로 사용가능합니다.
- 아래는 키패드 쉴드 기본 코드 링크입니다.
복사 붙여넣기 하여 활용 가능합니다.

https://github.com/cherub8128/Lecture/blob/master/keypad_shield.ino

- 안타깝게도... 키패드 쉴드는 아두이노 WiFi에 사용시 아날로그 입력이 불가능합니다. (WiFi보드에서 아날로그 입력 A1~이하의 모든 단자가 사용 불가능하기 때문...)

마무리하며...

아두이노 활용 교육에 대한 개인적 의견

- 아두이노를 초중등에서 이미 배워오는 경우가 많습니다.
(Micro bit 라는 더 쉬운 교구도 있습니다.)
- 인터넷에서 찾아서 따라 만드는 것은 학생들이 잘합니다.(이유도 모른채...)
- 학생들의 갈증. 고등학생답게 아두이노 활용하기.
 - 초중등에서는 원리를 잘 모른채 따라 만들기를 했다면
 - 고등학교에서는 센서의 과학적 원리를 높은 지식 수준에서 해결할 수 있습니다.
 1. 온도,에탄올,이산화탄소의 변화가 어떻게 전압의 변화로 바뀌는 지
 2. 실험 데이터를 통해 원하는 값을 어떤 수식으로 변환할 수 있는지(영문 센서 설명서)
 3. 센서의 측정 오차 등 정확도와 정밀도의 수준은 어떠한지
 - 단순히 코딩을 잘하는 것을 가르치는 것보다는
 - 과학적 원리를 이해해서 수식을 짤 수 있다면 코딩은 익숙함의 문제로 바뀝니다.
 - 분석 할 수 있는 능력이 고등학교의 차별화된 아두이노 교육이 될 것 같습니다.