



통합센서보드 매뉴얼 v0.3.0

2024.09.01
(주) 다운정보기술 양찬솔



WWW.DAONINFO.COM

Contents



- ▶ ESP32 보드 개발 환경 구성 (택 1)
 - [Arduino IDE \(라이브러리 수동 설치 / 사용 환경 수동 설정\)](#)
 - [VS Code + PlatformIO \(자동: 개발자가 사전에 정의한 사용 환경으로 자동 구성\)](#)
- ▶ 보드 매뉴얼
 - 통합센서보드 초기화 설정
 - 센서 하드웨어 연결
- ▶ 오류 감지
 - 센서 보드 Disconnect
- ▶ 추가 기능
 - 무선통신 추가 센서
 - 무선통신 배터리 전압 측정
- ▶ 부록
 - RK520-02 전원 공급 관련 이슈
 - 단종 - 감우 센서 연결 상세

Arduino IDE 구성

Arduino IDE



▶ 1. 프로젝트 경로

- Github: https://github.com/genesis0409/esp32_Daon_uxF_ISMS
- Arduino 소스코드 경로:
esp32_Daon_uxF_ISMS\esp32_Daon_uxF_ISMS_Arduino
(메인 프로젝트 내 Arduino 프로젝트 폴더)

이름	유형
.git	파일 폴더
.pio	파일 폴더
.vscode	파일 폴더
data	파일 폴더
esp32_Daon_uxF_ISMS_Arduino	파일 폴더
include	파일 폴더
lib	파일 폴더
src	파일 폴더
test	파일 폴더
.gitignore	Git Ignore 원본 파..
huge_app.csv	Excel.CSV
platformio.ini	INI 원본 파일

이름	유형
data	파일 폴더
esp32_Daon_uxF_ISMS_Arduino.ino	Arduino file
InputParams.cpp	C++ 원본 파일
InputParams.h	C Header 원본 파일

Arduino IDE



▶ 2. ESP32 보드매니저 등록

- 상단 메뉴) 파일 -> 환경설정 -> 추가적인 보드 매니저 URLs 에 아래 주소 입력 후 확인
- https://github.com/espressif/arduino-esp32/releases/download/2.0.4/package_esp32_dev_index.json

환경설정

설정 네트워크

스케치북 위치:
C:\Users\Wjd\Documents\Arduino 찾아보기

에디터 언어: System Default (아두이노를 재시작해야 함)

에디터 글꼴 크기: 16

Interface scale: ☒ 자동 100% (아두이노를 재시작해야 함)

테마: 디폴트 테마 (아두이노를 재시작해야 함)

다음 동작중 자세한 출력 보이기: ☐ 컴파일 ☐ 업로드

컴파일러 경고: None

☒ 줄 번호 표시 ☐ 코드 폴딩 사용하기

☒ 업로드 후 코드 확인하기 ☐ 외부 에디터 사용

☒ 시작시 업데이트 확인 ☒ 검증 또는 업로드 할 때 저장하기

☐ Use accessibility features

추가적인 보드 매니저 URLs 복사

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다
C:\Users\Wjd\AppData\Local\Arduino15\preferences.txt
(아두이노가 실행되지 않는 경우에만 수정 가능)

확인 취소

Arduino IDE



- 상단 메뉴) 툴 -> 보드 -> 보드매니저 선택 : 'esp32' 입력하여 패키지 설치 (*230424 버전 2.0.4)
- 보드는 'ESP32 Dev Module' 선택 / 그 외의 값들은 기본값으로 설정
- [중요] 포트는 개발 보드와 실제 연결된 COM포트 선택

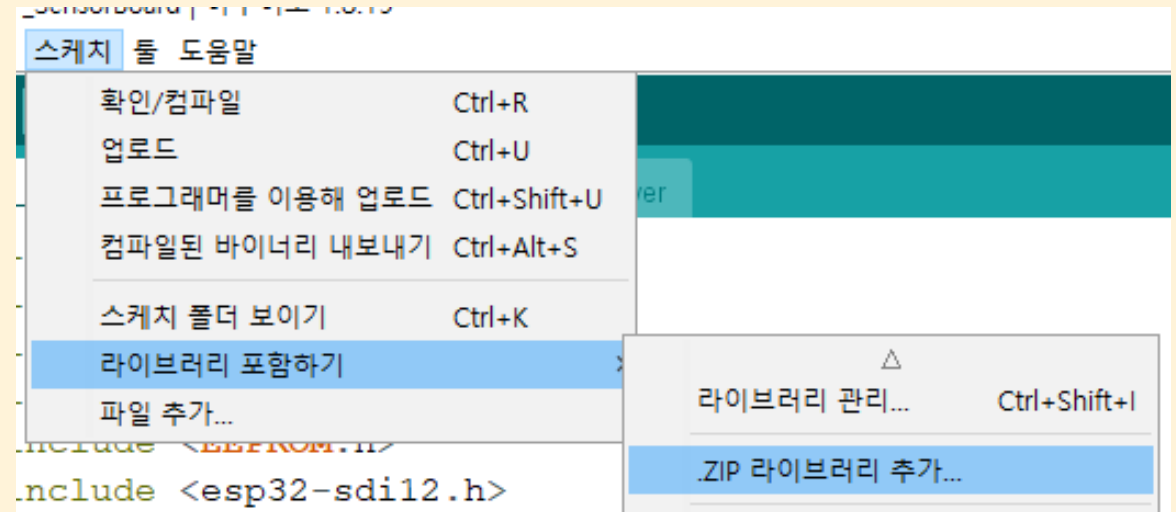
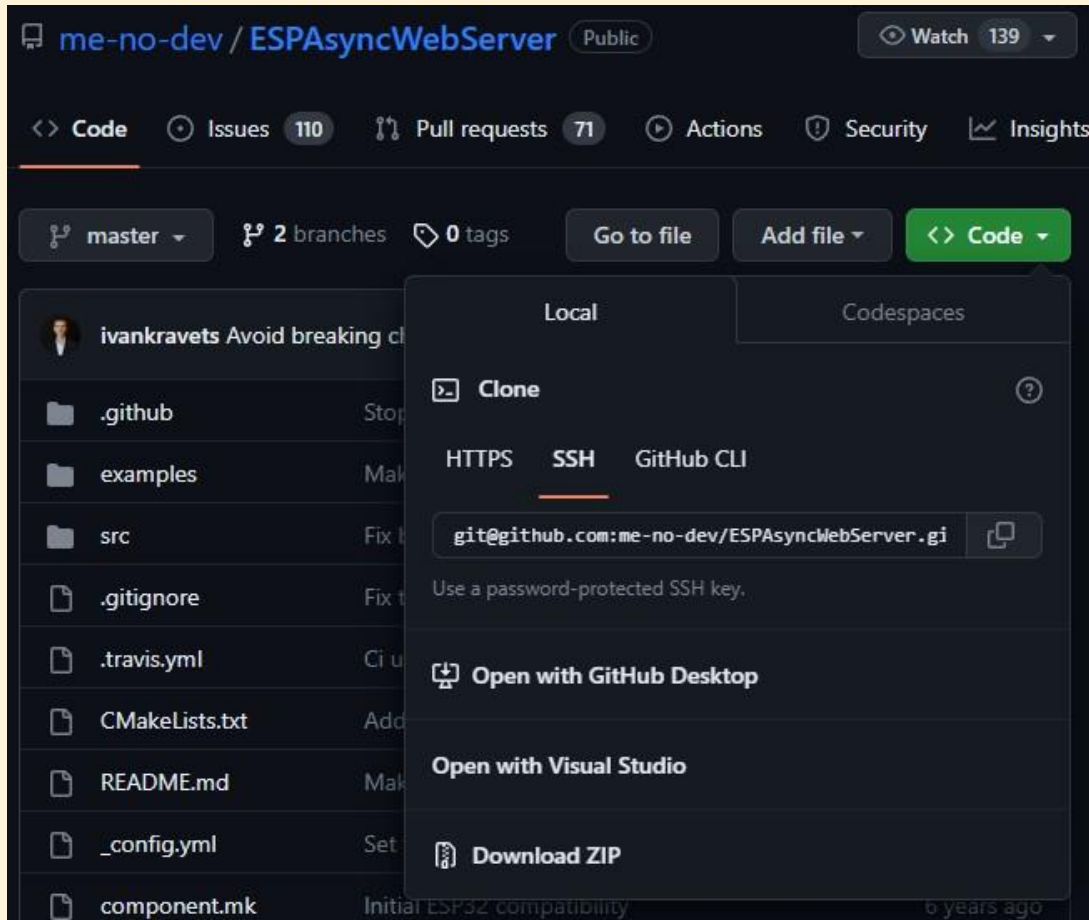
```
보드: "ESP32 Dev Module" >
Upload Speed: "921600" >
CPU Frequency: "240MHz (WiFi/BT)" >
Flash Frequency: "80MHz" >
Flash Mode: "QIO" >
Flash Size: "4MB (32Mb)" >
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" >
Core Debug Level: "None" >
PSRAM: "Disabled" >
Arduino Runs On: "Core 1" >
Events Run On: "Core 1" >
포트: "COM12" >
보드 정보 얻기
```

Arduino IDE



▶ 3. 라이브러리 설치

- Download ZIP 으로 파일 다운로드 후 Arduino IDE에서 '**ZIP 라이브러리 추가**'로 경로 지정



Arduino IDE



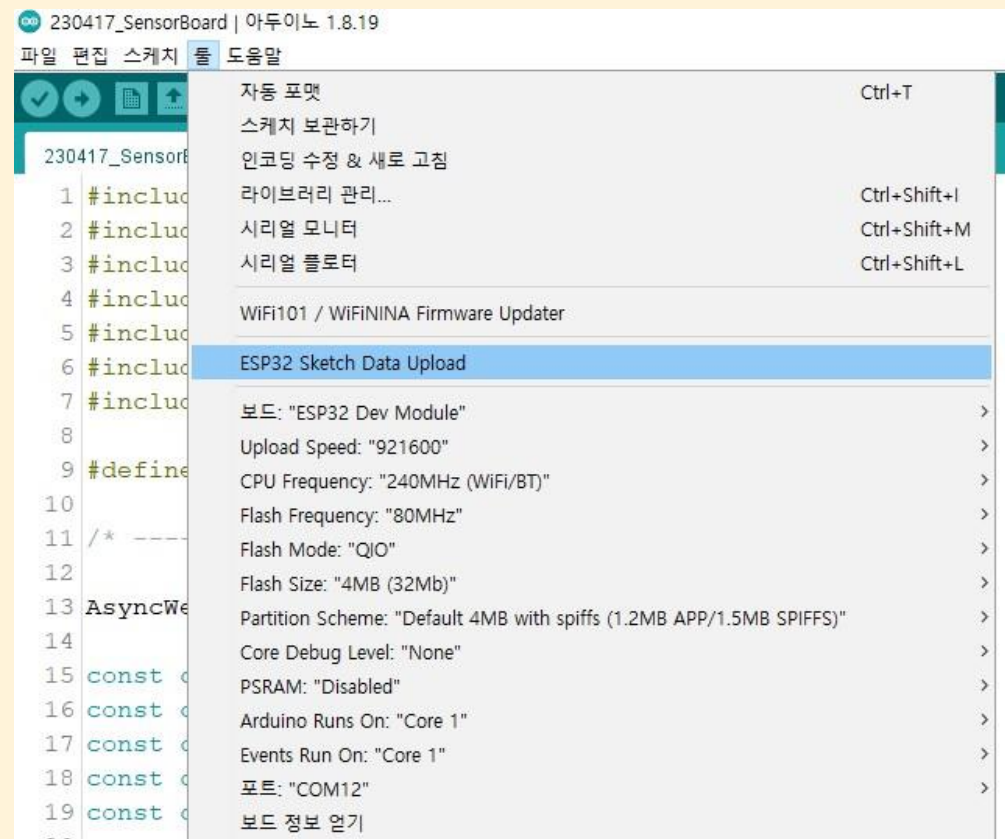
- ZIP 라이브러리 설치
 - <https://github.com/me-no-dev/ESPAsyncWebServer>
 - <https://github.com/me-no-dev/AsyncTCP>
 - <https://github.com/HarveyBates/ESP32-SDI12>
- 라이브러리 매니저에서 설치
 - CRC by RobTillaart 0.3.1
- SPIFFS 설치 및 참조 링크
 - SPIFFS 1.1
 - <https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/>

Arduino IDE



▶ ESP32 Filesystem Uploader

- 기능 : ESP32 Flash 메모리에 파일 업로드
 - SPIFFS 를 설치하면서 Arduino ESP32 Filesystem Uploader 환경을 구축할 수 있다.
 - 연결된 COM포트 선택 후 'ESP32 Sketch Data Upload' 로 데이터 폴더의 파일을 업로드한다
-
- 해당 기능은 크게 2가지 역할로 사용한다.
 1. 프로젝트의 웹 페이지 구성 파일을 보드에 업로드
 2. 웹 페이지를 통한 파라미터 설정 값 초기화



VS Code + PlatformIO 구성

VS Code + PlatformIO



▶ 1. VS Code 및 Python 설치

- 파일 -> 환경설정 -> 추가적인 보드 매니저 URLs 에 아래 주소 입력 후 확인
- <https://randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino/>

▶ 2. VS Code에 PlatformIO IDE 확장 프로그램 설치

- <https://randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino/#2>

VS Code + PlatformIO



▶ 3. VS Code로 메인 디렉토리 열기

- 상단 메뉴) 파일 -> 폴더 열기 -> 'esp32_Daon_uXF_ISMS' 열기

❖ Data 디렉토리

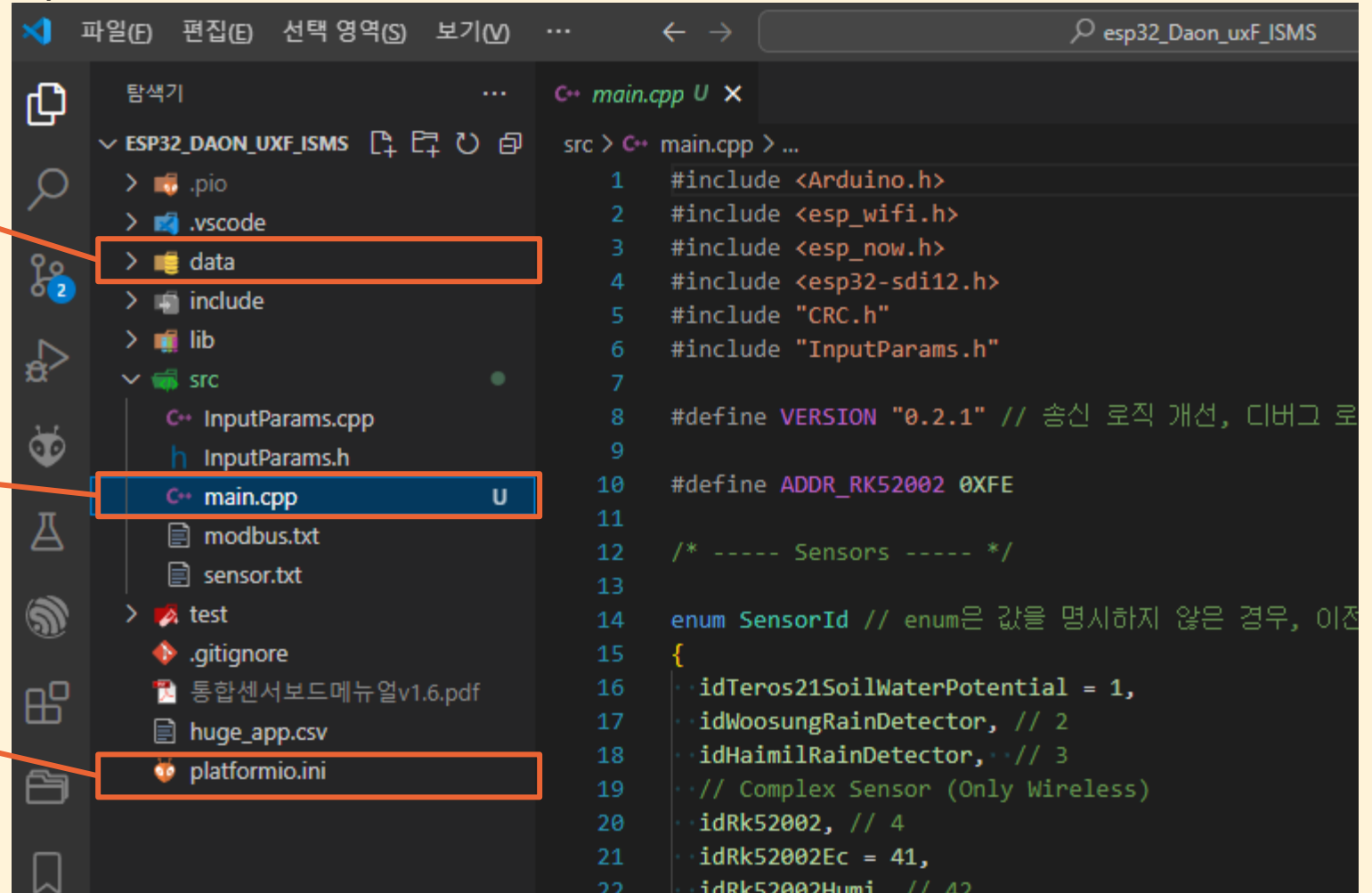
- 제어 정보 입력을 위한 웹 페이지 구성 요소
- SPIFFS 활용해 ESP32 플래시 메모리로 저장될 파일

❖ src/Main.cpp

- 펌웨어 소스코드
- 기능 구현

❖ Platformio.ini

- 개발 환경 구성 파일 (필수)
- 라이브러리 관리
- COM포트 지정



VS Code + PlatformIO



▶ 4. Platformio.ini 설정 파일로 라이브러리 자동 설치

- esp32_Daon_uXF_ISMS 프로젝트 open 시 **자동**으로 라이브러리 설치 및 개발 환경 구성
- Arduino IDE의 경우 일일이 선택/다운로드 작업이 요구된다.

▶ 5. Platformio.ini 에서 COM 포트 지정

- Upload_port & monitor_port 두 항목을 ESP32가 연결된 COM 포트로 지정한다.

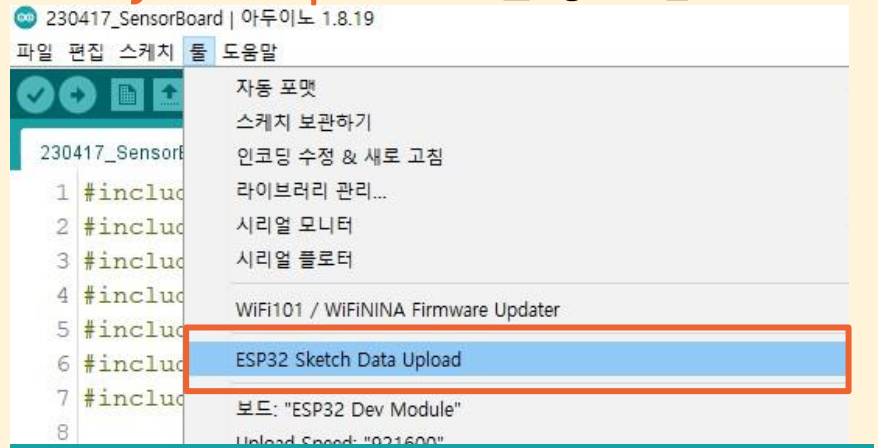
```
10
11 [env:esp32dev]
12 platform = espressif32@5.1.0 ; esp-idf 4.4.1 = esp32 arduino 2.0.4
13 board = esp32dev
14 framework = arduino
15 monitor_speed = 115200
16
17 upload_port = COM12
18 monitor_port = COM12
19
20 board_build.partitions = huge_app.csv
21 lib_deps =
22   ESP Async WebServer
23   robtillaart/CRC@^1.0.2
24   https://github.com/HarveyBates/ESP32-SDI12.git
25   plerup/EspSoftwareSerial@^8.1.0 ; sdi12 통신을 위한 라이브러리
26
```

보드 매뉴얼

보드 매뉴얼 - 웹 구성 요소 업로드

▶ 통합센서보드 초기화 설정

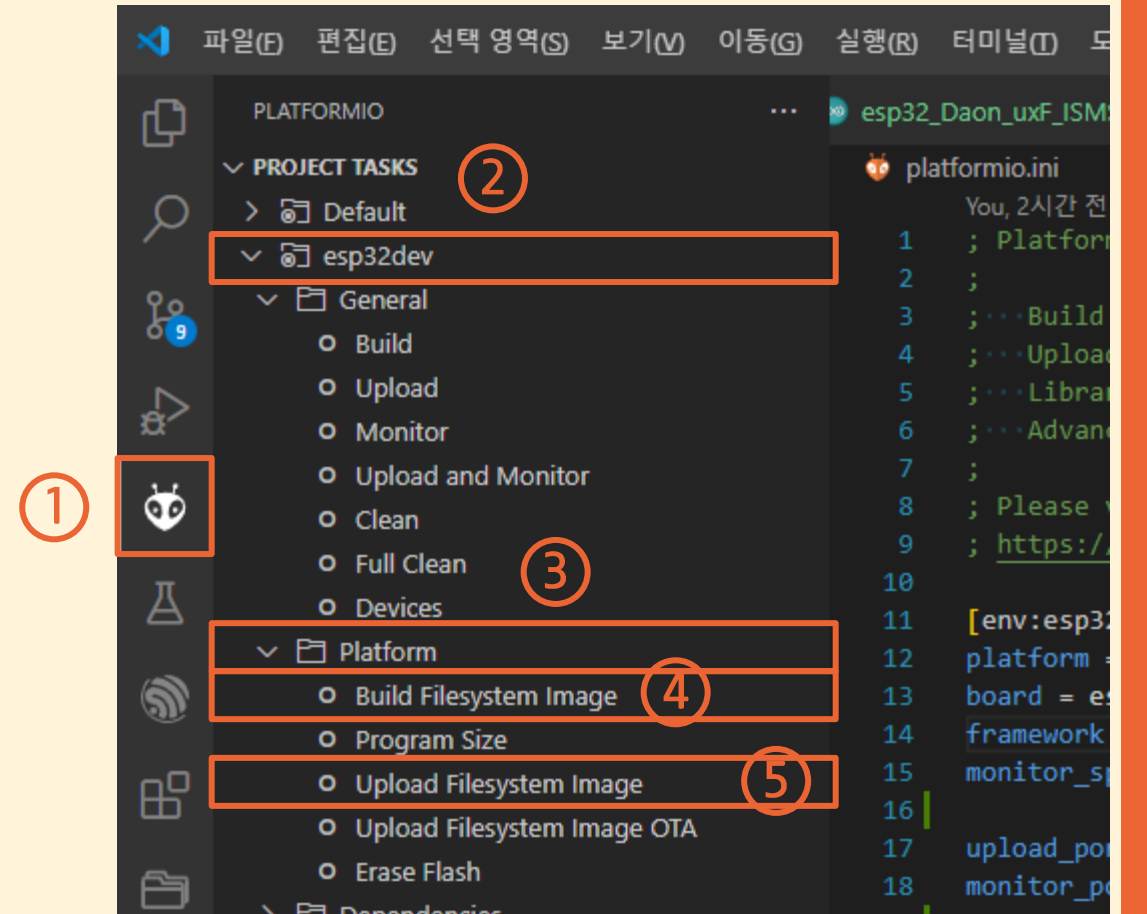
- 1. 프로젝트 내 data 폴더를 SPIFFS로 업로드한다.
 - Arduino IDE: 상단 메뉴) 툴 -> **ESP32 Filesystem Uploader** 를 통해 업로드



```
SPIFFS Image Uploaded
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash will be erased from 0x00290000 to 0x00290000...
Compressed 1507328 bytes to 4629...
Writing at 0x00290000... (100 %)
Wrote 1507328 bytes (4629 compressed) at 0x00290000...
Hash of data verified.

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

- PlatformIO: 좌측 PlatformIO 메뉴 -> esp32dev/Platform/ -> 파일시스템 이미지 **빌드** -> 이미지 **업로드**



① PlatformIO icon

② PROJECT TASKS

③ Build Filesystem Image

④ Upload Filesystem Image

⑤ Upload Filesystem Image OTA

```
Writing at 0x0039cc13... (100 %)
Wrote 983040 bytes (66526 compressed) at 0x00310000 in 3.8 seconds (effective 2087.2 kbit/s)...
Hash of data verified.

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

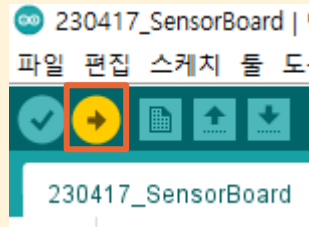
===== [SUCCESS] Took 8.02 seconds =====
* 터미널이 작업에서 다시 사용됩니다. 닫으려면 아무 키나 누르세요.
```

보드 매뉴얼



▶ 통합센서보드 초기화 설정

- 2. 프로젝트 소스코드를 업로드한다.

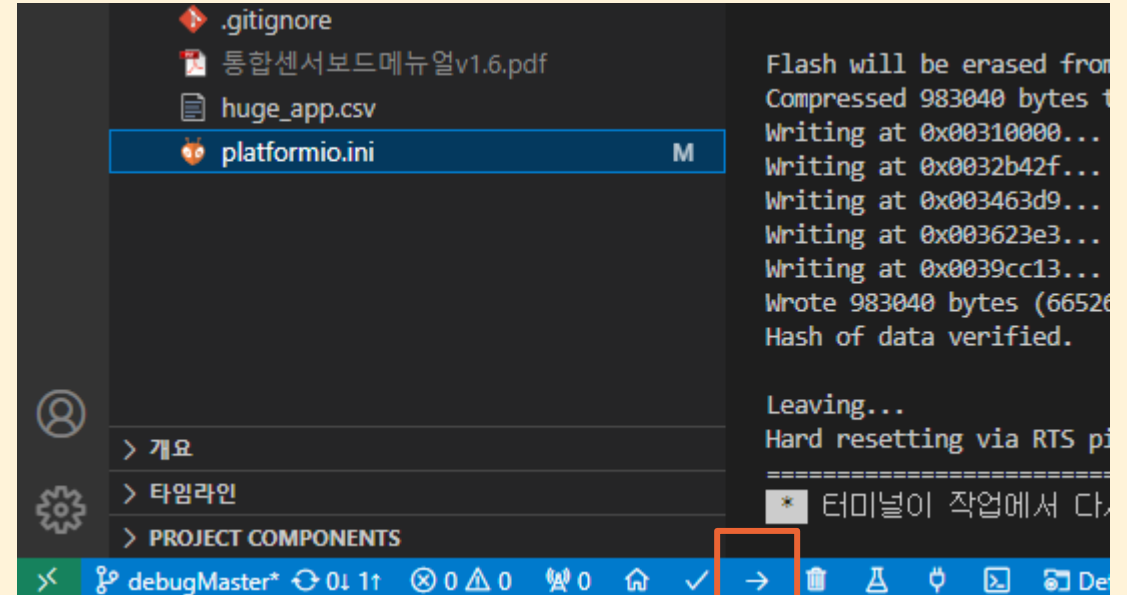


업로드 완료.

```
Writing at 0x000b1b24... (84 %)
Writing at 0x000ba0bf... (87 %)
Writing at 0x000bf196... (90 %)
Writing at 0x000c7193... (93 %)
Writing at 0x000ccb1e... (96 %)
Writing at 0x000dleff... (100 %)
Wrote 815280 bytes (522227 compressed) at 0x000
Hash of data verified.

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

- Arduino IDE



```
Writing at 0x0010b524... (95 %)
Writing at 0x00110d72... (97 %)
Writing at 0x00116027... (100 %)
Wrote 1095360 bytes (654350 compressed) at 0x00010000 in 15.0 seconds (effective 585.0 kbit/s)...
Hash of data verified.

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

===== [SUCCESS] Took 21.28 seconds
* 터미널이 작업에서 다시 사용될 것입니다. 닫으려면 아무 키나 누르세요.
```

- PlatformIO: 하단 상태 바 업로드 아이콘 사용

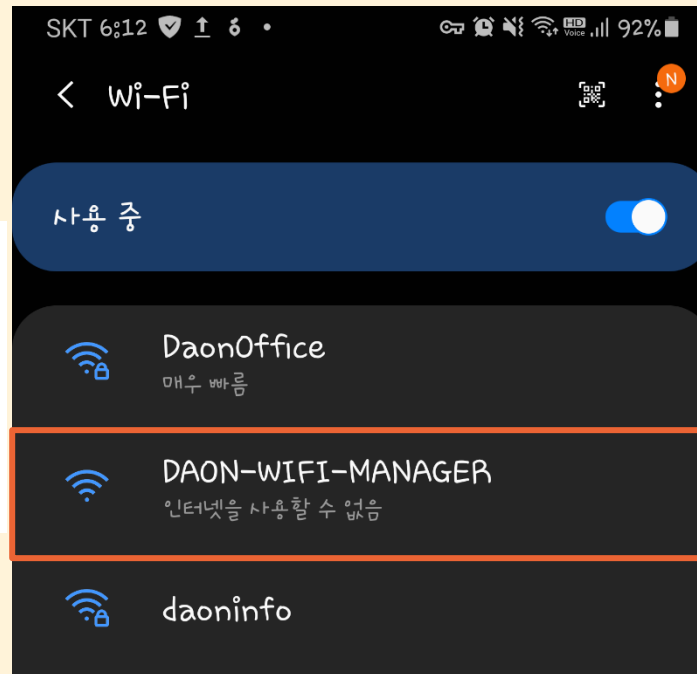
보드 매뉴얼

▶ 통합센서보드 초기화 설정

- 3. 보드가 AP 모드로 부팅되고 **DAON-WIFI-MANAGER**라는 WIFI AP가 생성된다.
해당 와이파이에 연결하고 시리얼 모니터에 출력되는 IP(192.168.4.1)로 접속한다.

```
Sensor Board ver0.2.1  
Board Mac Address: A0:B7:65:4B:58:18  
SPIFFS mounted successfully  
  
hasEssentialParams : false  
Start AP mode for board setting.  
Connect the WiFi called DAON-WIFI-MANAGER and connect to the following IP.  
Setting AP (Access Point)  
AP IP address: 192.168.4.1
```

- 시리얼 모니터



- WIFI 목록

- 192.168.4.1 접속 시

보드 매뉴얼



▶ 통합센서보드 초기화 설정

- 4. 값 설정 후 Save 버튼을 누르면 자동으로 보드가 재부팅된다.

Smart Farm Configuration

Device Settings

Protocol
Wired

Modbus Address

Sensor Type
Teros21 Soil Water Potential

Save

유선모드 UI

Smart Farm Configuration

Device Settings

Protocol
Wireless

RS485/Sensing Mode
RS485 (Standardization)

Modbus Address

Sensor Type
Teros21 Soil Water Potential

MAC Address
AA:BB:CC:DD:EE:FF

Save

무선모드 - RS485
표준화 보드 UI

Smart Farm Configuration

Device Settings

Protocol
Wireless

RS485/Sensing Mode
Sensing (Measurement)

Sensor Type
Teros21 Soil Water Potential

Measurement Period (sec)

MAC Address
AA:BB:CC:DD:EE:FF

Additional Device ? NO

Save

무선모드 - 센서보드 UI

보드 매뉴얼



▶ 통합센서보드 초기화 설정

• 4-1. 웹 UI 설명

Smart Farm Configuration

Device Settings

① Protocol
Wireless

② RS485/Sensing Mode
RS485 (Standardization)

③ Modbus Address

Sensor Type
Teros21 Soil Water Potential

MAC Address
AA:BB:CC:DD:EE:FF

Save

1. Protocol: 프로토콜
 - 통신 프로토콜 유선/무선 방식 선택
2. RS485/Sensing Mode
 - 무선통신 모드에서만 활성화
 - RS485 표준화 수집 보드로 작동할 것인지, 센서보드로 작동할 것인지 선택
3. Modbus Address
 - 2. RS485 선택 시 활성화
 - 표준화 수집 보드의 Modbus Slave ID를 지정한다.
 - USB to RS485 모듈을 사용해 표준화 테이블/정보 확인 가능

보드 매뉴얼



▶ 통합센서보드 초기화 설정

• 4-1. 웹 UI 설명

Smart Farm Configuration

Device Settings

Protocol
Wireless

RS485/Sensing Mode
Sensing (Measurement)

Sensor Type
Teros21 Soil Water Potential

Measurement Period (sec)

MAC Address
AA:BB:CC:DD:EE:FF

Additional Device ? NO

Save

4. Sensor Type

- 보드와 연결되는 센서 선택
- 무선 + RS485 표준화 보드의 경우 최대 2개 연결 가능

5. Measurement Period: 측정 주기 (초 단위)

6. MAC Address

- 무선통신 모드에서만 활성화
- 무선 통신에 사용할 MAC 주소 임의 지정
- Ex) AA:AA:AA:AA:AA:00
- 반드시 표준화 보드와 센서 보드에 동일한 주소를 기입하여야 한다.

7. Additional Device: 추가 센서보드임을 표시

- 추가 기능 항목에서 설명



▶ 통합센서보드 초기화 설정

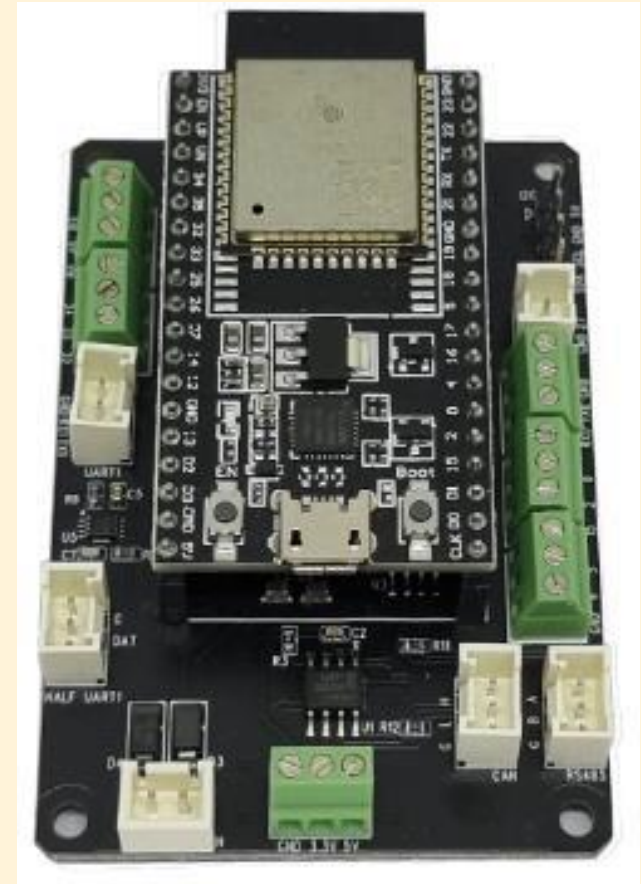
- 5. 본 문서 보드 매뉴얼의 **센서 하드웨어 연결** 자료를 참조하여 회로를 구성한다.

보드 매뉴얼



▶ 센서 하드웨어 연결

- TEROS-21 수분장력센서
 - GPIO5 연결
 - 실드 터미널의 5V 전원 공급
- RK520-02 토양복합센서/EC센서
 - 무선통신 전용 센서
 - RS485 포트에 연결
 - 12V 외부 전원 공급(단자대 사용)



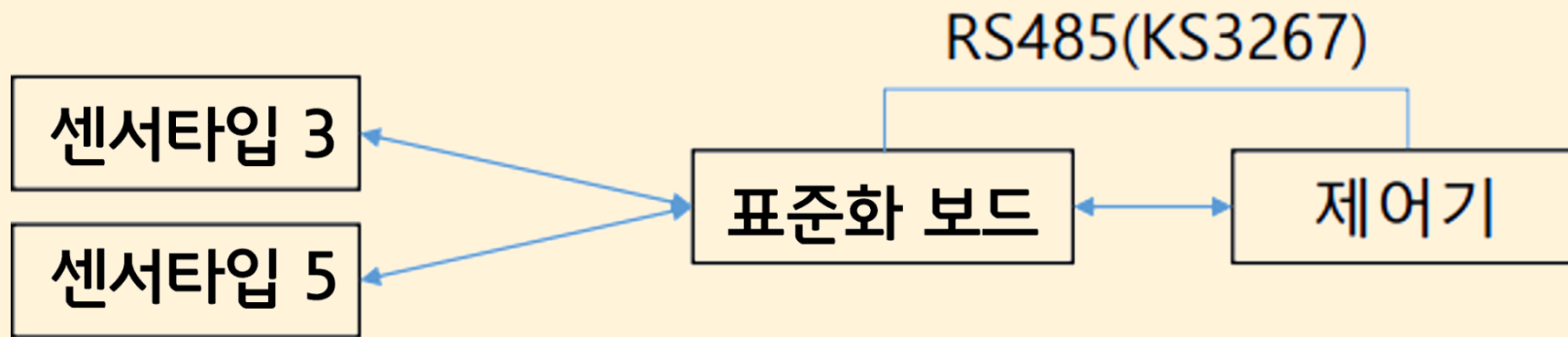
HiGenis CAN485 실드 + ESP32 보드

보드 매뉴얼



▶ 센서 하드웨어 연결

- 무선통신 노드 구성
 - 무선 + 표준화 보드에는 **최대 2개**의 센서 보드를 연결할 수 있다.



오류 감지



▶ 무선 통신 센서 보드 Disconnect 오류 감지

- 1번 이상의 센서 데이터를 받은 후, 30분간 센서 데이터가 전송되지 않으면 센서 데이터 레지스터 필드를 초기화하고 센서 상태를 Error(0x01) 로 변경한다.
- 이후 센서 데이터 재수신까지 30분마다 오류 메시지가 출력이 반복된다.

400242	EC	0	0000	0.00
400243		0	0000	0.00
400244	EC 센서상태	0	0000	0.00
400245	광양자	0	0000	0.00
400246		0	0000	0.00
400247	광양자 센서상태	0	0000	0.00
400248	토양함수율	0	0000	0.00
400249		0	0000	0.00
400250	토양함수율 센서상태	0	0000	0.00
400251	토양수분장력	0	0000	0.00
400252		0	0000	0.00
400253	토양수분장력 센서상태	0	0000	0.00
400254	PH	0	0000	0.00
400255		0	0000	0.00
400256	PH 센서상태	0	0000	0.00
400257	지온	13107	3333	24.90
400258		16839	41C7	0.00
400259	지온 센서상태	0	0000	0.00



400242	EC	0	0000	0.00
400243		0	0000	0.00
400244	EC 센서상태	1	0001	0.00
400245	광양자	0	0000	0.00
400246		0	0000	0.00
400247	광양자 센서상태	0	0000	0.00
400248	토양함수율	0	0000	0.00
400249		0	0000	0.00
400250	토양함수율 센서상태	1	0001	0.00
400251	토양수분장력	0	0000	0.00
400252		0	0000	0.00
400253	토양수분장력 센서상태	0	0000	0.00
400254	PH	0	0000	0.00
400255		0	0000	0.00
400256	PH 센서상태	0	0000	0.00
400257	지온	0	0000	0.00
400258		0	0000	0.00
400259	지온 센서상태	1	0001	0.00

Data

Status

추가 기능



▶ 무선 통신 추가 센서

- 개요 : 무선통신 환경에서 동일한 센서를 하나의 무선 표준화 보드에 2개 연결하는 기능
- 현재 지원 센서 : Teros21 수분장력센서, RK520-02 토양복합센서
- 사용 방법 : 추가로 연결하는 센서보드에 Additional device 옵션을 Checked(ADD)로 설정한다.
 - 기존 센서는 203~292에 저장되며, 추가 센서는 303~392에 저장한다. (Address + 100)
 - Ex. 기존 EC 센서 값이 Address 242에 저장되는데, 추가 EC 센서는 242+100 = 342에 저장.

MAC Address

AA:BB:CC:DD:EE:FF

Additional Device ?

ADD

Save



▶ 무선 통신 배터리 전압 측정

- 개요 : 무선통신 시 각 센서보드에 연결되는 배터리 잔량을 확인할 수 있게 한다.
- 상세 내용 : 각각의 무선 센서보드에서 GPIO32 핀을 ADC로 사용하여 SZH-SSBH-043모듈이 전송하는 ADC 신호로 배터리 전압 값 계산 및 전송, 무선 표준화 보드가 레지스터맵 293~300번에 저장한다.
- 배터리 전압 값을 저장하는 위치는 293이며 uint16_t형식으로 저장한다.
(추가 센서 보드는 393 이다.)

▶ RK520-02 관련 이슈

- HiGenis ESP32 실드 보드에 Micro 5 pin USB 전원과 12V 전원을 **동시에 공급**하면 노이즈가 발생한다.
- USB 연결 제거하고 **12V 전원만** 공급하면 정상적으로 동작한다.
- v0.2 패치로 노이즈 필터링 로직 추가로 개선됨 / 안정적인 외부 12V 사용 권장

```
15:36:00.392 -> 4
15:36:00.492 -> Rk520-02 response: 00 FE 03 06 00 F6 00 00 00 00 AC 94
15:36:00.492 -> Sensor Response CRC error
15:36:00.492 -> Last Packet Send Status:      Delivery Success.
15:36:00.492 -> Last Packet Send Status:      Delivery Success.
15:36:00.545 -> Last Packet Send Status:      Delivery Success.
```

☒ 자동 스크롤 ☒ 타임스탬프 표시

새 줄

115200 보드레이트

출력 지우기

▶ 단종 감우센서 연결 상세(우성, 해밀기연)

- 감우 센서의 출력 선 2개(백색, 녹색)를 각각 24V, EL817 모듈의 Input에 연결한다.
(ex. 백색 24v, 녹색 input 또는 백색 input, 녹색 24v. 두 가지 방식 동일하게 동작)
- SMPS, 센서, EL817 모듈에 공통 GND를 연결한다.
- ESP32의 3.3V 를 EN817 모듈의 Vcc에 연결.
- ESP32와 EN817 GND 연결.
- ESP32의 GPIO4번 핀에 EN817 output 연결.
- 비가 감지되지 않으면 출력이 3.3V,
비가 감지된 경우 모듈 출력이 0V로 떨어진다.
(= Digital Pin 이 LOW인 경우 강우)

