

Chapter 02

아두이노 인공지능 드론 준비하기

이번 장에서는 아두이노 인공지능 드론을 조립하고, 아두이노 개발 환경을 구성합니다. 다음과 같이 인공지능 드론을 조립합니다.



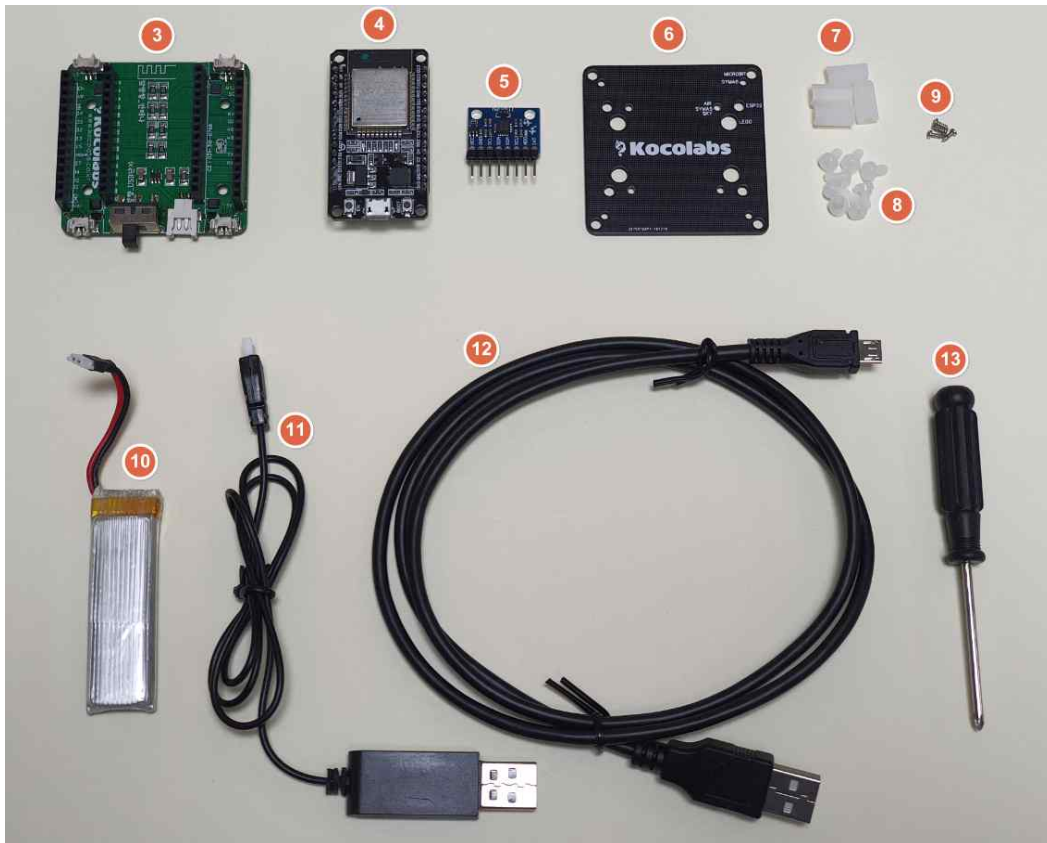
01 아두이노 인공지능 드론 조립하기

여기서는 아두이노 인공지능 드론 부품을 살펴보고 조립합니다.

01 부품 살펴보기

이 책에서 사용될 아두이노 인공지능 드론 부품 구성은 다음과 같습니다.





번호	이름	수량
1	날개 지지대	4개
2	드론 플라스틱 몸체	1개
3	드론 쉴드	1개
4	ESP32 아두이노	1개
5	MPU6050 센서	1개
6	드론 쉴드 고정판	1개
7	플라스틱 지지대	4개+ @1~2개
8	플라스틱 볼트	8개+ @1~2개
9	1.4x4 볼트	8개+ @1~2개
10	3.7V/500mA 배터리	1개
11	배터리 충전기	1개
12	USB A 미니 B 케이블	1개
13	드라이버	1개

02 인공지능 드론 조립하기

다음과 같은 순서로 아두이노 인공지능 드론을 조립해 봅니다.

1. 드론 쉴드 고정판 조립하기
2. 드론 쉴드 고정판과 드론 플라스틱 몸체 결합하기
3. 드론 쉴드 장착하기
4. MPU6050 센서 장착하기

5. ESP32 아두이노 장착하기
6. 드론 날개와 몸체 연결하기
7. 드론 몸체에 배터리 장착하기

드론 쉴드 고정판 조립하기

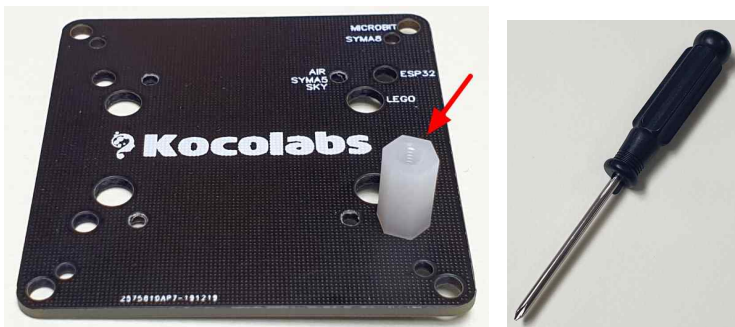
1. 다음과 같이 드론 쉴드 고정판, 플라스틱 지지대 1개, 플라스틱 볼트 1개를 준비합니다.



2. 다음과 같이 플라스틱 볼트를 드론 쉴드 고정판 아래쪽에서 위쪽으로 끼워 넣습니다.



3. 다음과 같이 플라스틱 지지대를 결합합니다. 드라이버를 이용하여 적당히 조여 줍니다.



4. 플라스틱 지지대 3개, 플라스틱 볼트 3개를 더 준비합니다.

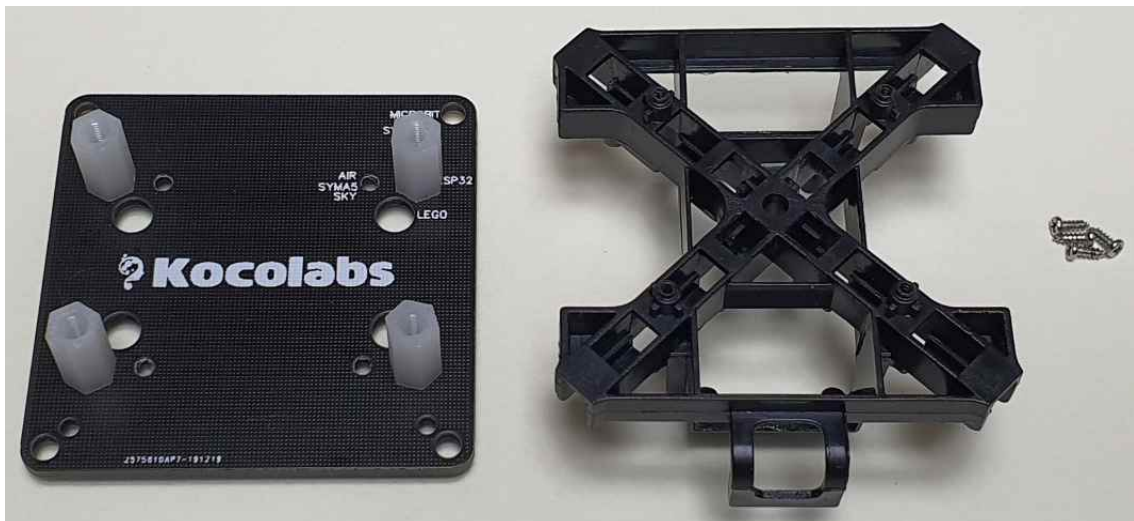


5. 같은 방식으로 나머지 플라스틱 지지대와 볼트를 결합합니다. 드라이버를 이용하여 적당히 조여 줍니다.



드론 쉴드 고정판과 드론 플라스틱 몸체 결합하기

1. 다음과 같이 드론 쉴드 고정판, 드론 플라스틱 몸체, 1.4x4mm 볼트를 준비합니다.



2. 다음과 같이 드론 쉴드 고정판을 드론 플라스틱 몸체와 맞춥니다.

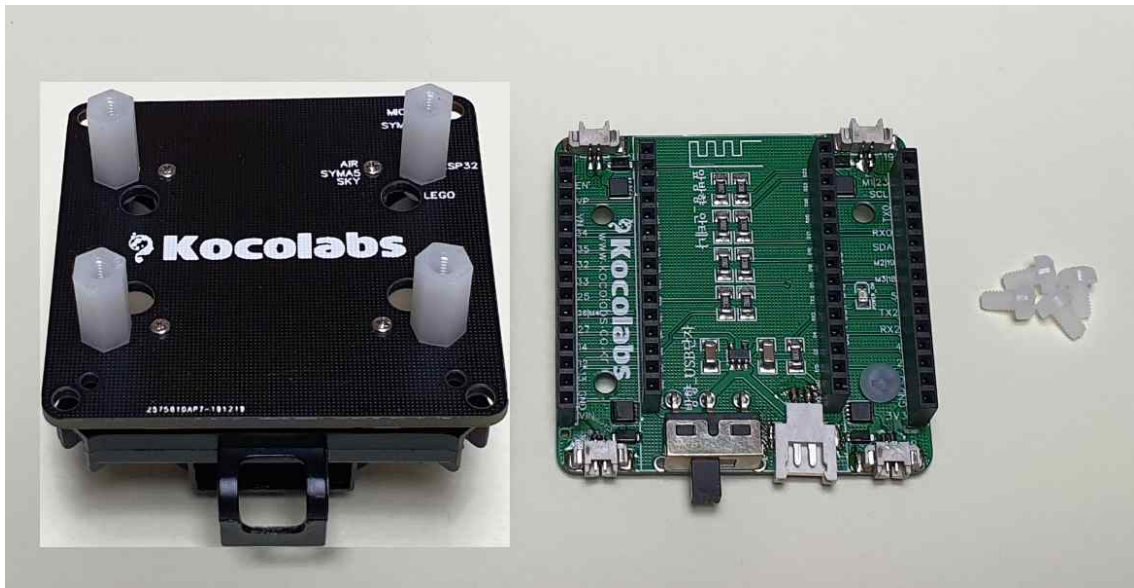


3. 드라이버를 이용하여 다음과 같이 4군데 볼트를 체결합니다. 드론 쉘드 고정판이 드론 플라스틱 몸체에서 흔들리지 않도록 적당히 단단히 조립합니다.

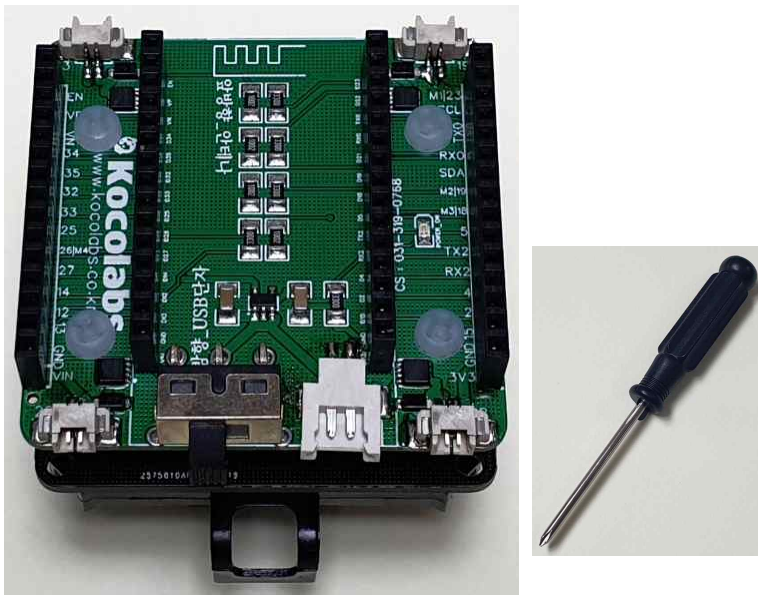


드론 쉘드 장착하기

1. 다음과 같이 드론 쉘드 고정판, 드론 쉘드, 플라스틱 볼트 4개를 준비합니다.

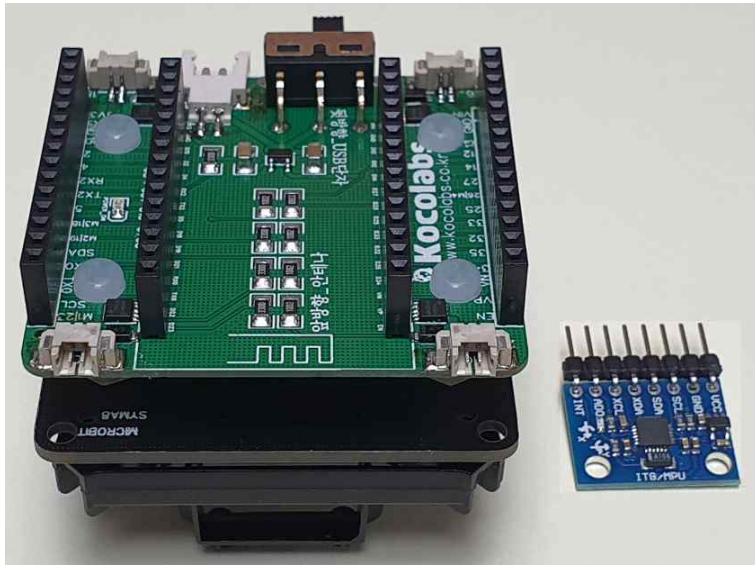


2. 다음과 같이 드론 쉴드를 드론 쉴드 고정판 위에 올린 후, 플라스틱 볼트 4개를 드라이버를 이용하여 체결합니다. 드론 쉴드가 흔들리지 않도록 적당히 조여 줍니다.

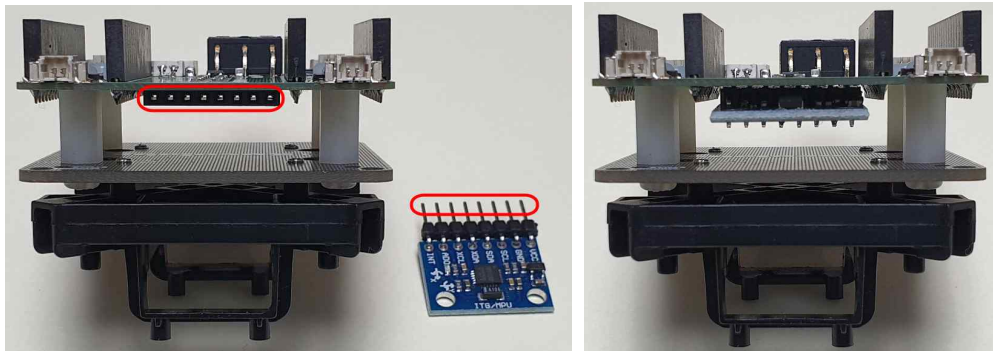


MPU6050 센서 장착하기

1. 다음과 같이 드론 쉴드, MPU6050 센서를 준비합니다.

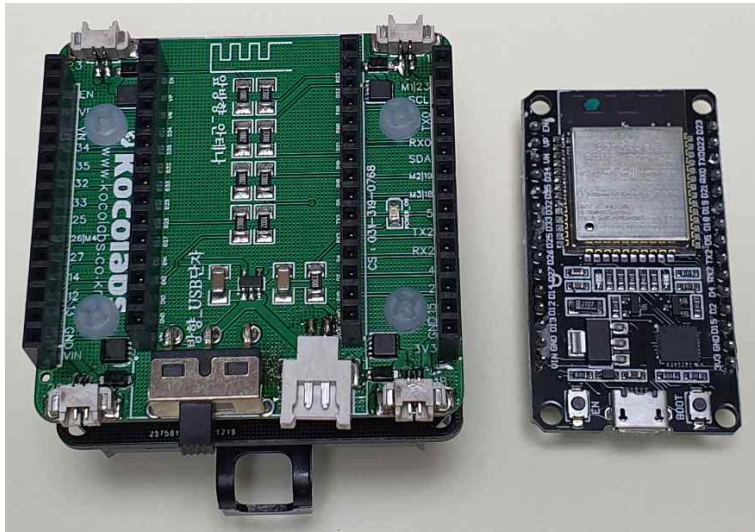


2. 다음과 같이 드론 쉴드에 MPU6050 센서를 장착합니다.



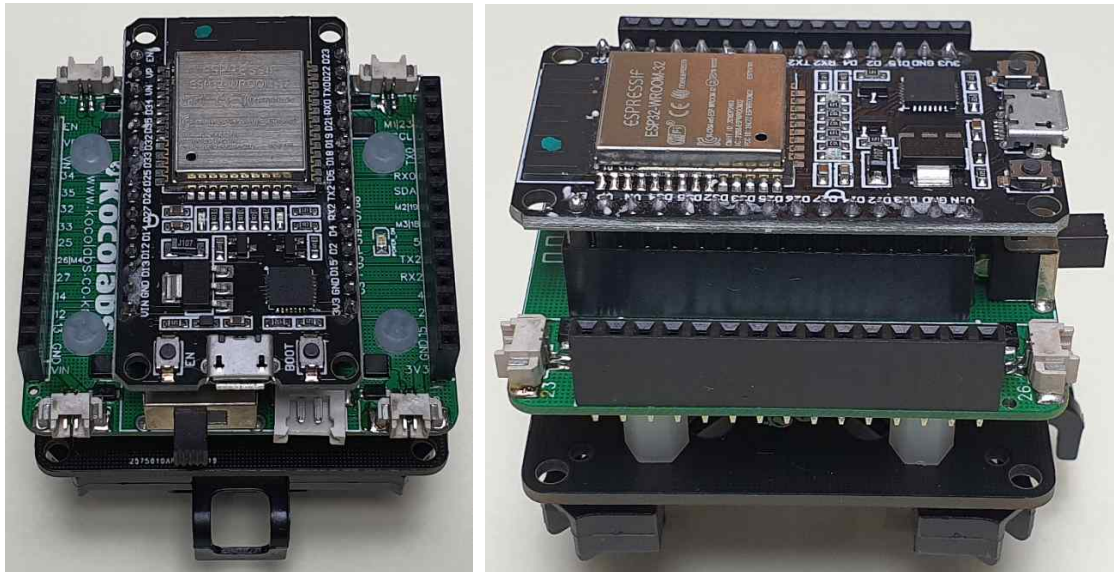
ESP32 아두이노 장착하기

1. 다음과 같이 드론 쉴드, ESP32 아두이노를 준비합니다.



*** 방향에 주의합니다. 반대로 장착할 경우 모터가 최대 속도로 돌게 되어 다칠수 있습니다.

2. 다음과 같이 드론 쉴드에 ESP32 아두이노를 장착합니다.

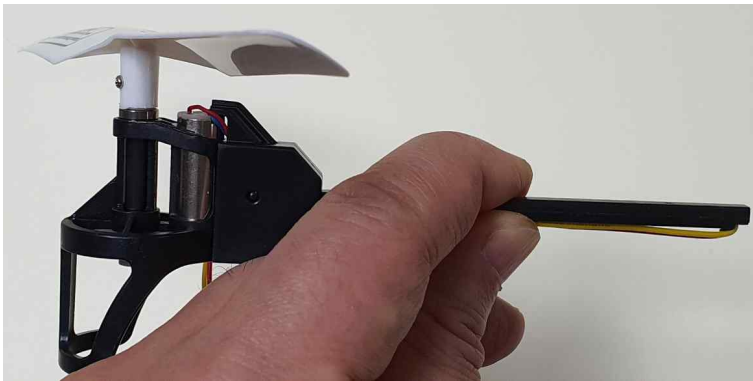


드론 날개와 몸체 연결하기

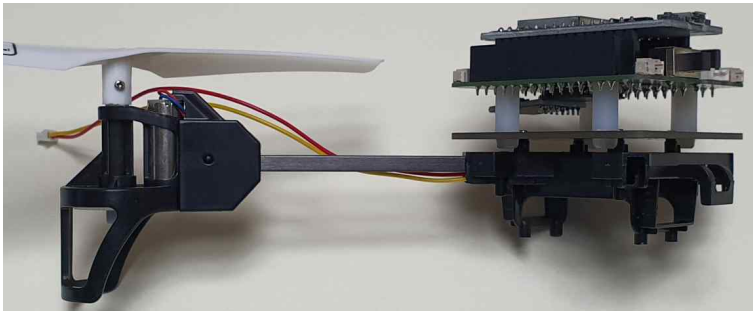
1. 다음과 같이 날개 지지대와 조립된 드론 쉴드 고정판을 준비합니다. 화살표 부분에 A2라고 쓰인 흰색 날개 지지대를 준비합니다.



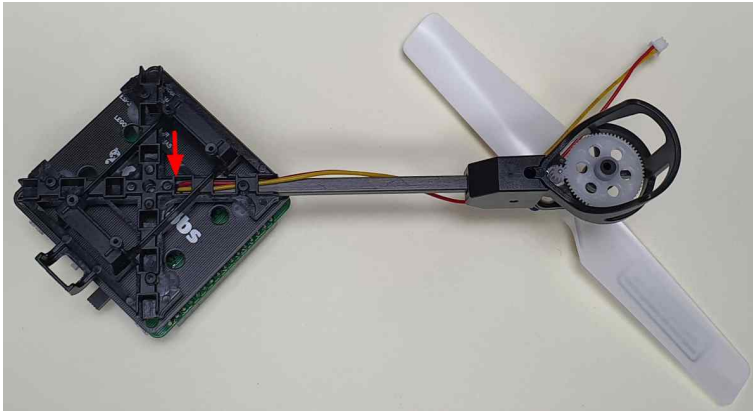
2. 다음과 같이 전선을 잡습니다.



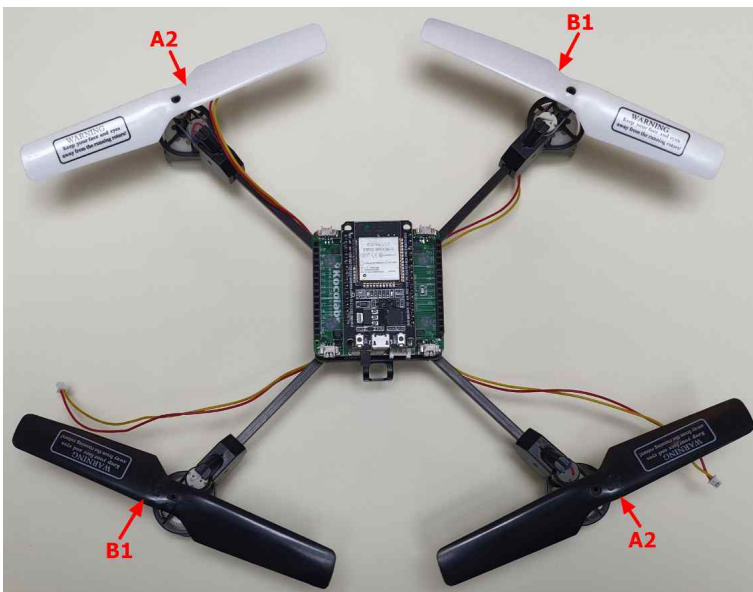
3. 다음과 같이 흰색 A2 날개 지지대를 드론 플라스틱 몸체에 끼워 넣습니다.



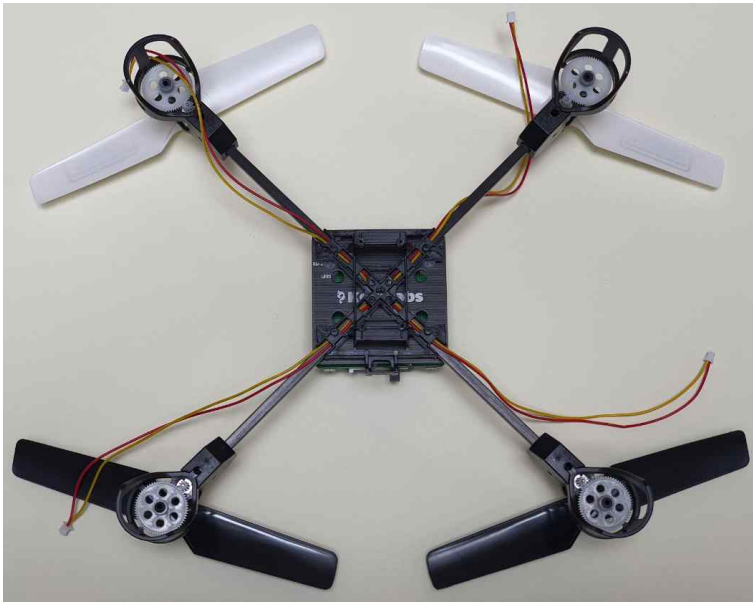
4. 화살표 표시 부분까지 밀어 넣습니다.



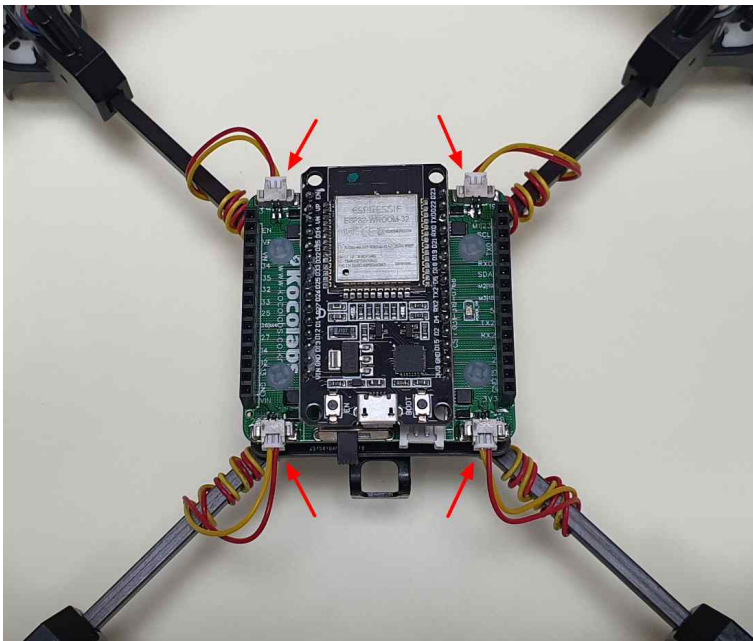
5. 나머지 날개도 다음과 같이 조립합니다.



6. 다음은 아래에서 본 모습입니다.

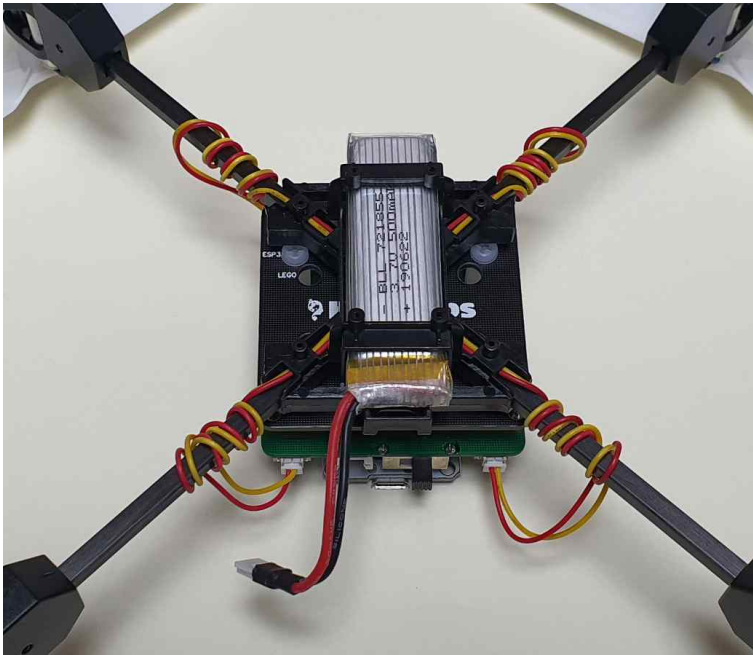


7. 다음과 같이 전선을 지지대에 4~5회 감아준 후, 커넥터를 드론 쉴드 고정판에 연결해 줍니다.

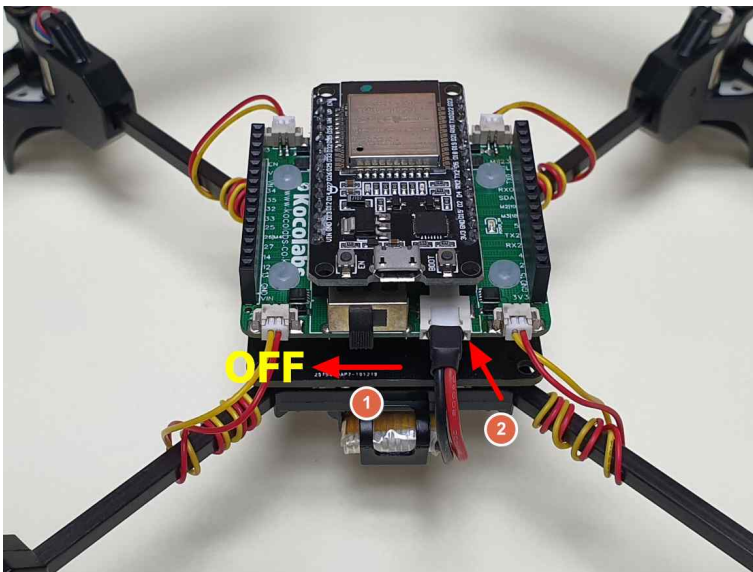


드론 몸체에 배터리 장착하기

1. 다음과 같이 배터리를 드론 플라스틱 몸체에 끼워 넣습니다.



2. 다음과 같이 ❶ 모터 전원을 OFF 상태로 두고, ❷ 배터리 커넥터를 연결합니다.



❶ 모터 전원은 모터를 구동할 때만 ON 상태로 두고 구동합니다. OFF 상태시 모터가 구동되지 않습니다.

3. 배터리 충전 시에는 다음과 같이 배터리 충전기에 배터리 커넥터를 연결한 후, 충전기를 USB 단자에 연결합니다. 완충시간은 약 70분입니다. 배터리 사용시간은 드론을 연속적으로 날릴 경우 5분 전후입니다.



다음은 조립이 완료된 모양입니다.



이상 아두이노 인공지능 드론 조립을 마칩니다.

02 아두이노 인공지능 드론 살펴보기

여기서는 아두이노 인공지능 실습을 위한 하드웨어 환경을 살펴봅니다.

01 ESP32 아두이노 살펴보기

우리가 사용할 비행 제어기(Flight Controller)는 [DOIT ESP32 DEVKIT V1]으로 다음과 같습니다. [DOIT ESP32 DEVKIT V1]은 ESP32 칩을 내장한 ESP-WROOM-32 모듈 기반으로 구성된 개발용 보드입니다. [DOIT ESP32 DEVKIT V1]은 4MB 또는 16MB 크기의 플래시 메모리와 160 또는 240MHz의 클럭, 50KB 정도의 사용 가능한 RAM(실제 크기는 520KB), Wifi 모듈과 Bluetooth 모듈을 내장하고 있습니다. [DOIT ESP32 DEVKIT V1]은 아두이노 소프트웨어 기반의 개발이 가능합니다.

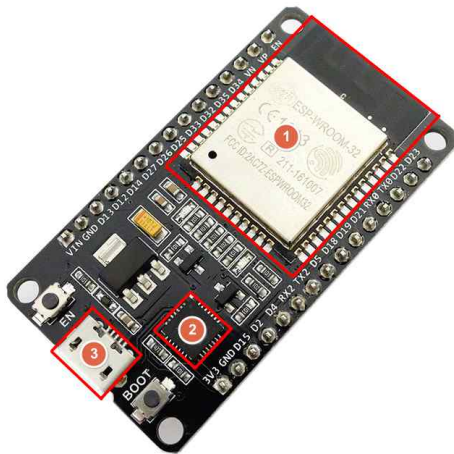


다음은 [DOIT ESP32 DEVKIT V1]의 사양입니다.

CPU : Xtensa 듀얼 코어 32 비트 LX6, 240MHz, 600DMIPS
SRAM: 520 KB
Flash Memory: 4 MB
Wi-Fi: IEEE 802.11 b/g/n
Bluetooth : v4.2 BR/EDR, BLE
LED PWM : 16 채널
ADC : 12-bit, 18 채널
DAC : 8-bit x 2개
터치 센서 : 10개
UART : 3개
I2C : 2개
SPI : 4개
I2S : 2개

동작 전압: 3.3V
입력 전압: 7-12V

[DOIT ESP32 DEVKIT V1] 모듈은 와이파이, 블루투스 인터페이스와 시리얼 인터페이스를 가지고 있습니다. 다음 그림은 [DOIT ESP32 DEVKIT V1]의 주요 부분을 나타내고 있습니다.



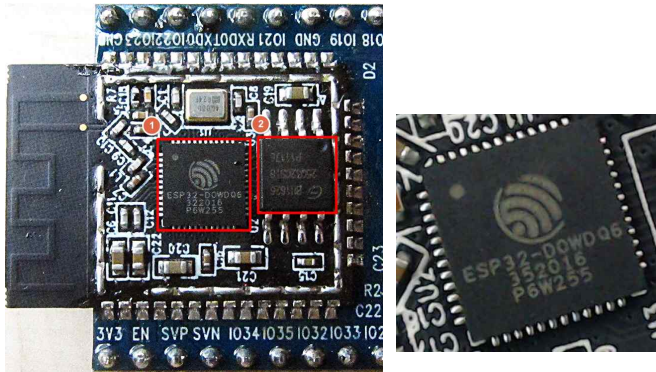
❶ 부분은 와이파이, 블루투스 기능을 포함한 ESP-WROOM-32 모듈을 붙인 부분입니다.



❷ 부분은 시리얼 영역으로 SILICON LABS 사에서 만든 CP2102 칩입니다. CP2102 칩은 USB-to-UART 브리지 모듈로 ❸ 부분의 USB 단자와 ❶ 부분에 포함된 UART를 연결해주는 역할을 합니다.

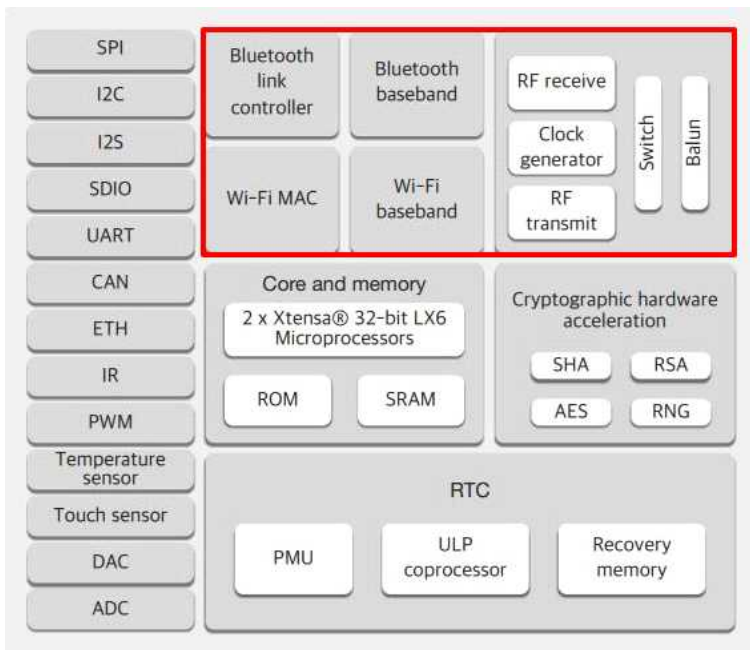
ESP-WROOM-32 모듈 살펴보기

ESP-WROOM-32 모듈의 금속 막을 벗기면 내부 모양은 다음과 같습니다. ❶ 부분은 ESP32 칩이고, ❷ 부분은 SPI 통신 방식을 사용하는 플래시 메모리입니다.



❶ ESP32 칩 내부에는 MCU, RAM, Wifi, Bluetooth, IO 등이 내장되어 있습니다. 그러나 프로그래밍 가능한 ROM은 없습니다. 그래서 사용자 프로그램을 저장하기 위해 ESP32 칩 외부에 SPI 플래시 메모리를 장착해야 합니다. ❷ 부분은 4MB 크기의 SPI 플래시 메모리입니다.

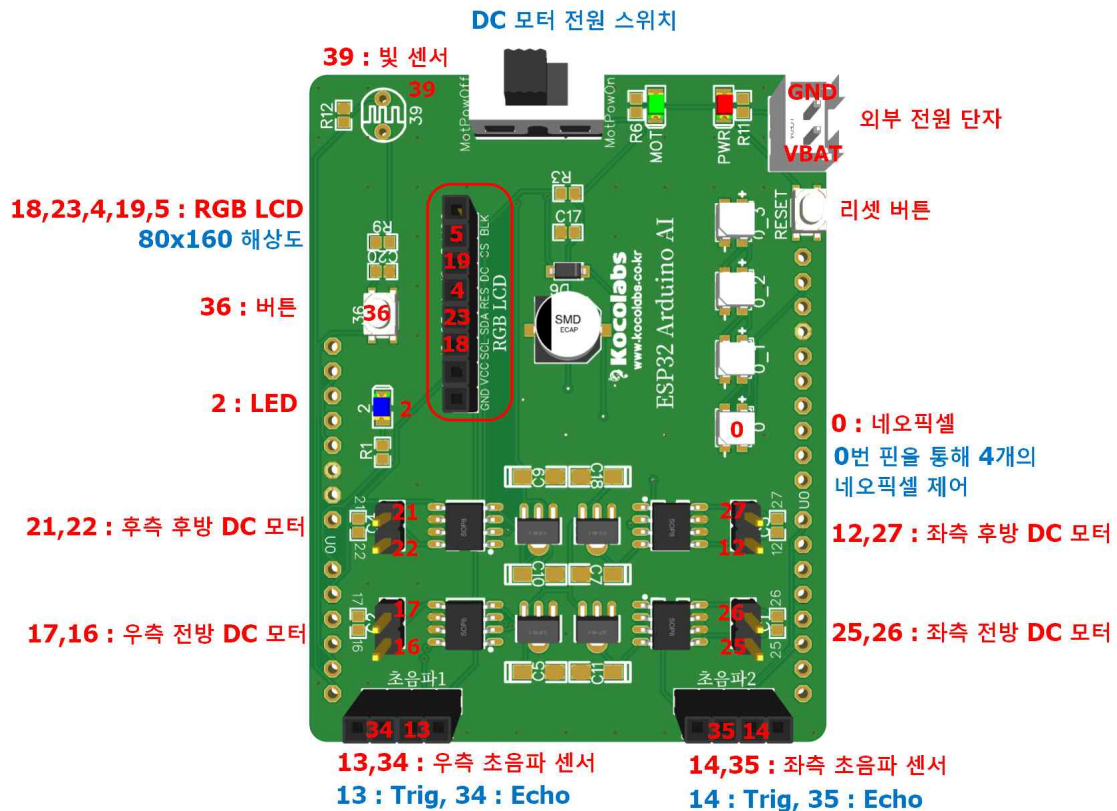
다음은 ESP32 칩의 내부 구조를 나타냅니다.



빨간색 사각 박스 부분은 Wifi, Bluetooth 영역이고 나머지 부분은 MCU 영역입니다. MCU 영역은 Xtensa 32 비트 LX6 마이크로 프로세서 2개로 구성됩니다.

02 아두이노 인공지능 쉘드 핀 살펴보기

다음은 아두이노 인공지능 드론 쉘드입니다. 아두이노는 이 쉘드를 통해서 아두이노 인공지능 드론의 DC 모터 제어, LED 제어, 네오픽셀 제어, 그래픽 LCD 제어, 가속도 자이로 센서 입력을 받게 됩니다. 아두이노 쉘드에는 각 부품이 연결된 아두이노 핀을 표시하고 있습니다. 예를 들어 네오픽셀은 아두이노 0번 핀에 연결되어 있고, 빛 센서는 39 핀에 연결되어 있습니다.



03 아두이노 개발 환경 구성하기

이제 아두이노 스케치를 구현하고, 컴파일하고, 업로드하기 위한 개발 환경을 구성하도록 합니다. 먼저 아두이노 개발을 위한 아두이노 소프트웨어를 설치하고, 다음은 ESP32 패키지를 설치합니다.

01 아두이노 소프트웨어 설치하기

먼저 아두이노 개발을 위한 아두이노 소프트웨어를 설치합니다.

다음과 같은 아두이노 소프트웨어를 설치해 봅니다.



우리는 이 프로그램을 이용하여

- ❶ 아두이노 스케치를 작성하고,
- ❷ 작성한 스케치를 컴파일하고,
- ❸ 컴파일한 스케치를 아두이노 보드상에 업로드하고,
- ❹ 시리얼 모니터를 통해 결과를 확인하게 됩니다.

1. [www.arduino.cc] 사이트에 접속합니다.



2. 홈페이지가 열립니다. [SOFTWARE] 메뉴를 선택합니다.



3. 새로 열린 페이지에서 아래로 조금 이동하여 다음 부분을 찾습니다.



*** 2021년 10월 기준 ARDUINO 1.8.16이 사용되며, 버전은 다운로드 시점에 따라 변경될 수 있습니다. 독자 여러분은 최신 버전을 설치하도록 합니다.

4. [Windows Win 7 and newer]를 마우스 클릭합니다.



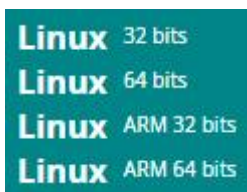
5. 그러면 다음 페이지로 연결됩니다. 하단에 있는 [JUST DOWNLOAD] 부분을 누릅니다.



*** 맥 OS 사용자의 경우엔 다음을 선택합니다.



*** 리눅스 OS 사용자의 경우엔 다음 중 하나를 선택합니다.

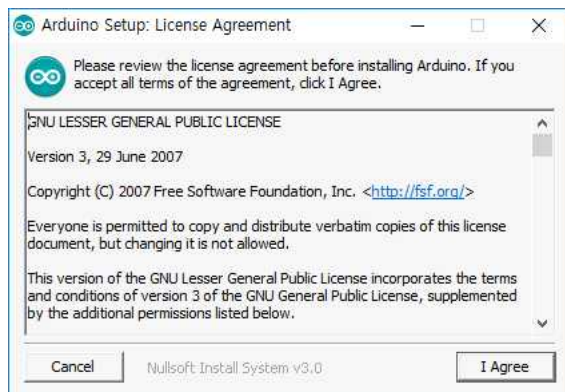


[Linux ARM 32 bits]나 [Linux ARM 64 bits]의 경우엔 라즈베리파이와 같이 ARM 기반 SOC에서 동작하는 리눅스에서 사용합니다.

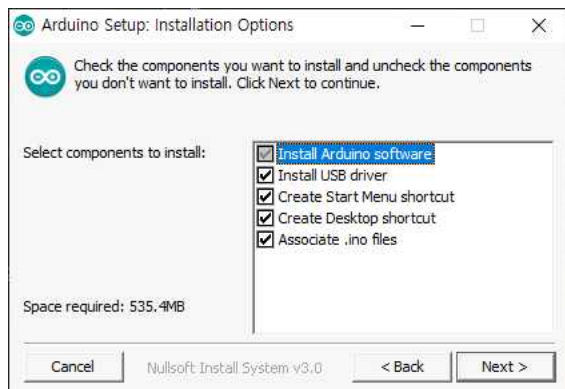
6. 다운로드가 완료되면 마우스 클릭하여 설치 프로그램을 실행시킵니다.



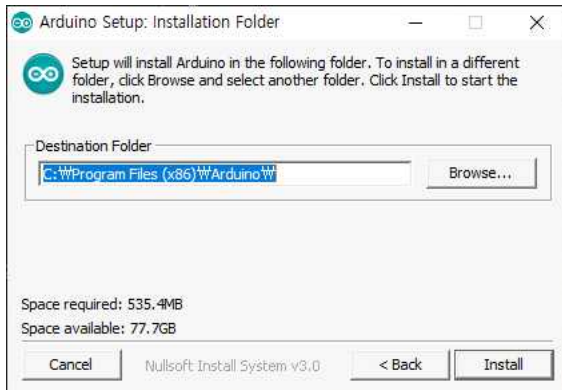
7. 다음과 같이 [Arduino Setup: License Agreement] 창이 뜹니다. 사용 조건 동의에 대한 내용입니다. [I Agree] 버튼을 눌러 동의합니다.



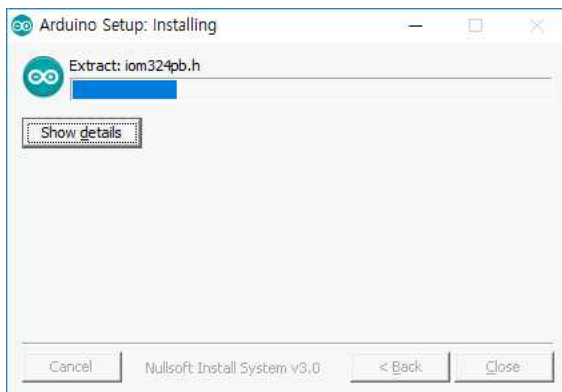
8. 다음과 같이 [Arduino Setup: Installation Options] 창이 뜹니다. 설치 선택에 대한 내용입니다. 기본 상태로 둔 채 [Next] 버튼을 누릅니다.



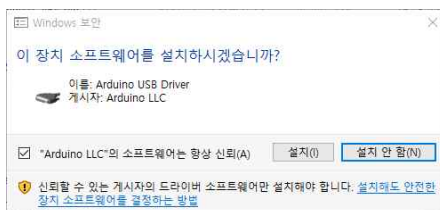
9. 다음과 같이 [Arduino Setup: Installation Folder] 창이 뜹니다. 설치 폴더 선택 창입니다. 기본 상태로 둔 채 [Install] 버튼을 누릅니다.



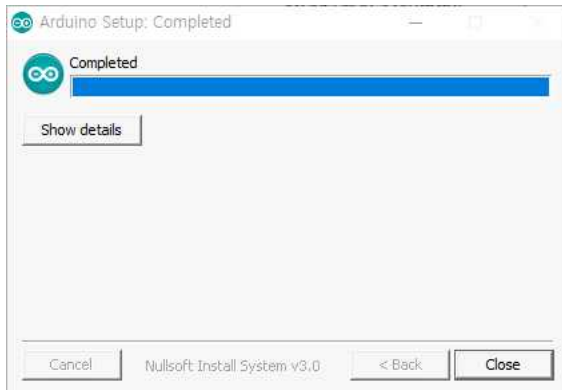
10. 그러면 다음과 같이 설치가 진행됩니다.



11. 설치 마지막 단계에 다음과 같은 창이 하나 이상 뜹니다. 아두이노 보드에 접근하기 위해 필요한 드라이버 설치 창입니다. [설치(I)] 버튼을 눌러줍니다.



12. 다음과 같이 [Arduino Setup: Completed] 창이 뜹니다. 설치 완료 창입니다. [Close] 버튼을 눌러 설치를 마칩니다.



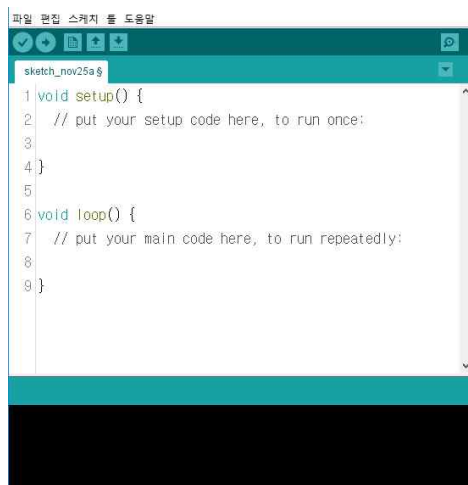
13. 바탕 화면에 다음 아이콘이 설치됩니다. 아이콘을 눌러 아두이노 소프트웨어를 실행시킵니다.



14. 처음엔 다음과 같은 보안 경고 창이 뜹니다. 아두이노 소프트웨어를 사용하기 위해 필요한 부분이기 때문에 [액세스 허용(A)] 버튼을 누릅니다.



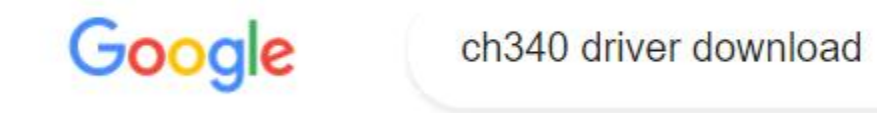
15. 그러면 다음과 같이 아두이노 소프트웨어 프로그램이 실행되는 것을 볼 수 있습니다.



02 USB 드라이버 설치하기

우리가 사용할 아두이노 보드 [DOIT ESP32 DEVKIT V1]을 USB에 연결하기 위해서는 CH340 드라이버를 설치해야 합니다.

1. 다음과 같이 [ch340 driver download]를 검색합니다.



2. 다음 사이트로 들어갑니다.

<https://sparks.gogo.co.nz> > ch340 ▾

CH340 Drivers for Windows, Mac and Linux - Gogo:Tronics

3. 다음 부분을 찾아 마우스 클릭합니다.

Windows

([Manufacturer's Chinese Info Link](#))

- Download the [Windows CH340 Driver](#)



4. 다음과 같이 드라이버 프로그램을 다운로드 받습니다.



CH34x_Install_Win....zip



5. 압축을 푼 후, 다음 파일을 이용하여 드라이버를 설치합니다.



CH34x_Install_Windows_v3_4.EXE

03 ESP32 아두이노 패키지 설치하기

여기서는 ESP32 아두이노 패키지를 설치하고, 아두이노 보드를 컴퓨터에 연결하는 방법, 아두이노 보드와 시리얼 포트를 선택하는 방법, 스케치를 작성하는 방법, 아두이노 오류 발생 시 대처 방법을 살펴봅니다.

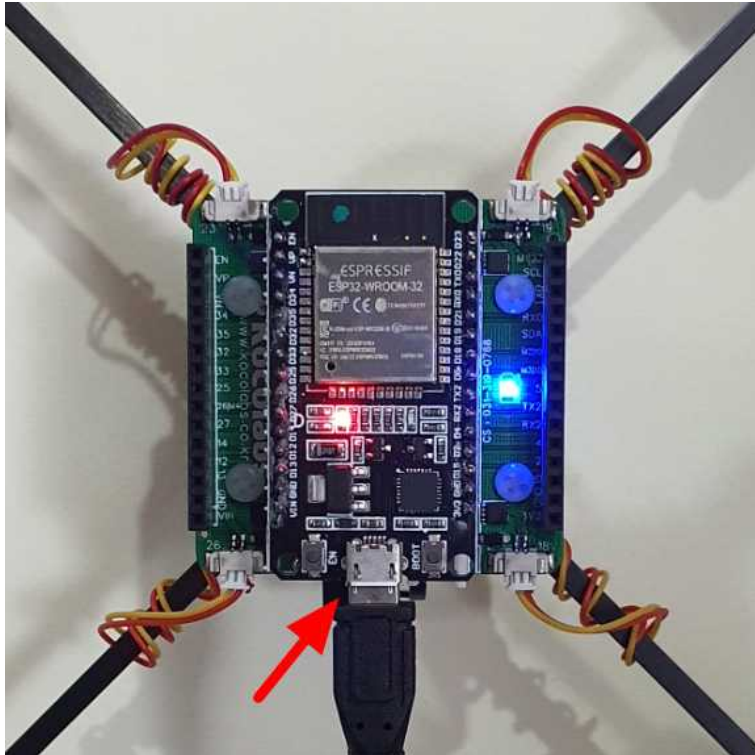
ESP32 보드 컴퓨터에 연결하기

이제 ESP32 보드와 컴퓨터를 연결해 봅니다. 아두이노 보드의 USB는 다음과 같이 세 가지 기능을 제공합니다.

- 1) 전원을 공급 받을 수 있고,
- 2) 시리얼 포트를 통해 컴파일한 프로그램을 업로드할 수 있고,
- 3) 시리얼 포트를 통해 디버깅 메시지를 볼 수 있습니다.

그래서 아두이노 보드는 USB 케이블 하나로 컴퓨터로 연결될 수 있으며, 간단한 인터페이스를 이용하여, 개발을 진행할 수 있습니다.

1. USB 케이블의 한쪽 끝(micro B 형)을 아두이노 보드에 연결합니다.



2. USB 케이블의 다른 쪽 끝을 컴퓨터에 연결합니다.



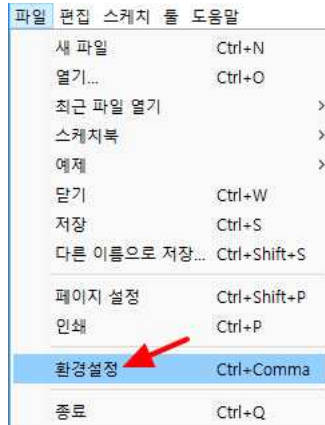
ESP32 패키지 설치하기

아두이노 소프트웨어에서 ESP32 아두이노 프로그래밍을 하기 위해서는 ESP32 아두이노 패키지를 설치해야 합니다. 다음과 같은 순서로 패키지를 설치합니다.

1. 아두이노 소프트웨어를 실행합니다.

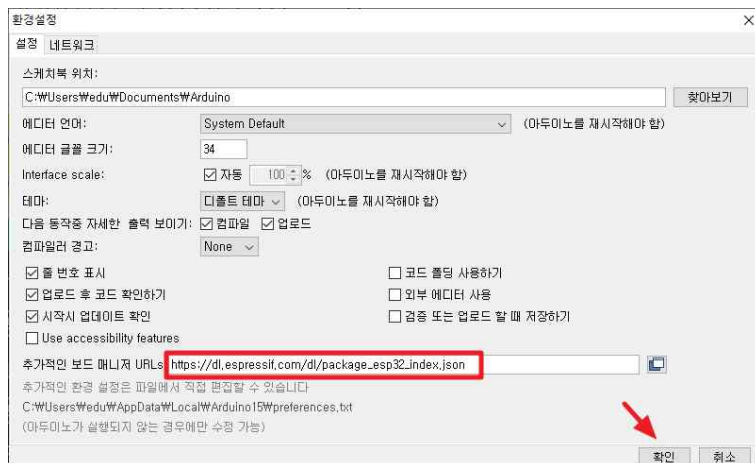


2. 아두이노 소프트웨어에서 [파일]-[환경설정] 메뉴를 선택합니다.



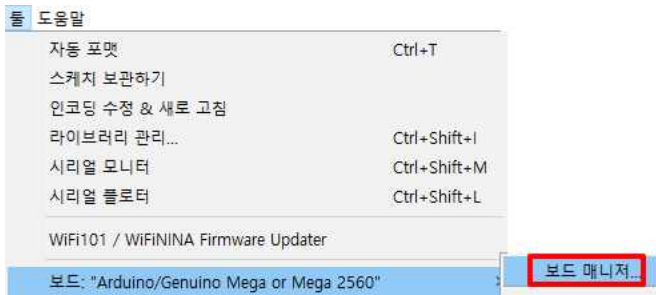
3. 환경설정 창의 [추가적인 보드 매니저 URLs] 입력 박스에 다음과 같이 입력한 후, [확인] 버튼을 눌러줍니다. 아래에서 dl은 디엘입니다.

https://dl.espressif.com/dl/package_esp32_index.json



*** AVR 계통의 아두이노가 아닌 추가적인 보드를 아두이노 소프트웨어에 추가할 때 사용하는 방법입니다. 여기서는 아두이노 소프트웨어에 ESP32 아두이노 보드 개발 환경을 추가하기 위해 해당 사이트를 환경설정에서 추가하였습니다.

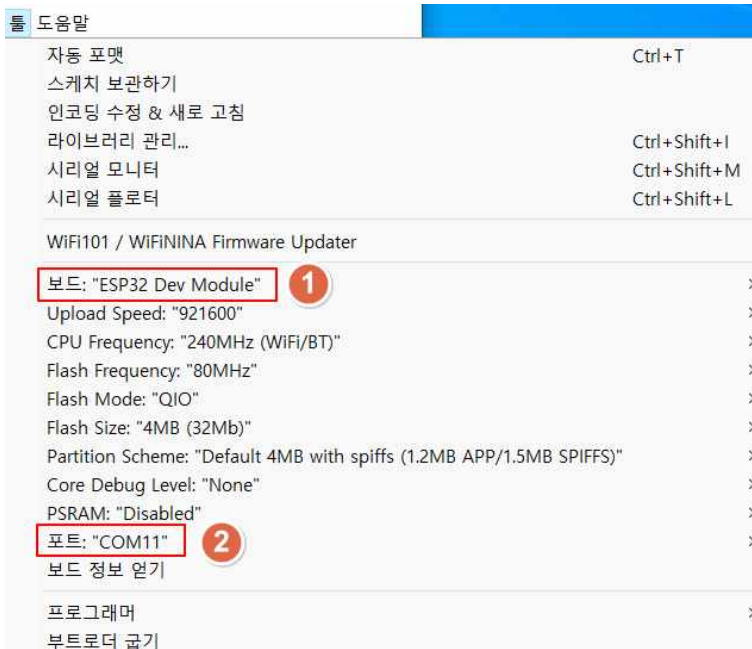
4. [툴]--[보드]--[보드 매니저...] 메뉴를 선택합니다.



5. [보드 매니저] 창이 뜨면 다음과 같이 esp32을 검색한 후, esp32 패키지를 선택한 후 [설치] 버튼을 누릅니다. 설치가 완료되면 [닫기] 버튼을 누릅니다.



6. [툴] 메뉴를 이용하여 보드, 포트를 다음과 같이 선택합니다.



*** 독자 여러분의 포트 번호는 다를 수 있으며, 해당 포트 번호를 사용하면 됩니다.

*** 우리는 포트를 통해서 아두이노 스케치 프로그램을 업로드하고, 시리얼 모니터를 통한

디버깅 메시지를 확인합니다.

04 아두이노 스케치 작성해 보기

여기서는

- 1) Hello PC 스케치를 작성한 후,
- 2) 컴파일하고,
- 3) 아두이노 보드에 업로드하고,
- 4) 시리얼 모니터를 통해 결과를 확인해 봅니다.

스케치 작성하기

1. 다음과 같이 예제를 작성합니다.

```
sketch_jan31a $  
1 void setup() {  
2   Serial.begin(115200);  
3 }  
4  
5 void loop() {  
6   Serial.println("Hello PC^^. I'm an ESP32~");  
7 }
```

2 : 아두이노가 Serial.begin 명령을 수행하여 PC로 연결된 Serial의 통신 속도를 115200bps로 설정하게 합니다. 115200bps는 초당 115200 비트를 보내는 속도입니다. 시리얼 포트를 통해 문자 하나를 보내는데 10비트가 필요합니다. 그러므로 1초에 $115200/10 = 11520$ 문자를 보내는 속도입니다. 11520 문자는 A4 용지 기준 5~6페이지 정도의 양입니다. 비트는 0 또는 1을 담을 수 있는 데이터 저장의 가장 작은 단위입니다.

6 : 아두이노가 Serial.println 명령을 수행하여 "Hello PC^^. I'm an ESP32~" 문자열을 PC로 출력하게 합니다. println은 print line의 약자입니다. ln의 l은 영문 대문자 아이(I)가 아니고 소문자 엘(l)입니다.

시리얼 통신은 다음 부분을 통해서 이루어집니다.



시리얼 통신의 원리는 종이컵과 실을 이용하여 말하고 들을 수 있는 원리와 같습니다. 우리가 하는 말이 실을 통해 순차적으로 전달되는 원리로 아두이노 보드와 컴퓨터도 통신을 하게 됩니다.

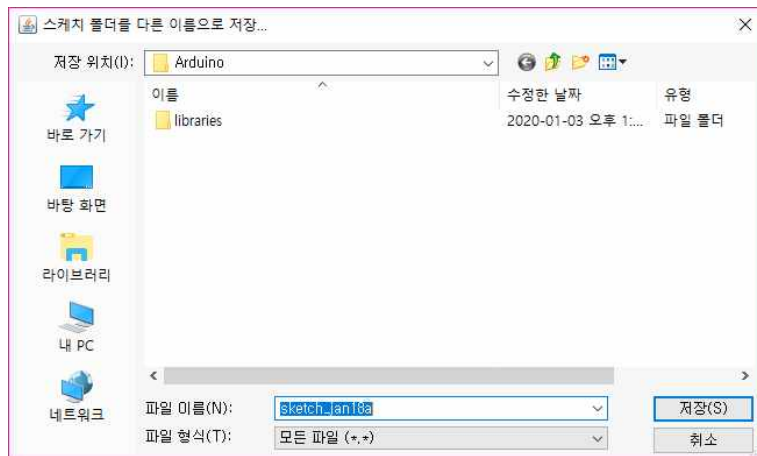


스케치 저장하기

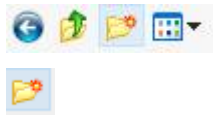
2. 다섯 번째 아이콘인 [저장] 버튼을 누릅니다.



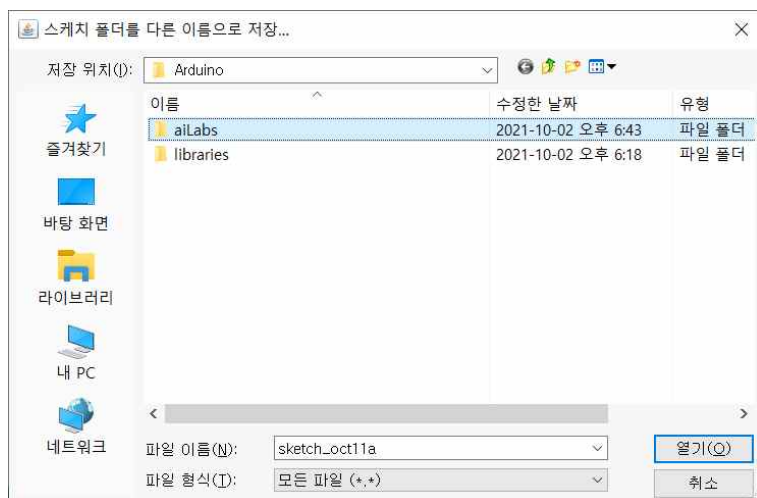
3. 그러면 다음과 같은 창이 뜹니다.



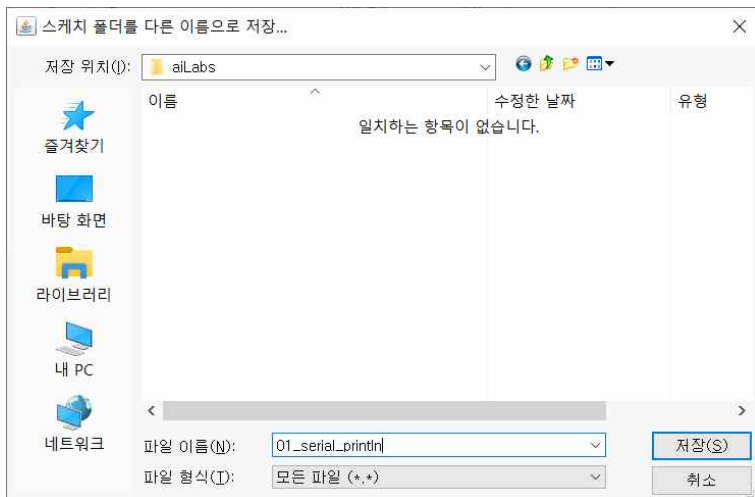
4. 프로젝트 디렉터리를 만들기 위해 오른쪽 상단에 있는 [새 폴더 만들기] 버튼을 누릅니다.



5. 디렉터리 이름을 [aiLabs]로 합니다. 한글 이름은 오류가 발생할 수 있으므로 사용하지 않습니다.



6. [aiLabs] 디렉터리로 이동하여 [01_serial_println]을 입력한 후, [저장] 버튼을 누릅니다.

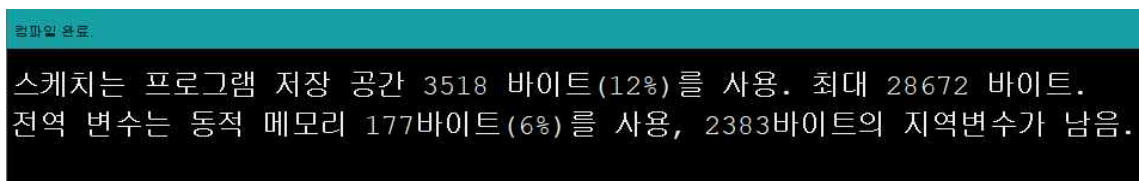


스케치 컴파일하기

7. 첫 번째 아이콘인 [확인] 버튼을 눌러 컴파일을 수행합니다.



8. [컴파일 완료]를 확인합니다.



용어설명 : 컴파일 작성한 스케치를 아두이노 보드 상에 있는 마이크로 컨트롤러가 읽을 수 있는 코드로 변형하는 작업이며, 컴파일러라는 프로그램이 이 작업을 수행합니다. 한글로 쓴 소스를 영어로 번역하여, 영어를 사용하는 사람들이 읽을 수 있도록 하는 작업과 같다고 보면 됩니다.

스케치 업로드하기

컴파일한 스케치를 아두이노 보드 상에 있는 마이컴에 쓰는 작업입니다. 업로드를 하면 전원을 꺼도 컴파일한 스케치의 내용은 마이컴 상에 남아 있습니다.

확인하기: '03 ESP32 아두이노 보드 패키지 설치하기'을 참고하여 업로드 직전에 아두이노 보드와 포트를 선택합니다.

9. [업로드] 버튼을 눌러줍니다.



10. 다음과 같이 업로드가 진행됩니다.

```
Hash of data verified.  
Compressed 206544 bytes to 104402...  
Writing at 0x00028000... (100 %)
```

11. [업로드 완료]를 확인합니다.

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

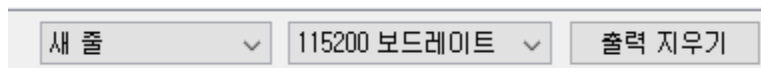
시리얼 모니터 확인

이제 결과를 시리얼 모니터를 통해 확인합니다.

12. [시리얼 모니터] 버튼을 눌러줍니다.



13. 시리얼 모니터 창이 뜨면, 우측 하단에서 통신 속도를 115200으로 맞춰줍니다.



16. 다음과 같은 메시지가 반복적으로 뜨는 것을 확인합니다.

```
Hello PC^^. I'm an ESP32~  
Hello PC^^. I'm an ESP32~  
Hello PC^^. I'm an ESP32~  
Hello PC^^. I'm an ESP32~  
Hello PC^^. I'm an ESP32~
```

loop 함수가 반복해서 호출되기 때문에 메시지도 반복해서 뿌려지게 됩니다.

05 아두이노 오류 발생시 대처방법

앞으로 여러분은 아두이노를 다루는 과정에서 몇 가지 정형화된 오류를 반복적으로 접하게 됩니다. 이 때는 다음 순서로 문제를 해결해 보도록 합니다.

1. USB 포트 연결에 문제가 발생하는 경우가 있습니다. 이 경우엔 USB 연결을 해제한 후 다시 연결합니다.
2. 아두이노 소프트웨어에서 설정이 제대로 안되어 있는 경우가 있습니다. 이 경우엔 아두이노 소프트웨어의 tool 메뉴에서 보드와 포트가 제대로 선택되어 있는지 확인합니다.
3. 아두이노 스케치에 C 문법 오류가 있는 경우가 있습니다. 이 경우엔 반점(; -세미콜론), 괄호(소괄호(), 중괄호{}, 대괄호[]), 점(.), 함수 색상 순서로 확인해 봅니다.
4. 아두이노 보드 자체에 하드웨어적인 문제가 있는 경우가 있습니다. 이 경우엔 blink 예제로 아두이노 보드의 상태를 확인합니다.

06 아두이노 스케치 구조 이해하기

여기서는 아두이노 스케치의 구조에 대해서 자세히 살펴보도록 합니다.

아두이노 스케치는 다음과 같이 두 개의 기본 함수로 구성됩니다.

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

setup 함수는 코드 실행을 시작할 때 한 번만 수행되며, 사용하고자 하는 하드웨어(입출력 모듈:센서, 모터 등)를 초기화시키는 부분입니다. loop 함수는 반복적으로 수행되며, 하드웨어를 반복적으로 동작 시키는 부분입니다.

*** 함수는 기능이라는 의미로 수학에서 유래하였으며 원하는 기능을 수행하기 위한 명령의 집합으로 구성됩니다.

1. setup 함수를 다음과 같이 수정해 봅니다.


```
00_hello_pc
1 void setup() {
2   Serial.begin(115200);
3
4   Serial.println("setup");
5 }
6
7 void loop() {
8
9
10 }
```

2: 시리얼 포트의 통신 속도를 115200으로 설정합니다.

4: setup 문자열을 시리얼 포트에 출력합니다.

2. 업로드를 수행합니다.

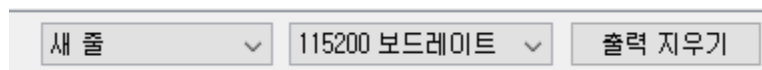


*** 업로드를 수행하면 컴파일 과정이 자동으로 포함됩니다.

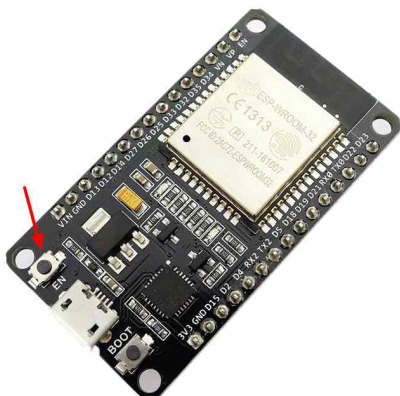
3. [시리얼 모니터] 버튼을 클릭합니다.



4. 시리얼 모니터 창이 뜨면, 우측 하단에서 통신 속도를 115200으로 맞춰줍니다.



5. 다음에 표시된 [EN] 버튼을 눌러 재부팅을 수행합니다.



6. setup 문자열이 출력되는 것을 확인합니다.

```
setup
```

7. 이번엔 loop 함수를 다음과 같이 수정합니다.

```
00_hello_pc
1 void setup() {
2   Serial.begin(115200);
3
4   Serial.println("setup");
5 }
6
7 void loop() {
8   Serial.println("loop");
9
10 }
```

8: loop 문자열을 시리얼 포트에 출력합니다.

8. 업로드를 수행합니다.



*** 업로드를 수행하면 컴파일 과정이 자동으로 포함됩니다.

9. 시리얼 모니터 창에 loop 문자열이 계속해서 출력되는 것을 확인합니다.

```
loop
loop
loop
loop
loop
```