



MachineJW github - <https://github.com/MachineJW/Arduino>

이 문서는 ESP32 MCU 제조사 ESPRESSIF 홈페이지에서 가져온 문서입니다.

## ESP32-DevKitC

ESP32-DevKitC는 ESP32 시리즈의 일부인 낮은 설치 공간 및 보급형 개발 보드입니다.

이 보드에는 풍부한 주변 장치 세트가 있습니다. 내장 ESP32 핀아웃은 번거로움 없는 프로토타이핑에 최적화되어 있습니다!



## 주요 특징들



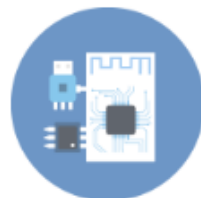
### Wi-Fi 및 블루투스 연결

이 최소 시스템 개발 보드는 ESP32 모듈로 구동됩니다. Wi-Fi 및 Bluetooth 기능을 통합하고 신속한 프로토타이핑을 위한 풍부한 주변 장치 세트를 제공합니다!



### 신속한 프로토 타입

ESP32-DevKitC는 최적의 RF 성능을 달성합니다. RF 성능 및 안테나 설계에 대해 걱정할 필요 없이 애플리케이션 설계 및 개발에 바로 들어갈 수 있습니다. ESP32-DevKitC에는 기본 시스템 요구 사항이 이미 포함되어 있습니다. USB 케이블을 연결하기만 하면 준비가 완료됩니다!



### 유연하고 다양한 기능

ESP32-DevKitC에는 ESP32-WROOM 시리즈, ESP32-WROVER 시리즈 및 ESP32-SOLO 시리즈 모듈의 전체 지원 회로가 포함되어 있으며 USB-UART 브리지, 리셋 및 부트 모드 버튼, LDO 레귤레이터 및 마이크로-USB 커넥터. 모든 중요한 GPIO는 개발자가 사용할 수 있습니다.



### 브레드보드 친화적

ESP32-DevKitC 핀아웃은 브레드보드에서 프로토타이핑이 가능하도록 최적화되어 있습니다. 온보드 LDO 출력은 추가적인 오프보드 전자 장치에 전원을 공급하기 위해 출력됩니다. 주변 장치 출력은 번거롭지 않은 프로토타이핑을 위해 함께 그룹화됩니다.

### 아마존 인증 기기

ESP32-DevKitC 개발 보드는 Amazon Web Services(AWS)의 인증을 받았습니다. Espressif의 자체 IoT 개발 프레임워크인 ESP-IDF 외에도 ESP32-DevKitC에서 FreeRTOS를 사용할 수 있습니다. FreeRTOS는 AWS IoT, AWS Greengrass 및 기타 AWS 서비스와의 즉시 사용 가능한 연결을 제공합니다.

FreeRTOS 시작하기



qualified  
device  
FreeRTOS

# ESP32-DevKitC V4 시작 가이드

이 가이드는 ESP32-DevKitC V4 개발 보드 사용을 시작하는 방법을 보여줍니다.

## 필요한 것

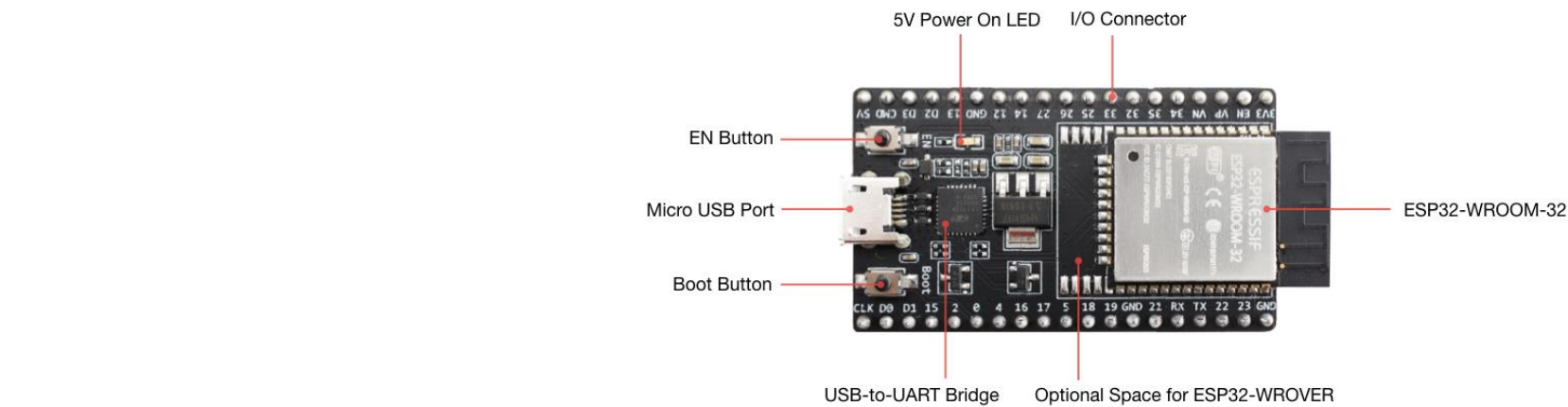
- [ESP32-DevKitC V4 보드](#)
- USB A / 마이크로 USB B 케이블
- Windows, Linux 또는 macOS를 실행하는 컴퓨터

## 개요

[ESP32-DevKitC V4](#)는 [Espressif](#) 에서 생산하는 소형 ESP32 기반 개발 보드입니다 . 대부분의 I/O 핀은 쉽게 인터페이스할 수 있도록 양쪽에 있는 핀 헤더로 분리되어 있습니다. 개발자는 주변기기를 점퍼 와이어로 연결하거나 ESP32-DevKitC V4를 브레드보드에 장착할 수 있습니다. 광범위한 사용자 요구 사항을 충족하기 위해 다음 버전의 ESP32-DevKitC V4를 사용할 수 있습니다.

- [ESP32-WROOM-DA](#)
- [ESP32-WROOM-32E](#)
- [ESP32-WROOM-32UE](#)
- [ESP32-WROOM-32D](#)
- [ESP32-WROOM-32U](#)
- [ESP32-SOLO-1](#)
- [ESP32-WROVER-E](#)
- [ESP32-WROVER-IE](#)

기능 설명



ESP32-WROOM-32 모듈이 납땜 된 ESP32-DevKitC V4

Key Component	Description
ESP32-WROOM-32	핵심에 ESP32가 있는 모듈입니다. 자세한 내용은 <a href="#">ESP32-WROOM-32 데이터시트</a> 를 참조하십시오 .
EN	리셋 버튼.
Boot	다운로드 버튼. Boot 를 누른 상태에서 EN을 누르면 <b>직렬 포트</b> 를 통해 펌웨어를 다운로드하기 위한 펌웨어 다운로드 모드가 시작됩니다.
USB-to-UART Bridge	단일 USB-UART 브리지 칩은 최대 3Mbps의 전송 속도를 제공합니다.
Micro USB Port	USB 인터페이스. 보드용 전원 공급 장치 및 컴퓨터와 ESP32-WROOM-32 모듈 간의 통신 인터페이스.
5V Power On LED	USB 또는 외부 5V 전원 공급 장치가 보드에 연결되면 켜집니다. 자세한 내용은 <a href="#">관련 문서</a> 의 회로도를 참조하십시오
I/O	ESP 모듈의 대부분의 핀은 보드의 핀 헤더로 분리되어 있습니다. ESP32를 프로그래밍하여 <b>PWM, ADC, DAC, I2C, I2S, SPI</b> 등과 같은 여러 기능을 활성화 할 수 있습니다.

## 전원 공급 장치 옵션

전원을 공급하는 세 가지 상호 배타적인 방법이 있습니다.

- 마이크로 USB 포트, 기본 전원 공급 장치
- 5V / GND 헤더 핀
- 3V3 / GND 헤더 핀

### ⓘ 경고

전원 공급 장치는 위의 옵션 중 하나만 사용하여 제공해야 합니다. 그렇지 않으면 보드 및/또는 전원 공급 장치가 손상될 수 있습니다.

1. USB 포트를 통해.
2. 5V와 12V 사이의 조정되지 않은 전압을 사용하여 5V 및 GND 핀에 연결합니다. 이 전압은 온 보드에서 조정됩니다.
3. 조정된 3.3V 전압을 사용하여 3.3V 및 GND 핀에 연결합니다. 3.3V 제한을 초과하지 않도록 주의하십시오. 그렇지 않으면 ESP32 모듈이 손상됩니다.

**주의 : 동시에 이러한 옵션 중 하나만 사용하도록 매우 주의 하십시오.**

예를 들어, USB를 통해 모듈을 컴퓨터에 연결하는 동시에 10V 입력을 사용하여 5V 핀을 통해 ESP32 개발 키트에 전원을 공급하지 마십시오. 이것은 확실히 당신의 모듈과 컴퓨터를 손상시킬 것입니다.

### ESP32 Power options

1: USB



ESP32 for busy people Tech Explorations

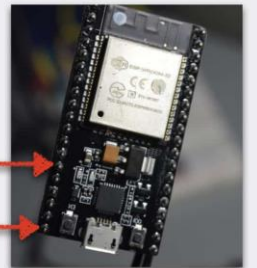
### ESP32 Power options

2: 5V / GND header pins

CAUTION: Keep input voltage below 12V to reduce heat on the voltage regulator

GND

5V



ESP32 for busy people Tech Explorations

### ESP32 Power options

3: 3.3V / GND header pins

CAUTION: Voltage must be regulated externally. Do not provide more than 3.3V on the 3.3V pin!

3.3V

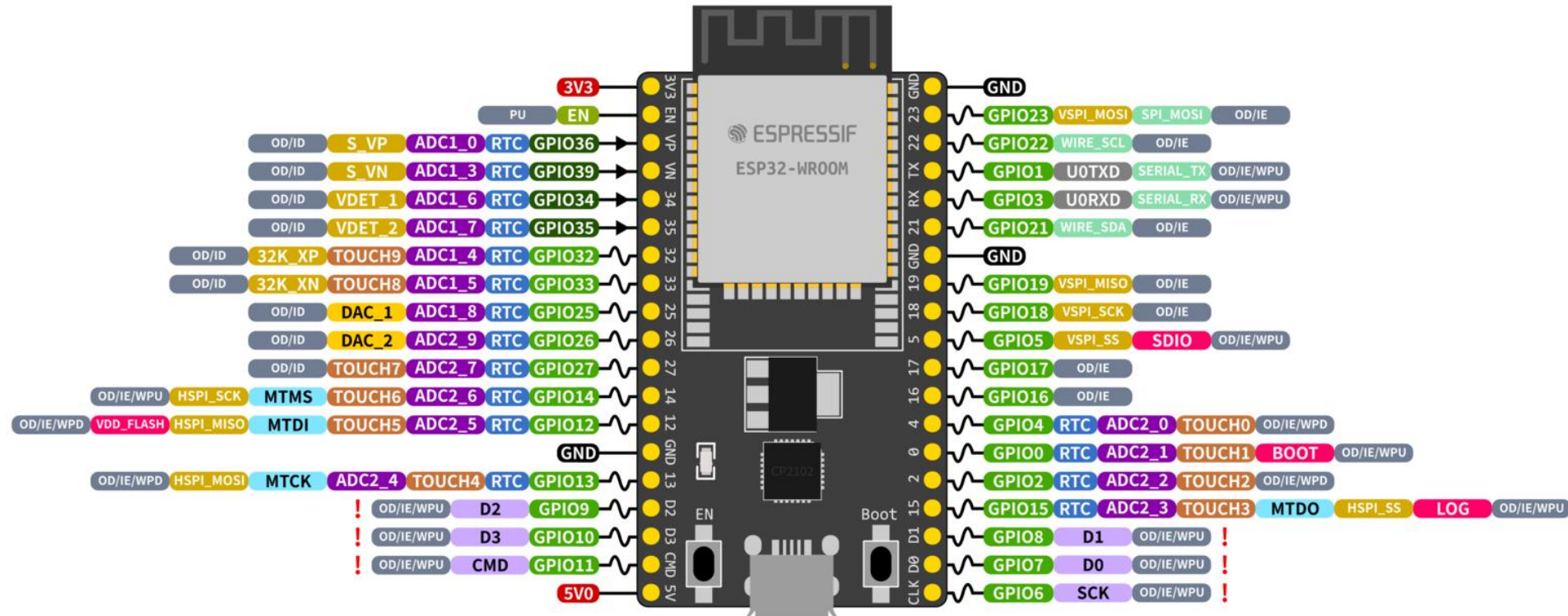
GND



ESP32 for busy people Tech Explorations

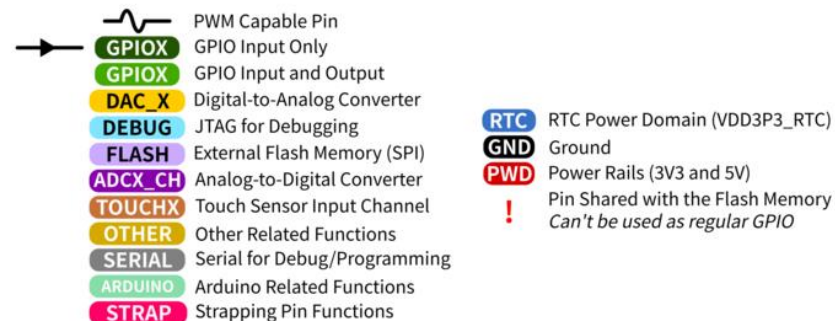


# ESP32-DevKitC



## ESP32 Specs

32-bit Xtensa® dual-core @240MHz  
Wi-Fi IEEE 802.11 b/g/n 2.4GHz  
Bluetooth 4.2 BR/EDR and BLE  
520 KB SRAM (16 KB for cache)  
448 KB ROM  
34 GPIOs, 4x SPI, 3x UART, 2x I2C,  
2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,  
1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet



**GPIO STATE**

- WPU:** Weak Pull-up (Internal)
- WPD:** Weak Pull-down (Internal)
- PU:** Pull-up (External)
- IE:** Input Enable (After Reset)
- ID:** Input Disabled (After Reset)
- OE:** Output Enable (After Reset)
- OD:** Output Disabled (After Reset)

# 헤더 블록

아래 두 표는 [ESP32-WROOM-32 모듈이 납땜된 ESP32-DevKitC V4에](#) 표시된 것처럼 보드 양쪽에 있는 I/O 헤더 핀 의 이름 과 기능 을 제공합니다 .

No.	Name	Type	Function
1	3V3	P	3.3 V power supply
2	EN	I	CHIP_PU, Reset
3	VP	I	GPIO36, ADC1_CH0, S_VP
4	VN	I	GPIO39, ADC1_CH3, S_VN
5	IO34	I	GPIO34, ADC1_CH6, VDET_1
6	IO35	I	GPIO35, ADC1_CH7, VDET_2
7	IO32	I/O	GPIO32, ADC1_CH4, TOUCH_CH9, XTAL_32K_P
8	IO33	I/O	GPIO33, ADC1_CH5, TOUCH_CH8, XTAL_32K_N
9	IO25	I/O	GPIO25, ADC1_CH8, DAC_1
10	IO26	I/O	GPIO26, ADC2_CH9, DAC_2
11	IO27	I/O	GPIO27, ADC2_CH7, TOUCH_CH7
12	IO14	I/O	GPIO14, ADC2_CH6, TOUCH_CH6, MTMS
13	IO12	I/O	GPIO12, ADC2_CH5, TOUCH_CH5, MTDI
14	GND	G	Ground
15	IO13	I/O	GPIO13, ADC2_CH4, TOUCH_CH4, MTCK
16	D2	I/O	GPIO9, D2
17	D3	I/O	GPIO10, D3
18	CMD	I/O	GPIO11, CMD
19	5V	P	5 V power supply



No.	Name	Type <a href="#">1</a>	Function
1	GND	G	Ground
2	IO23	I/O	GPIO23
3	IO22	I/O	GPIO22
4	TX	I/O	GPIO1, U0TXD
5	RX	I/O	GPIO3, U0RXD
6	IO21	I/O	GPIO21
7	GND	G	Ground
8	IO19	I/O	GPIO19
9	IO18	I/O	GPIO18
10	IO5	I/O	GPIO5
11	IO17	I/O	GPIO17 <a href="#">3</a>
12	IO16	I/O	GPIO16 <a href="#">3</a>
13	IO4	I/O	GPIO4, ADC2_CH0, TOUCH_CH0
14	IO0	I/O	GPIO0, ADC2_CH1, TOUCH_CH1, Boot
16	IO2	I/O	GPIO2, ADC2_CH2, TOUCH_CH2
17	IO15	I/O	GPIO15, ADC2_CH3, TOUCH_CH3, MTDO
17	D1	I/O	GPIO8, D1 <a href="#">2</a>
18	D0	I/O	GPIO7, D0 <a href="#">2</a>
19	CLK	I/O	GPIO6, CLK <a href="#">2</a>

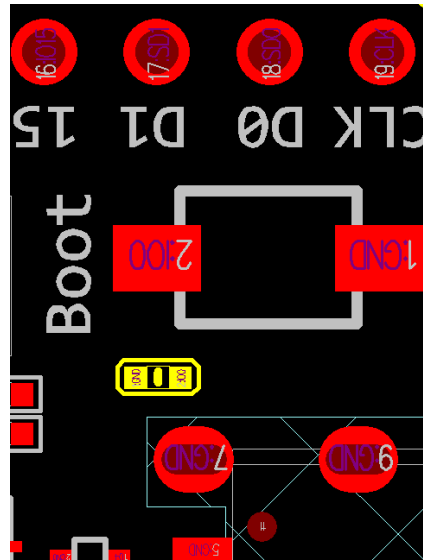
P: Power supply; I: Input; O: Output.

핀 D0, D1, D2, D3, CMD 및 CLK는 ESP32와 SPI 플래시 메모리 간의 통신을 위해 내부적으로 사용됩니다. USB 커넥터 근처의 양쪽에 그룹화되어 있습니다. SPI 플래시 메모리/SPI RAM에 대한 액세스를 방해할 수 있으므로 이러한 핀을 사용하지 마십시오.  
GPIO16 및 GPIO17 핀은 ESP32-WROOM 및 ESP32-SOLO-1 모듈이 있는 보드에서만 사용할 수 있습니다. ESP32-WROVER 모듈이 있는 보드에는 내부용으로 예약된 핀이 있습니다.

## C15에 대한 참고 사항

구성 요소 C15는 이전 ESP32-DevKitC V4 보드에서 다음 문제를 일으킬 수 있습니다.

- 보드가 다운로드 모드로 부팅될 수 있음
- GPIO0에서 클럭을 출력하면 C15가 신호에 영향을 줄 수 있습니다.
- 이러한 문제가 발생하면 구성 요소를 제거하십시오. 아래 그림은 노란색으로 강조 표시된 C15의 위치를 보여줍니다.



*ESP32-DevKitC V4 보드의 C15(노란색) 위치*



## 애플리케이션 개발 시작

ESP32-DevKitC V4의 전원을 켜기 전에 보드가 손상의 흔적 없이 양호한 상태인지 확인하십시오.

[그런 다음 시작하기](#) 로 진행 하십시오 . 여기서 섹션 [설치](#) 는 개발 환경을 빠르게 설정하고 보드에 예제 프로젝트를 플래시하는 데 도움이 됩니다.

## 시작하다

이 문서는 Espressif의 ESP32 칩을 기반으로 하는 하드웨어에 대한 소프트웨어 개발 환경을 설정하는 데 도움을 주기 위한 것입니다. 그런 다음 간단한 예제에서 메뉴 구성에 ESP-IDF( Espressif IoT Development Framework )를 사용한 다음 ESP32 보드에 펌웨어를 빌드하고 플래싱하는 방법을 보여줍니다.

### 0 메모

ESP-IDF의 마스터 브랜치(최신 버전)에 대한 문서입니다. 이 버전은 계속 개발 중입니다. [안정 버전](#) 문서와 다른 [ESP-IDF 버전](#) 을 사용할 수 있습니다 .

## 소개

**ESP32는 다음 기능을 통합한 시스템 온 칩입니다.**

- Wi-Fi(2.4GHz 대역)
- 블루투스
- 듀얼 고성능 Xtensa® 32비트 LX6 CPU 코어
- 초저전력 보조 프로세서
- 여러 주변기기

40nm 기술로 구동되는 ESP32는 효율적인 전력 사용, 컴팩트한 디자인, 보안, 고성능 및 신뢰성에 대한 지속적인 요구를 충족하는 데 도움이 되는 강력하고 고도로 통합된 플랫폼을 제공합니다.

Espressif는 애플리케이션 개발자가 ESP32 시리즈 하드웨어를 사용하여 아이디어를 실현할 수 있도록 기본 하드웨어 및 소프트웨어 리소스를 제공합니다.

다. Espressif의 소프트웨어 개발 프레임워크는 Wi-Fi, Bluetooth, 전원 관리 및 기타 여러 시스템 기능을 갖춘 사물 인터넷(IoT) 애플리케이션 개발을 위한 것입니다.

## 필요한 것

### 하드웨어

- ESP32 **보드** .
- **USB 케이블** - USB A / 마이크로 USB B.
- Windows, Linux 또는 macOS를 실행하는 **컴퓨터** .

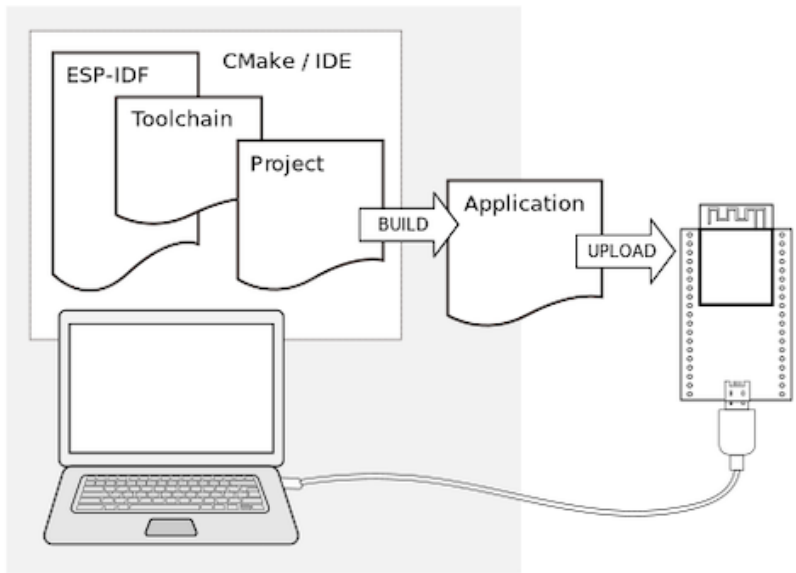
#### ! 메모

현재 일부 개발 보드는 USB Type C 커넥터를 사용하고 있습니다. 보드를 연결하는 올바른 케이블이 있는지 확인하십시오!

### 소프트웨어

**ESP32** 에서 ESP-IDF 사용을 시작하려면 다음 소프트웨어를 설치하세요.

- **ESP32** 용 코드를 컴파일하는 도구 모음
- **빌드 도구** - ESP32용 전체 **애플리케이션** 을 빌드하기 위한 CMake 및 Ninja
- ESP32용 API(소프트웨어 라이브러리 및 소스 코드)와 **Toolchain** 을 작동하는 스크립트를 기본적으로 포함하는 **ESP-IDF**



## IDE

### ① 메모

선호하는 IDE를 통해 ESP-IDF를 설치하는 것이 좋습니다.

- 이클립스 플러그인
- VSCode 확장

## 수동 설치

수동 절차의 경우 운영 체제에 따라 선택하십시오.

- [윈도우 인스톨러](#)
- [리눅스와 맥OS](#)

## 첫 번째 프로젝트 구축

[ESP-IDF가 이미 설치되어 있고 IDE를 사용하지 않는 경우 Windows에서 프로젝트](#) 시작 또는 [Linux 및 macOS에서 프로젝트](#) 시작 다음 명령줄에서 첫 번째 프로젝트를 빌드할 수 있습니다 .

# 프로그래밍 환경

---

ESP32는 다양한 프로그래밍 환경에서 프로그래밍할 수 있습니다. 당신이 사용할 수 있는:

- [아두이노 IDE](#)
- Espressif IDF(IoT 개발 프레임워크)
- [마이크로파이썬](#)
- 자바스크립트
- 루아
- ...

## ESP32 핀아웃 가이드

ESP32 에는 ESP8266에 비해 더 많은 기능과 더 [많은 GPIO 가 있습니다](#).

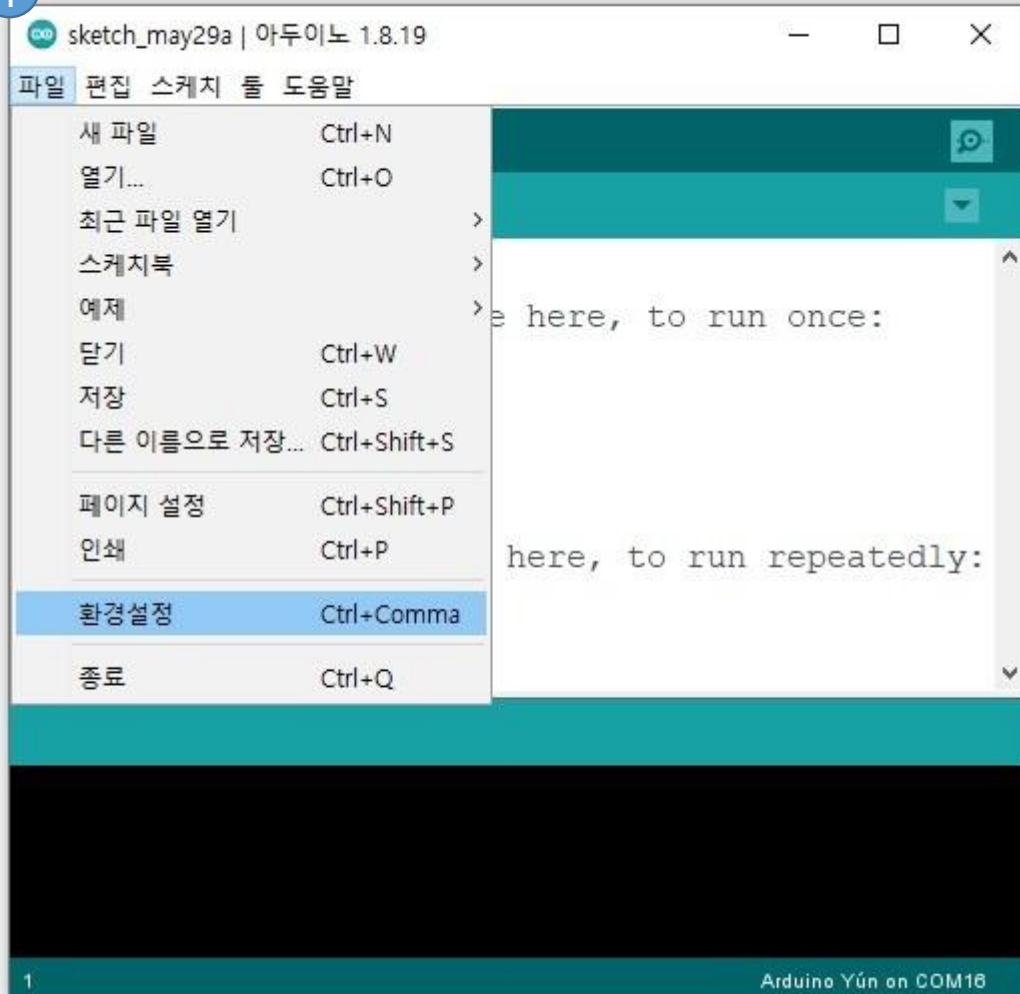
ESP32를 사용하면 UART, I2C 또는 SPI 핀을 결정할 수 있습니다. 코드에서 설정하기만 하면 됩니다. 이것은 ESP32 칩의 멀티플렉싱 기능으로 인해 동일한 핀에 여러 기능을 할당할 수 있기 때문에 가능합니다. 코드에서 설정하지 않으면 아래 그림과 같이 핀이 기본값으로 사용됩니다(핀 위치는 제조업체에 따라 다를 수 있음).

## Arduino IDE 개발환경 설정

### Getting Started with ESP32 WROOM Devkitc v4



1



2





3

추가적인 보드 매니저 URLs

각 행에 하나의 추가적인 URL을 입력하세요

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\_esp32\_dev\_index.json

비공식 보드 지원 URL 목록을 보려면 클릭하세요

확인

취소

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)

5

보드 매니저

타입

All

esp32

esp32

by Espressif Systems

보드는 이 패키지에 포함:  
ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board.

[More Info](#)

2.0.3

설치

보드 매니저

타입

All

esp32

esp32

by Espressif Systems

보드는 이 패키지에 포함:  
ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board.

[More Info](#)

설치...

닫기

4

sketch\_may29a | 아두이노 1.8.19

파일 편집 스케치 도구 도움말

자동 포맷  
Ctrl+T

스케치 보관하기

인코딩 수정 & 새로 고침  
Ctrl+Shift+I

라이브러리 관리...  
Ctrl+Shift+M

시리얼 모니터  
Ctrl+Shift+L

WIFI101 / WIFININA Firmware Updater

보드: "Arduino Yún"

포트

보드 정보 얻기

프로그래머: "AVRISP mkII"

부트로더 굽기

보드 매니저...

• Arduino Yún

Arduino Uno

Arduino Duemilanove or Diecimila

Arduino Nano

Arduino Mega or Mega 2560

Arduino Mega ADK

Arduino Leonardo

Arduino Leonardo ETH

Arduino Micro

Arduino Espora

Arduino Mini

Arduino Ethernet

Arduino Fio

Arduino 8T

LilyPad Arduino USB

LilyPad Arduino

Arduino Pro or Pro Mini

Arduino NG or older

Arduino Robot Control

Arduino Robot Motor

Arduino Gemma

Adafruit Circuit Playground

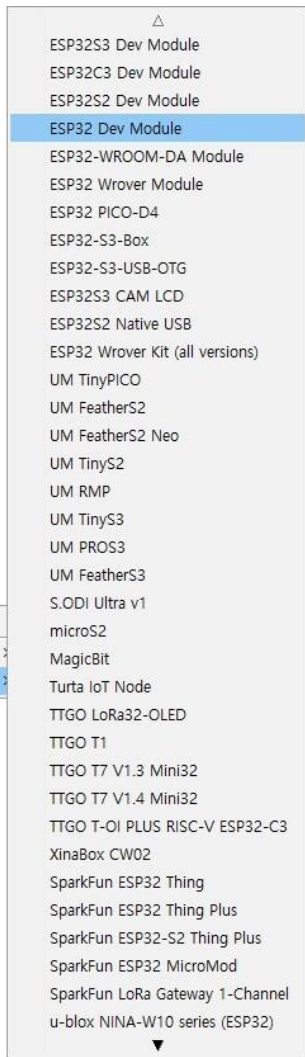
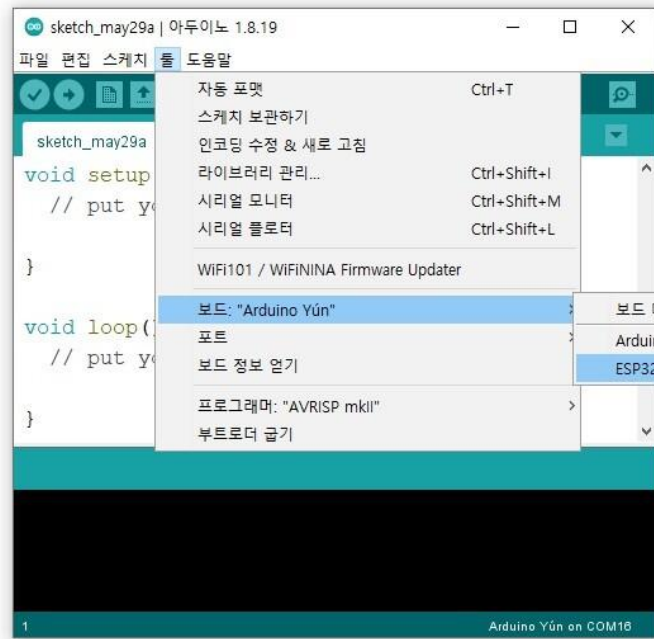
Arduino Yún Mini

Arduino Industrial 101

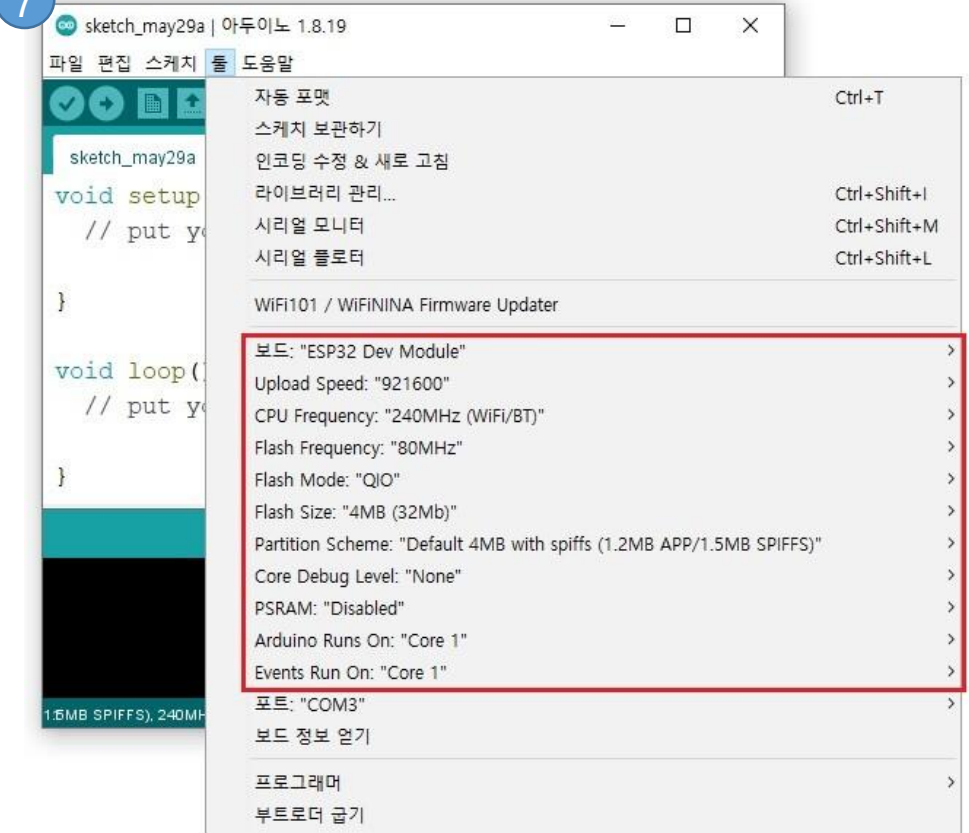
Linino One

Arduino Uno WIFI

6



7

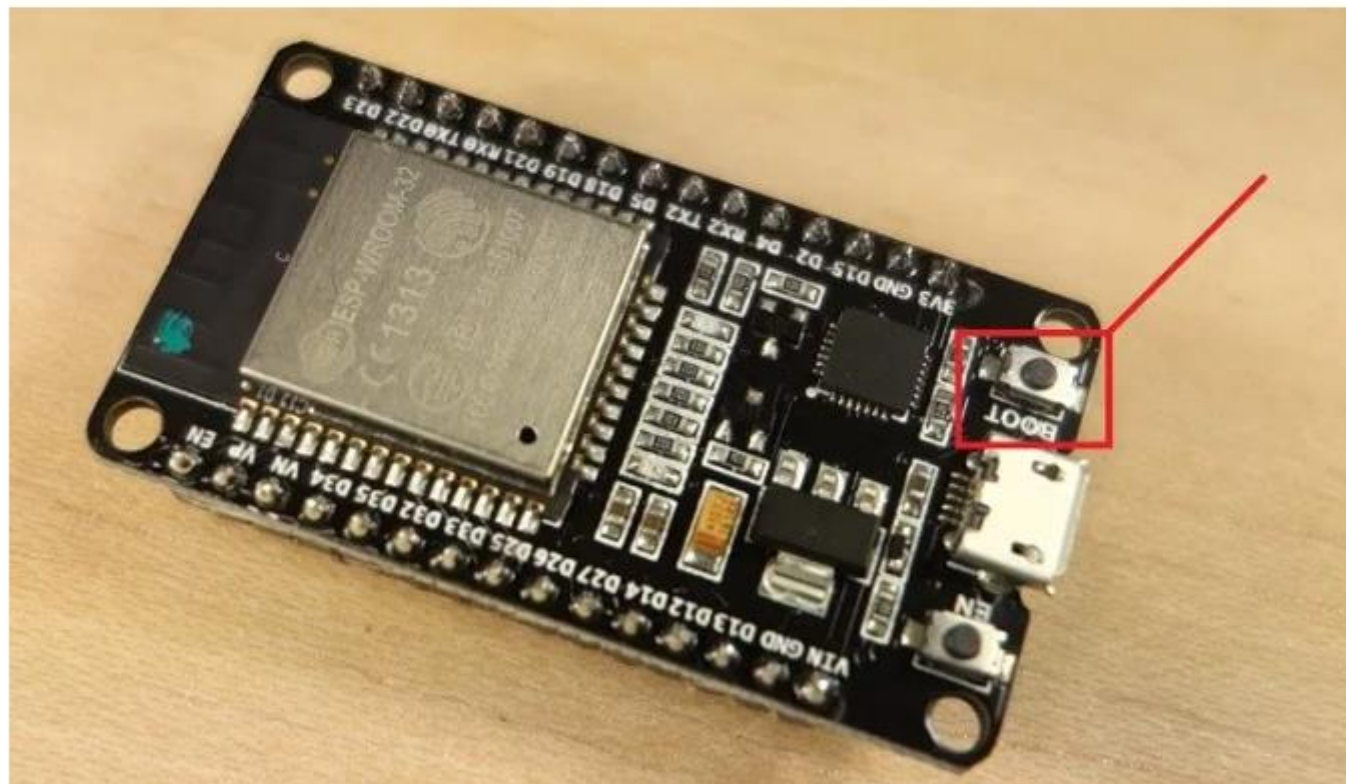


## 문제 해결

ESP32에 새 스케치를 업로드하려고 하면 "치명적인 오류가 발생했습니다: ESP32에 연결하지 못했습니다: 시간 초과... 연결 중...". ESP32가 깜박임/업로드 모드가 아님을 의미합니다.

올바른 보드 이름과 COM 포트를 선택하고 다음 단계를 따르십시오.

- ESP32 보드의 "BOOT" 버튼을 길게 누릅니다.



- Arduino IDE에서 "업로드" 버튼을 눌러 스케치를 업로드합니다.

ESP32 개발 사이트

<https://randomnerdtutorials.com/>  
<https://esp32io.com/esp32-tutorials>

ESP32 MCU 제조 공식 사이트

<https://www.espressif.com/en/products/devkits/esp32-devkitc>