

[ESP32 GPIO]

MachineJW

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

□ ESP32에 포함되어 있는 기능 핀

1. 18개의 아날로그-디지털 변환기(ADC) 채널
2. 3개의 SPI 인터페이스
3. 3개의 UART 인터페이스
4. 2개의 I2C 인터페이스
5. 16 PWM 출력 채널
6. 2개의 디지털-아날로그 컨버터(DAC)
7. 2개의 I2S 인터페이스
8. 10 정전 용량 감지 GPIO

ADC(아날로그-디지털 변환기) 및 DAC(디지털-아날로그 변환기) 기능은 특정 **고정 핀에** 할당됩니다. 그러나 UART, I2C, SPI, PWM 등의 핀을 결정할 수 있습니다. 코드에서 할당하기만 하면 됩니다. 이것은 **ESP32 칩의 다중화 기능**으로 인해 가능합니다.

녹색으로 강조 표시된 핀은 사용할 수 있습니다. 노란색으로 강조 표시된 것은 사용하기에 적합하지만 주로 부팅 시 예기치 않은 동작을 할 수 있으므로 주의가 필요합니다. 빨간색으로 강조 표시된 핀은 입력 또는 출력으로 사용하지 않는 것이 좋습니다.

GPIO	Input	Output	Notes
0	pullup	OK	부팅시 PWM 신호 출력
1	TX pin	OK	부팅 시 디버그 출력
2	OK	OK	온보드 LED에 연결
3	OK	RX pin	부팅 시 높음
4	OK	OK	
5	OK	OK	부팅시 PWM 신호 출력
6	X	X	통합 SPI 플래시에 연결
7	X	X	통합 SPI 플래시에 연결
8	X	X	통합 SPI 플래시에 연결

9	✗	✗	통합 SPI 플래시에 연결
10	✗	✗	통합 SPI 플래시에 연결
11	✗	✗	통합 SPI 플래시에 연결
12	OK	OK	높게 당기면 부팅 실패
13	OK	OK	
14	OK	OK	부팅시 PWM 신호 출력
15	OK	OK	부팅시 PWM 신호 출력
16	OK	OK	
17	OK	OK	
18	OK	OK	
19	OK	OK	
21	OK	OK	
22	OK	OK	
23	OK	OK	
25	OK	OK	
26	OK	OK	
27	OK	OK	
32	OK	OK	
33	OK	OK	
34	OK		input only
35	OK		input only
36	OK		input only
39	OK		input only

GPIO 34~39는 입력 전용 핀인 GPI입니다. 이 핀에는 내부 풀 업 또는 풀다운 저항이 없습니다. 출력으로 사용할 수 없으므로 다음 핀을 입력으로만 사용하십시오. (GPIO 34, GPIO 35, GPIO 36, GPIO 39)

GPIO 6 ~ GPIO 11은 일부 ESP32 개발 보드에 노출되어 있습니다. 그러나 이 핀은 ESP-WROOM-32 칩의 통합 SPI 플래시에 연결되며 다른 용도로는 권장되지 않습니다. 따라서 프로젝트에서 다음 핀을 사용하지 마십시오.

ESP32에는 10개의 내부 정전식 터치 센서가 있습니다. 이들은 인간의 피부와 같이 전하를 보유하는 모든 것의 변화를 감지할 수 있습니다. 따라서 손가락으로 GPIO를 만질 때 발생하는 변화를 감지할 수 있습니다. 이 핀은 정전식 패드에 쉽게 통합되어 기계식 버튼을 대체할 수 있습니다. 정전식 터치 핀을 사용 하여 ESP32를 깊은 절전모드에서 깨울 수도 있습니다.

T0(GPIO 4), T1(GPIO 0), T2(GPIO 2), T3(GPIO 15), T4(GPIO 13), T5(GPIO 12)
T6(GPIO 14), T7(GPIO 27), T8(GPIO 33), T9(GPIO 32)

□ 아날로그-디지털 변환기(ADC)

ESP32에는 18 x 12비트 ADC 입력 채널이 있습니다(ESP8266에는 1x 10비트 ADC만 있음). ADC 및 각 채널로 사용할 수 있는 GPIO는 다음과 같습니다.

ADC1_CH0(GPIO 36)

ADC1_CH1(GPIO 37)

ADC1_CH2(GPIO 38)

ADC1_CH3(GPIO 39)

ADC1_CH4(GPIO 32)

ADC1_CH5(GPIO 33)

ADC1_CH6(GPIO 34)

ADC1_CH7(GPIO 35)

ADC2_CH0(GPIO 4)

ADC2_CH1(GPIO 0)

ADC2_CH2(GPIO 2)

ADC2_CH3(GPIO 15)

ADC2_CH4(GPIO 13)

ADC2_CH5(GPIO 12)

ADC2_CH6(GPIO 14)

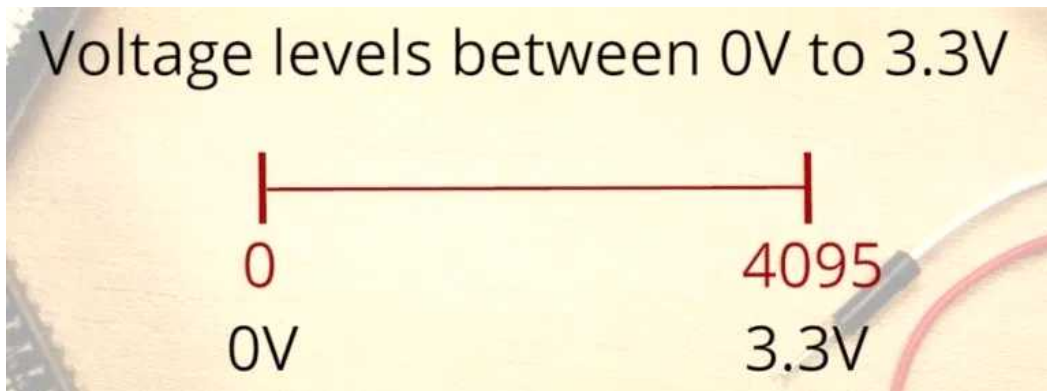
ADC2_CH7(GPIO 27)

ADC2_CH8(GPIO 25)

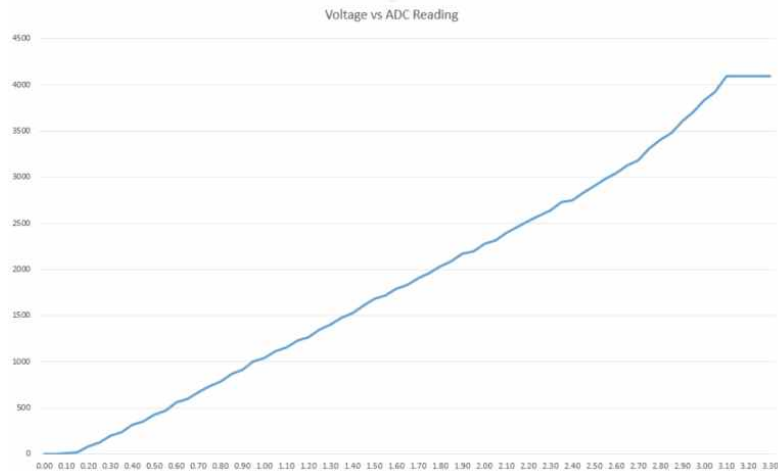
ADC2_CH9(GPIO 26)

참고: Wi-Fi를 사용할 때는 ADC2 핀을 사용할 수 없습니다. 따라서 Wi-Fi를 사용 중이고 ADC2 GPIO에서 값을 가져오는 데 문제가 있는 경우 ADC1 GPIO를 대신 사용할 수 있습니다. 그것은 당신의 문제를 해결해야 합니다.

ESP32로 아날로그 값을 읽으면 0V와 3.3V 사이의 다양한 전압 레벨을 측정할 수 있습니다. 그런 다음 측정된 전압은 0에서 4095 사이의 값에 할당됩니다. 여기서 0V는 0에 해당하고 3.3V는 4095에 해당합니다. 0V와 3.3V 사이의 모든 전압에는 그 사이의 해당 값이 지정됩니다.



ESP32 ADC 핀에는 선형 동작이 없습니다. 아마도 0과 0.1V 또는 3.2와 3.3V를 구별하지 못할 것입니다. **ADC 핀을 사용할 때 이를 염두에** 두어야 합니다. 다음 그림에 표시된 것과 유사한 동작을 얻게 됩니다.



디지털-아날로그 변환기(DAC)

ESP32에는 디지털 신호를 아날로그 전압 신호 출력으로 변환하기 위한 2 x 8비트 DAC 채널이 있습니다. 다음은 DAC 채널입니다.

DAC1 (GPIO25)

DAC2 (GPIO26)

□ RTC GPIO

ESP32에는 RTC GPIO가 지원됩니다. RTC 저전력 하위 시스템으로 라우팅된 GPIO는 ESP32가 최대 절전 모드일 때 사용할 수 있습니다. 이러한 RTC GPIO는 ULP(초저전력) 보조 프로세서가 실행 중일 때 ESP32를 깊은 절전 모드에서 깨우는 데 사용할 수 있습니다. 다음 GPIO를 외부 웨이크업 소스로 사용할 수 있습니다.

RTC_GPIO0(GPIO36)

RTC_GPIO3(GPIO39)

RTC_GPIO4(GPIO34)
RTC_GPIO5(GPIO35)
RTC_GPIO6(GPIO25)
RTC_GPIO7(GPIO26)
RTC_GPIO8(GPIO33)
RTC_GPIO9(GPIO32)
RTC_GPIO10(GPIO4)
RTC_GPIO11(GPIO0)
RTC_GPIO12(GPIO2)
RTC_GPIO13(GPIO15)
RTC_GPIO14(GPIO13)
RTC_GPIO15(GPIO12)
RTC_GPIO16(GPIO14)
RTC_GPIO17(GPIO27)

□ PWM

ESP32 LED PWM 컨트롤러에는 16개의 독립적인 채널이 있어 다양한 속성을 가진 PWM 신호를 생성하도록 구성할 수 있습니다. 출력으로 작동할 수 있는 모든 핀은 PWM 핀으로 사용할 수 있습니다(GPIO 34~39는 PWM을 생성할 수 없음). PWM 신호를 설정하려면 코드에서 다음 매개변수를 정의해야 합니다.

신호의 주파수;
듀티 사이클;
PWM 채널;
신호를 출력하려는 GPIO입니다.

□ I2C

ESP32에는 2개의 I2C 채널이 있으며 모든 핀을 SDA 또는 SCL로 설정할 수 있습니다. Arduino IDE와 함께 ESP32를 사용할 때 기본 I2C 핀은 다음과 같습니다.

GPIO 21(SDA)
GPIO 22(SCL)

□ SPI

기본적으로 SPI에 대한 핀 매핑은 다음과 같습니다.

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

□ 인터럽트

모든 GPIO는 인터럽트로 구성할 수 있습니다.

□ 부팅 시 핀 높음

일부 GPIO는 상태를 HIGH로 변경하거나 부팅 또는 재설정 시 PWM 신호를 출력합니다. 즉, 이러한 GPIO에 연결된 출력이 있는 경우 ESP32가 재설정되거나 부팅될 때 예기치 않은 결과를 얻을 수 있습니다.

GPIO 1, GPIO 3, GPIO 5

~~GPIO 6 ~ GPIO 11(ESP32 통합 SPI 플래시 메모리에 연결 — 사용하지 않는 것이 좋습니다).~~

GPIO 14, GPIO 15

□ 활성화(KO)

활성화(EN)는 3.3V 레귤레이터의 활성화 핀입니다. 풀업되어 있으므로 접지에 연결하여 3.3V 레귤레이터를 비활성화하십시오. 이는 예를 들어 푸시 버튼에 연결된 이 핀을 사용하여 ESP32를 다시 시작할 수 있음을 의미합니다.