

Raspberry Pi

간편하게 사용하는

# 라즈베리파이 입문

## Table of Contents

---

# 오늘의 목표

01. 라즈베리파이 하드웨어 이해하기
02. 환경설정
03. GPIO
04. Python으로 센서 데이터 다루기

# 라즈베리파이4

영국의 Raspberry Pi Foundation에서  
교육용 프로젝트의 일환으로 개발된 저가형 SBC

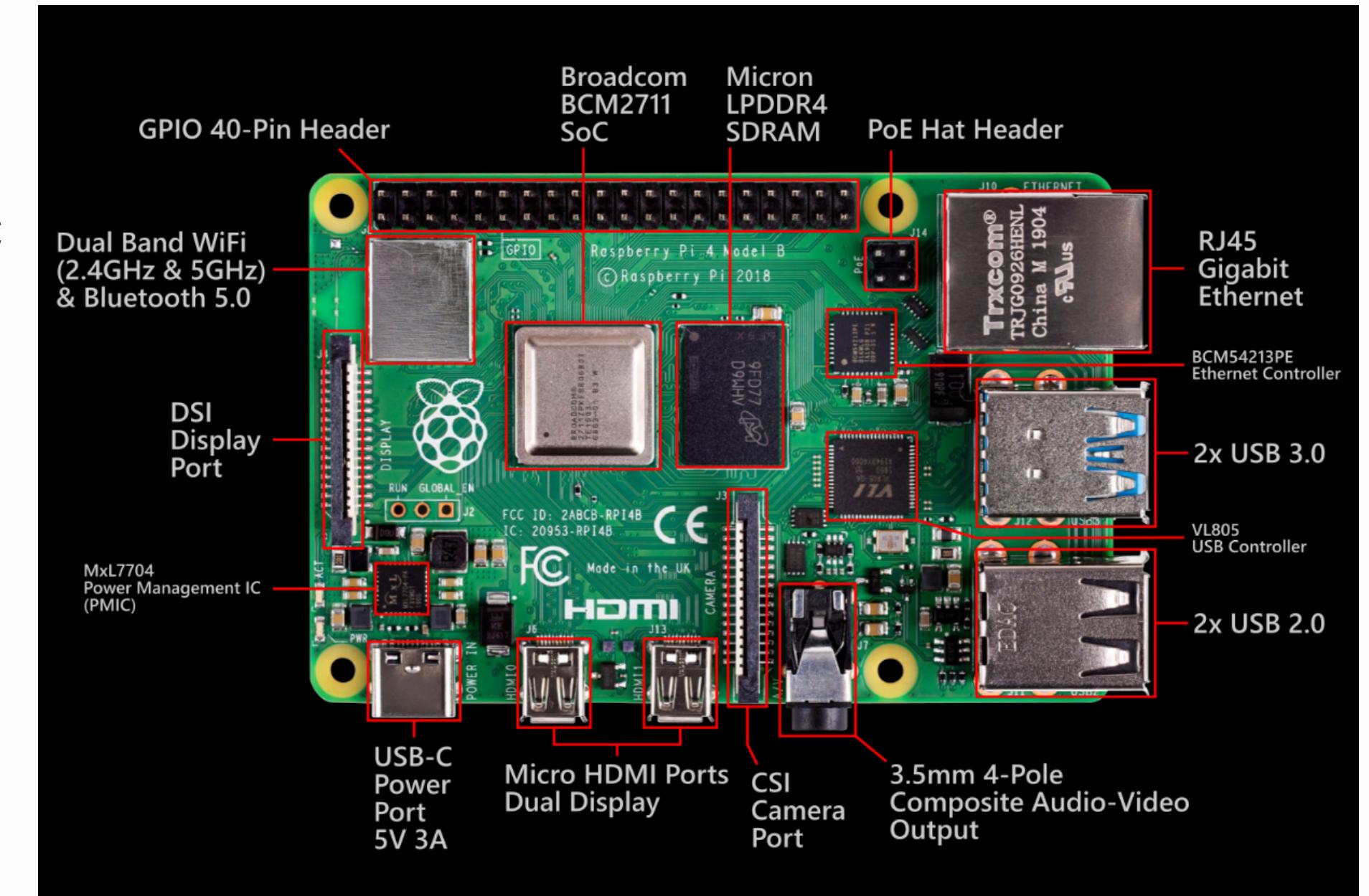
**cpu** ARM Cortex-A72 / 4 Core

**ram** 1, 2, 4, 8 GB

**storage** 최대 2 TB

- 용도**
- 교육, 취미 프로젝트, 홈 서버 구축
  - 스마트 홈 자동화, 로봇 제어

**USB** USB 3.0 2포트, USB 2.0 2포트



# SBC (Single Board Computer)

SBC란?

단일 PCB에 컴퓨터의 모든 기능을 통합한  
완전한 컴퓨터 시스템

소형, 저전력, 고신뢰성

IoT 디바이스 및 스마트 홈 제어

산업 자동화 및 기계 제어

키오스크, ATM, 슬롯머신

미디어 센터 및 서버

주요 용도

RPI 외의  
SBC

NVIDIA Jetson, ODROID, ROCK Pi,  
LattePanda, Orange Pi 등등

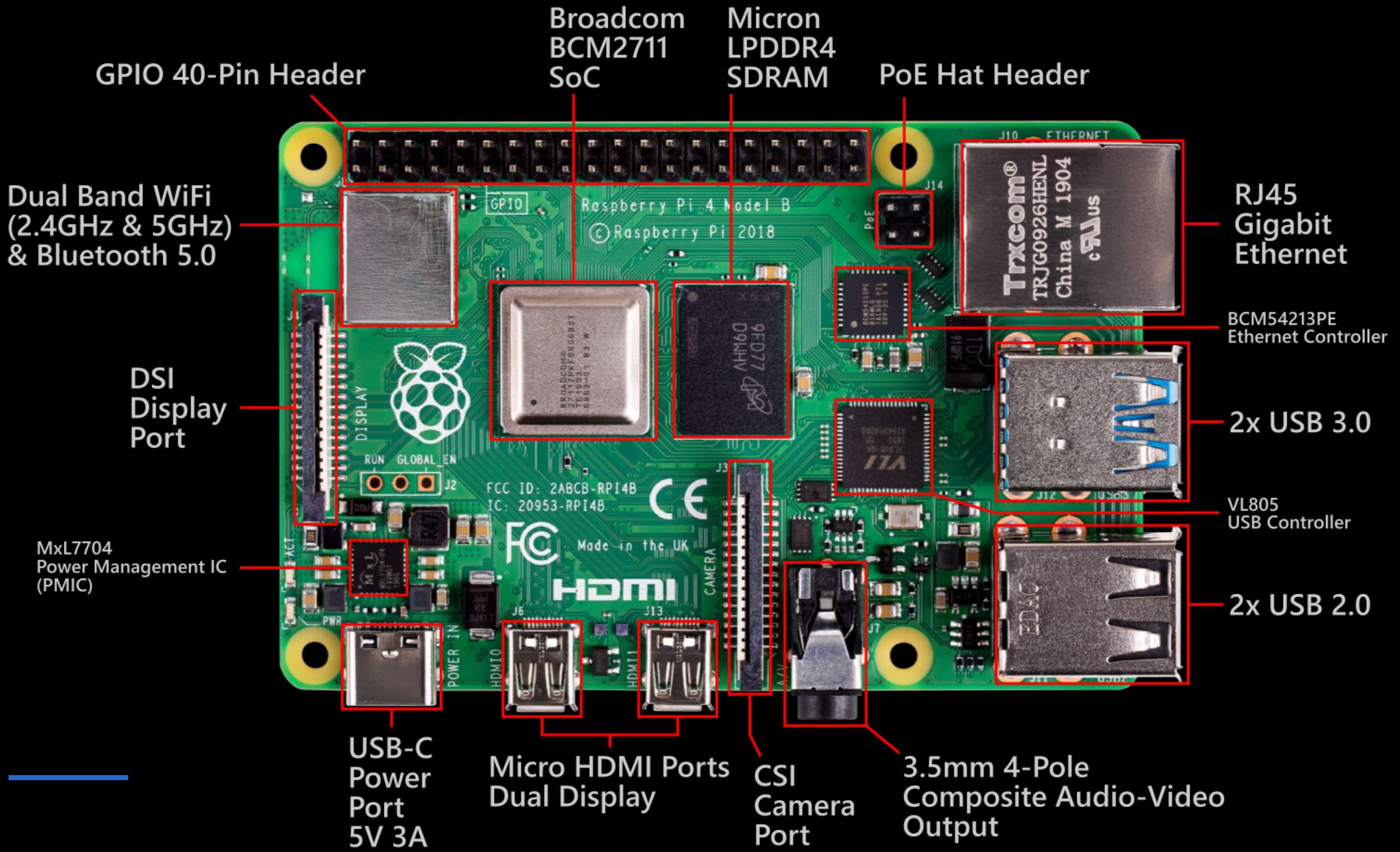


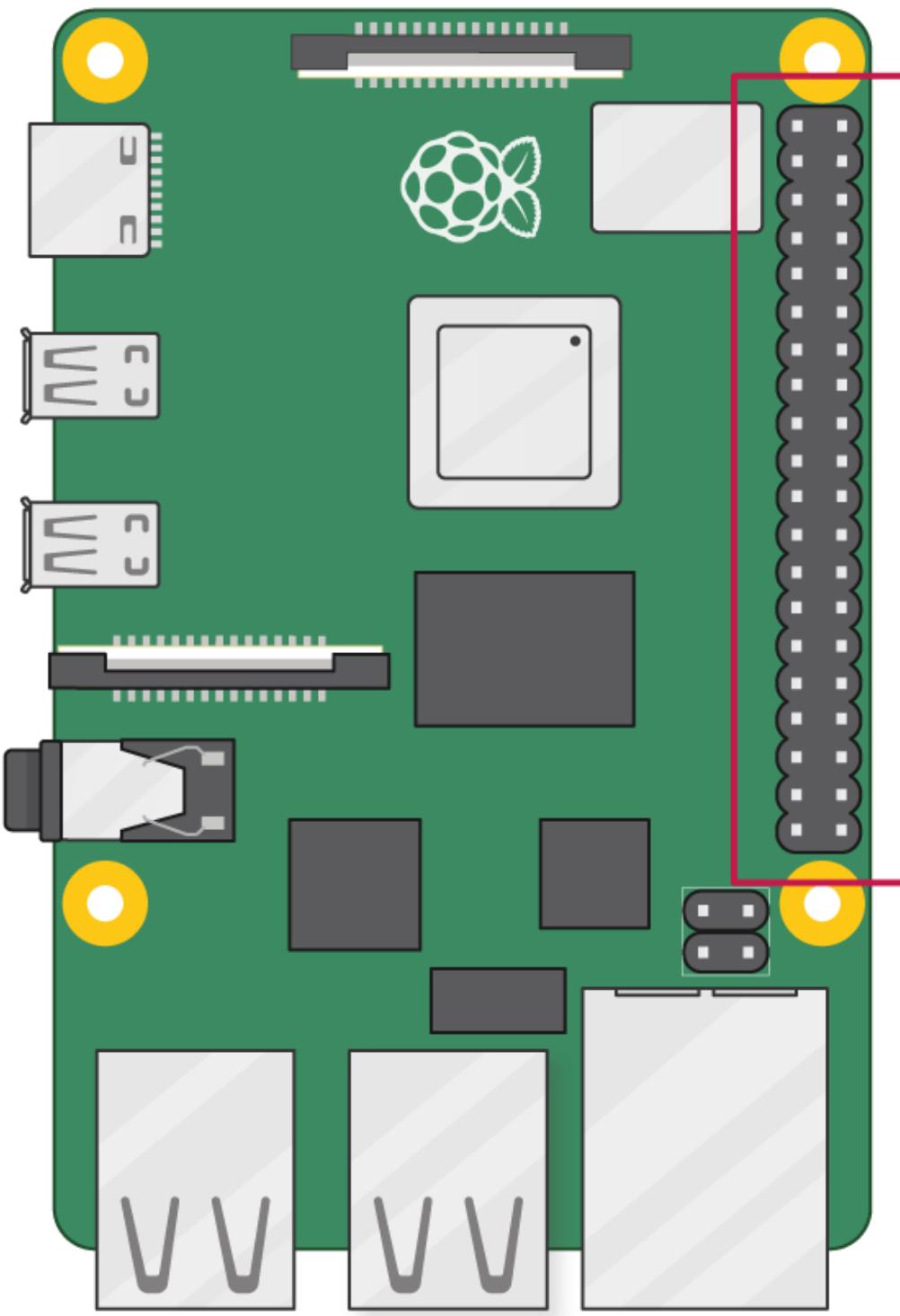
# CPU 아키텍처

RISC	CISC
<ul style="list-style-type: none"><li>작고 단순한 명령어 세트 (보통 100개 이하)</li><li>각 명령어는 1클럭 사이클에 실행</li><li>고정 길이 명령어 (디코딩 간단)</li><li>저전력 소비</li><li>발열 적음</li><li>레지스터 중심 연산 (메모리 접근 최소화)</li></ul>	<ul style="list-style-type: none"><li>크고 복잡한 명령어 세트 (수백 개)</li><li>한 명령어가 여러 작업 수행 가능</li><li>가변 길이 명령어 (1~15바이트)</li><li>메모리 직접 조작 가능</li><li>복잡한 작업을 적은 명령어로 처리</li><li>코드 크기 감소</li></ul>

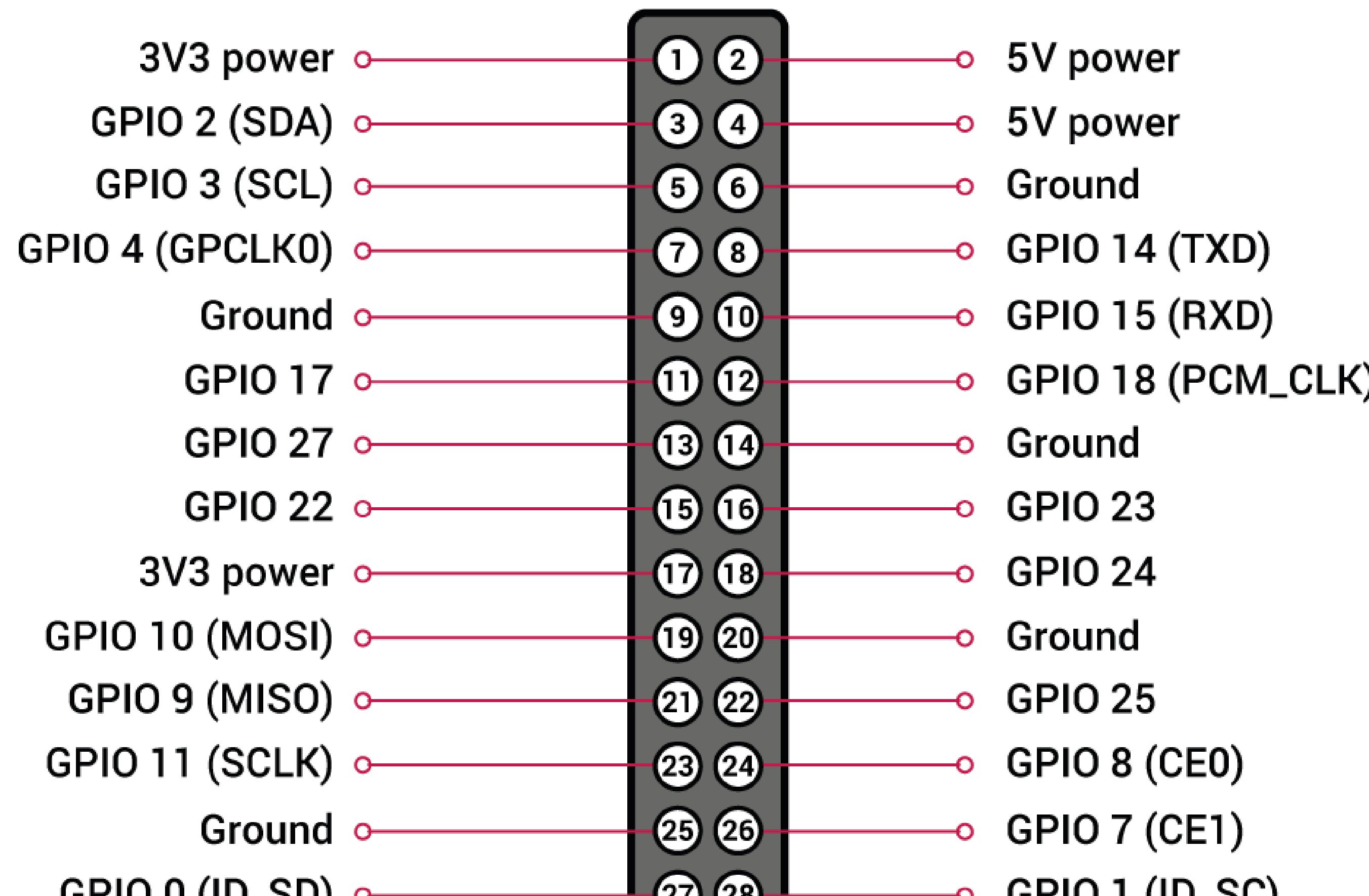
# CPU 아키텍처

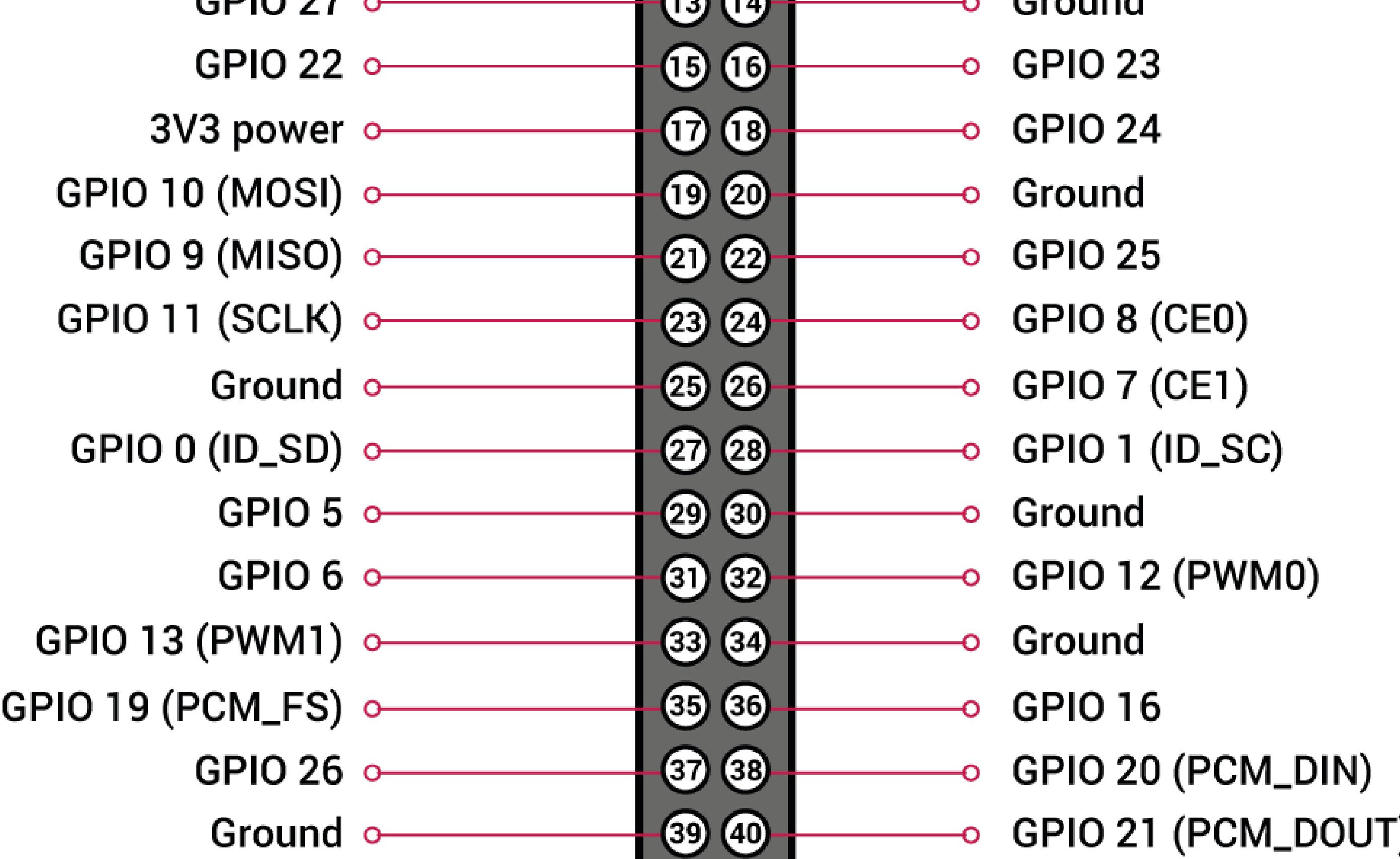
	RISC	CISC
전력 소비	0.5 ~ 15 W	15 ~ 125 W
발열	낮음	높음
가격	낮음	높음
크기	소형	대형
소프트웨어	Linux, 오픈소스 중심	Linux, windows등 풍부함



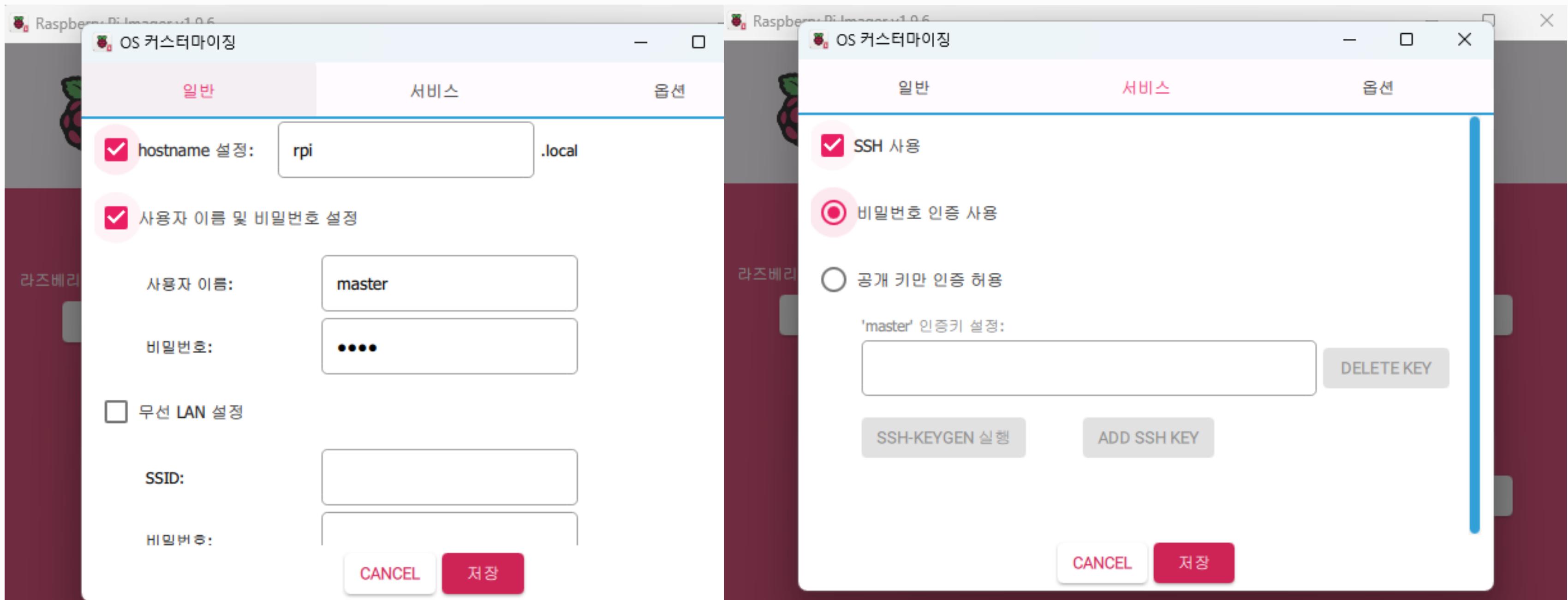


3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)





# 환경설정



# 환경설정

The screenshot shows a Windows PowerShell window titled "master@rpi: ~". The window displays the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

PS C:\Users\kybpl> ssh master@192.168.0.4
The authenticity of host '192.168.0.4 (192.168.0.4)' can't be established.
ED25519 key fingerprint is SHA256:RK5SHC6n3rFfTGfRv8D4xcg1hoPPlay6K21vdwAUJDT0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.4' (ED25519) to the list of known hosts.
master@192.168.0.4's password:
Linux rpi 6.12.47+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.47-1+rpt1 (2025-09-16) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
master@rpi:~ $ uname -a
Linux rpi 6.12.47+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.47-1+rpt1 (2025-09-16) aarch64 GNU/Linux
master@rpi:~ $ |
```

# GPIO로 하드웨어 제어하기

```
mkdir my_dht_project
```

```
cd my_dht_project
```

```
python3 -m venv .venv
```

```
source .venv/bin/activate
```

```
pip install adafruit-circuitpython-dht
```

```
pip install RPi.GPIO
```

# Python으로 센서 데이터 다루기

```
import adafruit_dht
```

```
import board
```

```
dht = adafruit_dht.DHT11(board.D2)
```

```
print(f"temperature: {dht.temperature}\nhumidity: {dht.humidity}")
```

# Python Flask Web Server

```
from flask import Flask  
app = Flask(__name__)  
  
@app.route('/')  
def asdfasdfasdasdf():  
    return 'Hello World!'  
  
> export FLASK_APP=pybo  
> flask run
```

# 과제

1. 라즈베리파이 os 설치하기
2. 라즈베리파이 ssh 연결하기
3. 라즈베리파이에 flask 서버 만들고  
연결 확인하기