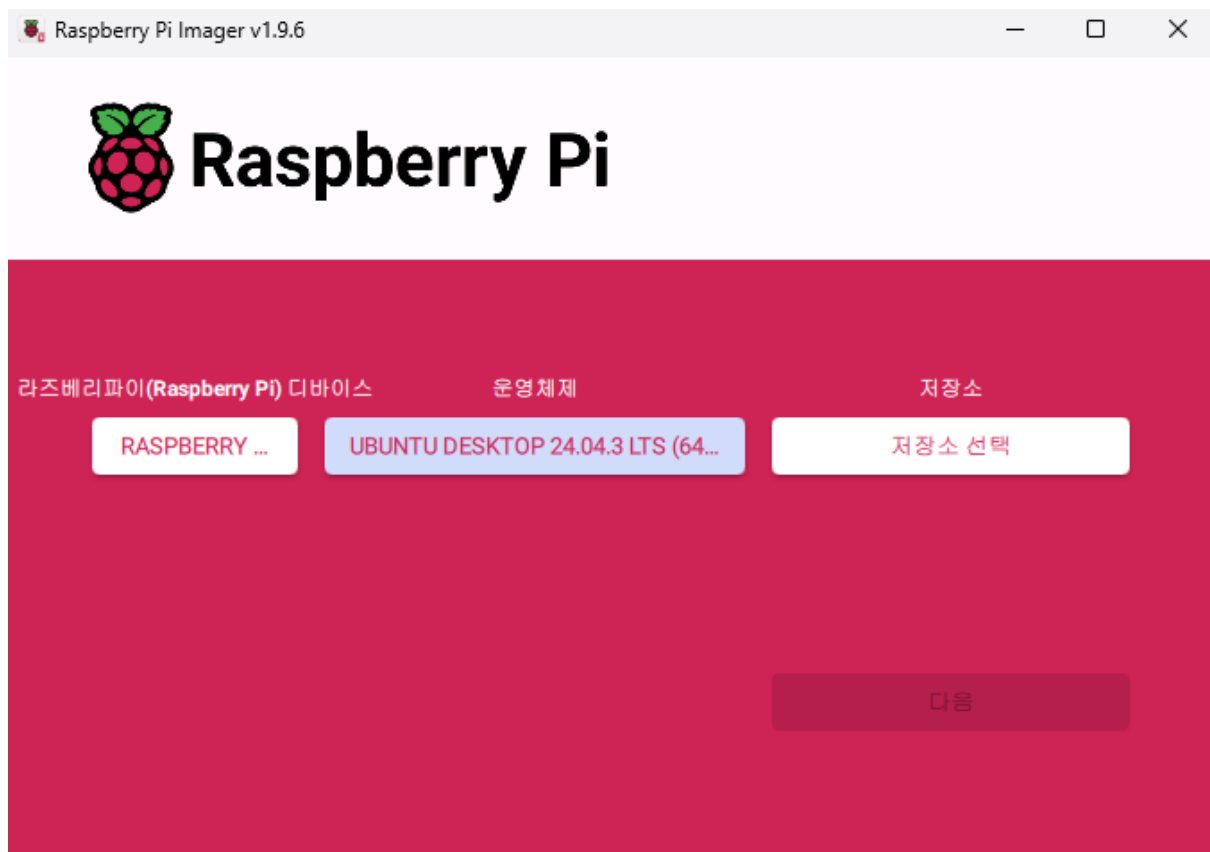


# MQTT 판단 서버 S/W 제작 매뉴얼

## 1. Raspberry Pi 5에서 OS 설치하는 방법.

1-1. <https://www.raspberrypi.com/software/> 사이트에서 Raspberry Pi Imager 설치하기.

1-2. 디바이스 선택에서 [Raspberry Pi 5] 선택 후 운영체제는 [Ubuntu Desktop 24.04.3 LST (64-bit)] 선택하고, 저장소는 SD 카드에다가 넣기.



## 2. Python 3.10.14 설치 (pyenv 방식)

- 터미널에 아래와 같은 코드들을 차례대로 입력하기.

→ `sudo apt install -y make build-essential libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev git`

패키지 이름	용도 설명
<code>make</code> , <code>build-essential</code>	컴파일 도구 (gcc 포함)
<code>libssl-dev</code>	HTTPS, SSL 인증 처리
<code>zlib1g-dev</code> , <code>libbz2-dev</code>	압축 라이브러리 지원
<code>libreadline-dev</code>	CLI 편의 기능 (예: 화살표 키 이력)
<code>libsqlite3-dev</code>	SQLite 내장 DB 지원
<code>tk-dev</code>	Tkinter GUI
<code>libffi-dev</code> , <code>liblzma-dev</code>	외부 라이브러리 지원
<code>git</code> , <code>curl</code> , <code>wget</code>	pyenv 설치용 도구

→ `nano ~/.bashrc` # .bashrc 열기

→ .bashrc를 열고 나서 `export PATH="$HOME/.pyenv/bin:$PATH"` 밑에 코드 치기

- `eval "$(pyenv init --path)"`
- `eval "$(pyenv init -)"`
- `eval "$(pyenv virtualenv-init -)"`

\* 3가지 코드를 친 후 저장 및 종료 후 `exec "$SHELL"`로 셸 재시작

→ `pyenv --version` # 정상 작동 확인 (pyenv 2.5.x 또는 비슷하게 뜨면 성공)

→ `pyenv install 3.10.14`

→ `pyenv global 3.10.14` # 시스템 전체에서 기본 Python 지정

\* global에서 오류나면 `mkdir -p ~/.pyenv/cache` 후 `cd ~/.pyenv/cache`로 이동하기.

\* `wget https://ftp.gnu.org/gnu/readline/readline-8.1.tar.gz` # 수동으로 다운

\* `pyenv install 3.10.14` # 설치 재시도 하면 해결

\* 그 이후 `pyenv global 3.10.14`로 다시 지정

### 3. ARM64용 Miniforge3 설치

- 터미널에 아래와 같은 코드들을 차례대로 입력하기.

→ wget <https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-aarch64.sh>

→ bash Miniforge3-Linux-aarch64.sh

→ 설치 중에 "Do you want to run conda init"? 뜨면 Yes 누르고 설치 완료 후  
exec "\$SHELL" 치면 된다.

→ conda create -n mqtt-env python=3.10 # mqtt-env는 이름으로 원하는걸로 지정

→ conda activate mqtt-env

→ pip install paho-mqtt firebase-admin opencv-python

### 4. Mosquitto 설치하기 : MQTT 프로토콜 기반의 Broker 서버

- 터미널에 아래와 같은 코드들을 차례대로 입력하기.

→ sudo apt upgrade

→ sudo apt update # 업데이트

→ sudo apt install -y mosquitto mosquitto-clients # Mosquitto 설치

→ sudo systemctl status mosquito # 서비스 실행 확인

→ sudo systemctl start mosquito # Mosquitto 시작

→ sudo systemctl enable mosquito # Mosquitto 활성화

→ sudo nano /etc/mosquitto/mosquitto.conf # 설정 파일 열기

→ allow\_anonymous true # listener (포트 번호) 아래에 코드 추가

→ max\_keepalive 0 # 코드 맨 밑에 추가 (클라이언트의 keepalive 제한 없음)

\* 수정 후 ctrl + o 으로 저장 ctrl + x 로 종료하기

→ sudo systemctl restart mosquito # Mosquitto 재시작

## 5. 전원 인가 시 conda 환경 및 메인 코드 자동 실행하기.

- 터미널에 아래와 같은 코드들을 차례대로 입력하기.

→ nano ~/MQTTpr/run\_decision.sh      # /home/mqtt/MQTTpr/run\_decision.sh 생성

→ bash로 들어가서 밑에 2가지 코드를 추가하기.

```
* source ~/miniforge3/bin/activate mqtt-env
```

```
* python /home/mqtt/MQTTpr/MQTT_decision_server.py
```

→ chmod +x ~/MQTTpr/run\_decision.sh      # 실행 권한 부여

→ [유닛 파일 만들기]

```
# systemd 서비스 유닛 파일 만들기
```

[Unit]

Description=MQTT 판단서버 자동 실행

After=network.target

[Service]

Type=simple

User=mqtt

WorkingDirectory=/home/mqtt/MQTTpr

ExecStart=/home/mqtt/MQTTpr/run\_decision.sh

Restart=always

[Install]

WantedBy=multi-user.target

→ 이후 sudo systemctl daemon-reload, sudo systemctl enable decision.service,

sudo systemctl start decision.service 3가지를 친 후 서비스 등록 및 활성화 하기

→ sudo systemctl status decision.service      # 상태 확인하기.

## 6. VS CODE 설치 후 폴더 및 코드

- 아래 링크로 들어가면 Git에 폴더 및 코드들이 있다. 해당 링크에서 코드를 복사하여 Python 파일 및 폴더를 생성하면 된다.

• [https://github.com/elecage/DU\\_AT/tree/d5c91020869304485130639e43d70b475ee7ebed/AuditoryAssist/MQTT\\_Server\\_CODE](https://github.com/elecage/DU_AT/tree/d5c91020869304485130639e43d70b475ee7ebed/AuditoryAssist/MQTT_Server_CODE)

- 완료가 되면 Python MQTT\_decision\_server.py를 실행시켜주면 된다.  
(단, 가상환경은 아까 만든 conda로 가상환경을 지정해줘야 한다.)