

바이트코딩을 이용한 사물인터넷

□ Raspberry Pi Camera

Pi Camera

□ PI Camera

- Camera module for Raspberry Pi



흰색

파란색



렌즈가 부착된 카메라 보드와 연결된 플렉스 케이블로 구성
플렉스 케이블을 라즈베리파이의 CSI 커넥터에 연결

PI Camera

□ PI camera specification

- 5-mega pixel (2592x1944)
- full HD video support: 30 fps H.264 format video
- ports: Still, Video, Preview ports
- Broadcom VideoCore multimedia processor generates still image or video
 - still port: still image (JPEG) capture
 - video port: H.264 format video capture

PI Camera Lab.

□ 카메라 기능: 자동 활성화됨 (최근 OS)

□ 명령어

- **rpicas-still** : for image capture (jpeg 형식)

```
pi@rp84041 ~ $ rpicas-still -t 1000 -o capture.jpg
```

-t : 1000ms (preview 시간)

- **rpicas-vid** : for video capture (h.264 형식)

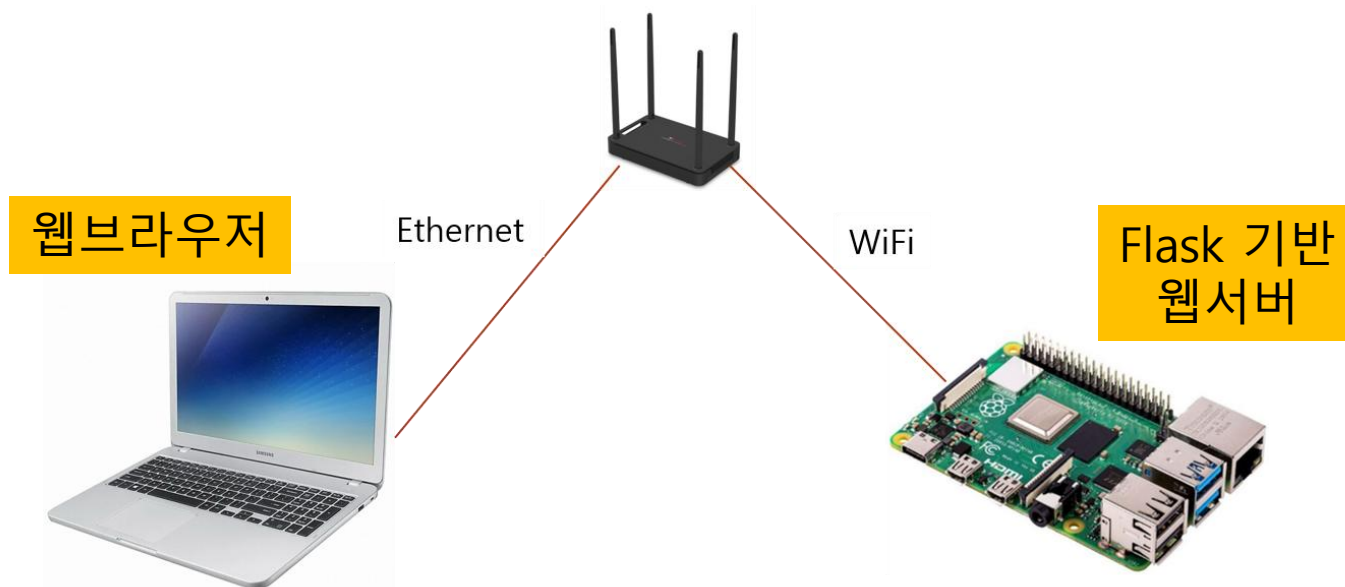
```
pi@rp84041 ~ $ rpicas-vid -t 2000 -o video.h264
```

-t : 1000ms (촬영 시간)

PI Camera Lab.

□ 카메라 실습 프로그램

- 웹 브라우저를 이용한 사진 접근



<http://rp84041:5000/photo>
<http://rp84041:5000/get-list>
<http://rp84041:5000/get-photo?name=<파일명>>

PI Camera Lab.

□ 카메라 실습 프로그램

■ 프롬프트:

로컬네트워크상의 라즈베리파이 rp84041의 pi 사용자 홈폴더에 app, Pictures 라는 폴더들이 있어. 라즈베리파이에서 flask를 사용하여 아래와 같은 url 요청에 대해 처리하는 서버 프로그램을 만들어줘.

/photo url: 카메라로 사진을 찍어 파일로 저장하고, 저장된 파일명을 return,

/get-photo url: 파일명을 인자로 받아 해당 파일을 브라우저로 return,

/get-last url: 마지막에 저장된 사진 파일을 return,

/get-list url: 저장된 사진 파일 리스트를 return

프로그램은 가능한 이해하기 쉽게 간단하게 만들어줘. python 가상환경도 필요없어.

PI Camera Lab.

□ 카메라 실습 프로그램

■ 프롬프트:

-  `app.py (line 1)`: Flask 서버 프로그램
-  `README.md (line 1)`: 라즈베리파이에서 실행하는 방법과 URL 예시

지원 URL:

- `/photo` : 카메라로 촬영 후 파일명 반환
- `/get-photo?filename=...` : 해당 사진 반환
- `/get-last` : 마지막 사진 반환
- `/get-list` : 저장된 사진 파일 목록 JSON 반환

PI Camera Lab.

□ 카메라 실습 프로그램

- 프롬프트:

만들어진 라즈베리파이 서버 프로그램을 pi 계정 아래 app 폴더에 새로운 이름으로 복사해줘.

- 복사 확인: `/home/pi/app/camera_server.py`

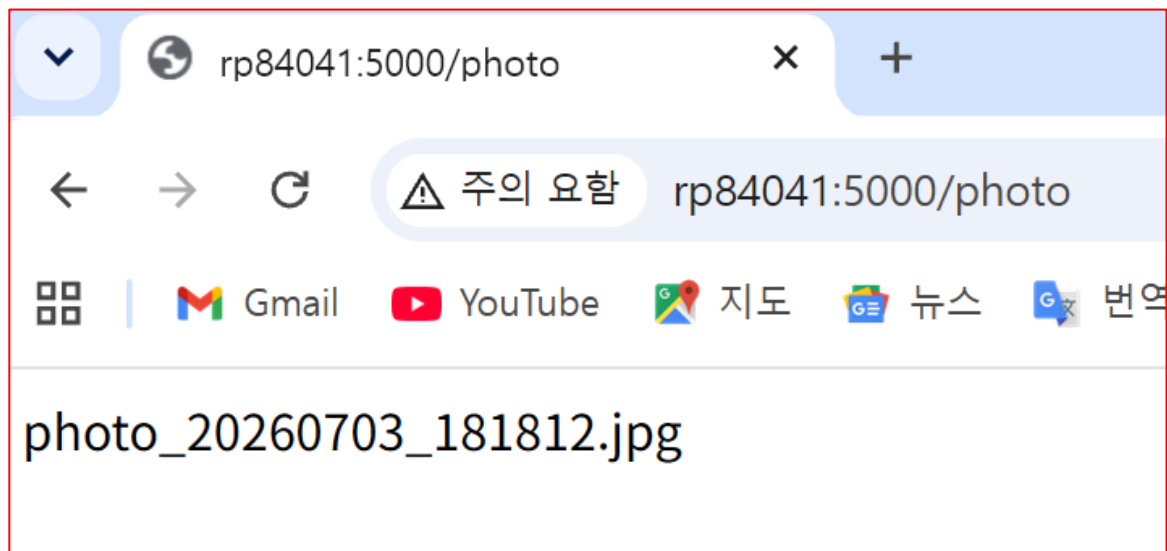
- 서버 실행:

```
cd ~/app
python3 camera_server.py
```

PI Camera Lab.

□ 카메라 실습 프로그램

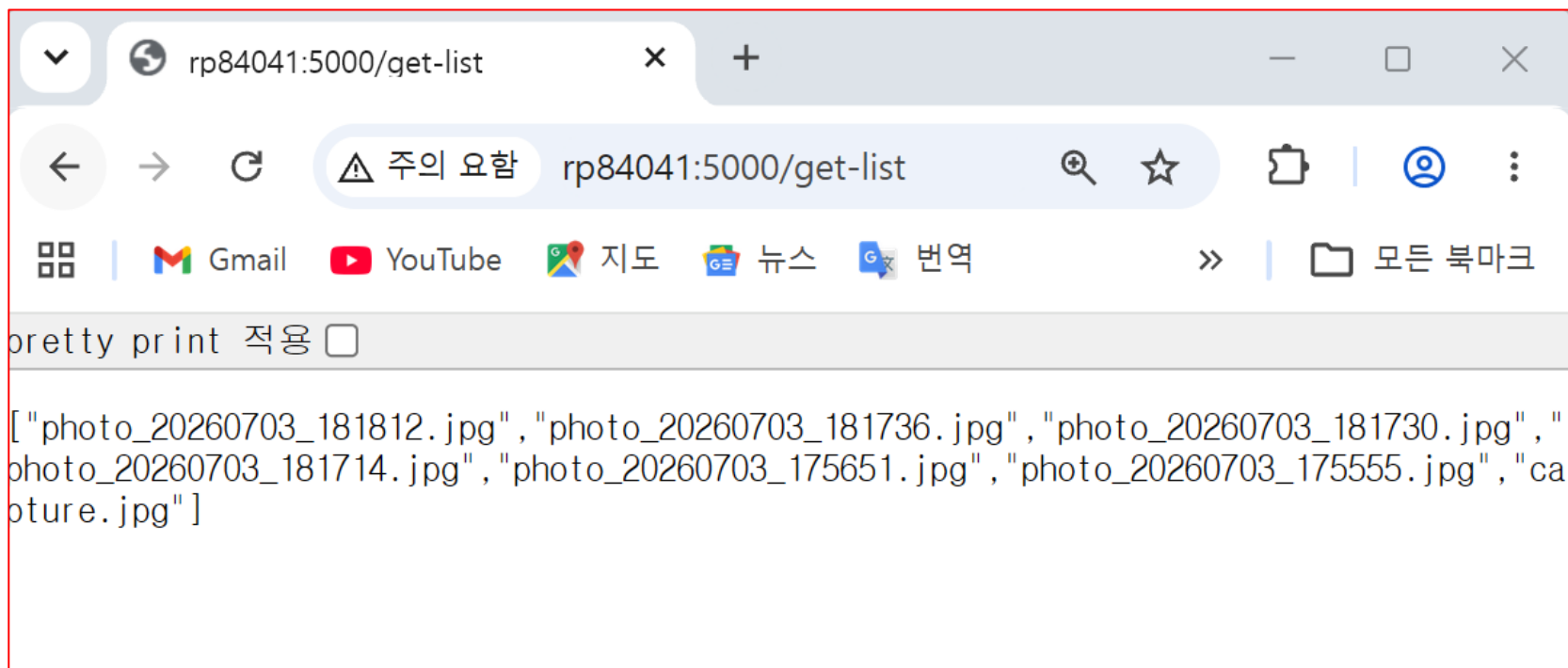
- PC브라우저에서 접속: “/photo” url



PI Camera Lab.

□ 카메라 실습 프로그램

- PC브라우저에서 접속: "/get-list" url



PI Camera Lab.

□ 카메라 실습 프로그램

- PC브라우저에서 접속: `"/get-photo"` url

