

소프트웨어응용학부 201804088 이지훈

소프트웨어응용학부 201904032 김태현

## 목차

- 작품 소개
  - 작품 기능 소개
- 작품 구현 소개
  - 일반 모드
  - 얼굴 인식 모드
- 마무리
  - 보완점, 추가 기능

# 작품 소개

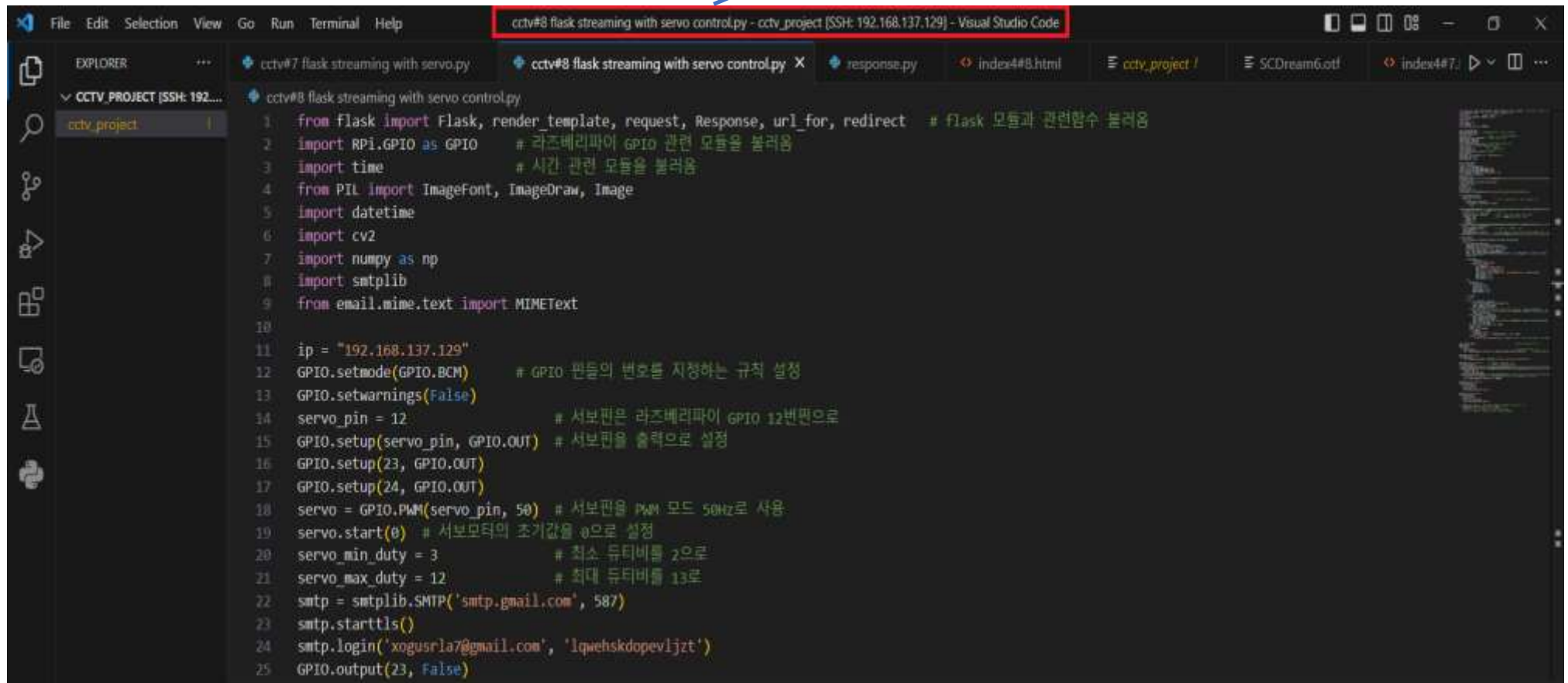
## - 작품 기능 소개

- Visual Studio Code를 활용
- 서보 모터를 통한 카메라 각도 제어
- 캡처, 얼굴 인식, 메일 전송, LED 기능

# 작품 구현 소개

## 작품 구현 - 연결 방식

- SSH를 사용한 VSCode, Raspberry Pi 연결



The screenshot shows the Visual Studio Code editor interface. The title bar indicates the connection: 'cctv#8 flask streaming with servo control.py - cctv\_project [SSH: 192.168.137.129] - Visual Studio Code'. The Explorer sidebar on the left shows the project structure with 'CCTV\_PROJECT [SSH: 192...]' and 'cctv\_project'. The main editor displays a Python script named 'cctv#8 flask streaming with servo control.py'. The script includes imports for Flask, RPi.GPIO, time, PIL, cv2, numpy, smtplib, and email.mime.text. It sets up a servo motor on GPIO pin 12 and configures an SMTP email client. A blue arrow points from the text 'SSH를 사용한 VSCode, Raspberry Pi 연결' to the title bar of the VS Code window.

```
1 from flask import Flask, render_template, request, Response, url_for, redirect # Flask 모듈과 관련함수 불러옴
2 import RPi.GPIO as GPIO # 라즈베리파이 GPIO 관련 모듈을 불러옴
3 import time # 시간 관련 모듈을 불러옴
4 from PIL import ImageFont, ImageDraw, Image
5 import datetime
6 import cv2
7 import numpy as np
8 import smtplib
9 from email.mime.text import MIMEText
10
11 ip = "192.168.137.129"
12 GPIO.setmode(GPIO.BCM) # GPIO 핀들의 번호를 지정하는 규칙 설정
13 GPIO.setwarnings(False)
14 servo_pin = 12 # 서보핀은 라즈베리파이 GPIO 12번핀으로
15 GPIO.setup(servo_pin, GPIO.OUT) # 서보핀을 출력으로 설정
16 GPIO.setup(23, GPIO.OUT)
17 GPIO.setup(24, GPIO.OUT)
18 servo = GPIO.PWM(servo_pin, 50) # 서보핀을 PWM 모드 50Hz로 사용
19 servo.start(0) # 서보모터의 초기값을 0으로 설정
20 servo_min_duty = 3 # 최소 듀티비를 2으로
21 servo_max_duty = 12 # 최대 듀티비를 13로
22 smtp = smtplib.SMTP('smtp.gmail.com', 587)
23 smtp.starttls()
24 smtp.login('xogusr1a7@gmail.com', 'lqwehskdopev1jzt')
25 GPIO.output(23, False)
```

## 작품 구현 - 일반 모드

- 카메라 작동
- 각도 변경
- 화면 캡처



## 작품 구현 - 일반모드

- 서보 모터 각도 제어

- 화면 캡처

- 일반 모드 (CCTV)

- 얼굴 인식 모드

```
@app.route('/')
def index():
    global push_btn
    return render_template('index4#8.html', deg=current_deg, push_btn = push_btn)

@app.route('/video_feed')
def video_feed():
    return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/servo_control')
def servo_control():
    deg = request.args.get('deg')
    deg = int(deg)
    if deg < 0: deg = 0
    elif deg > 180: deg = 180
    deg = set_servo_degree(deg)
    return render_template('index4#8.html', deg=deg)

@app.route('/push_capture')
def push_capture():
    global is_capture
    is_capture = True
    return redirect(url_for('index'))

@app.route('/push_switch')
def push_switch():
    global push_btn
    push_btn = not push_btn
    return redirect(url_for('index'))
```



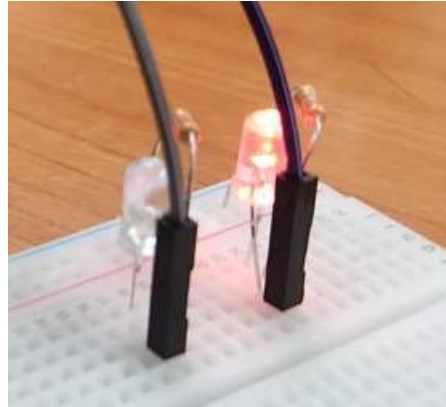
## 작품 구현 - 얼굴 인식 모드

- 얼굴 인식 시 카메라 작동
- LED 점등

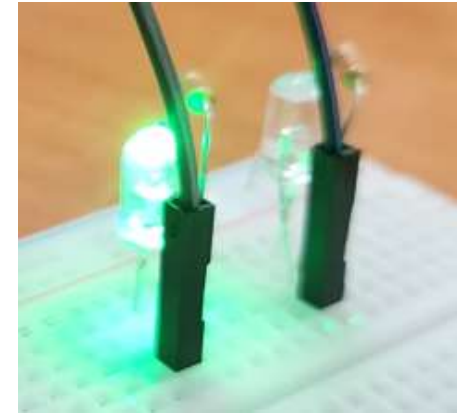


## 얼굴 인식 모드 - 얼굴 인식 시

- 빨간 LED 점등



- 얼굴 미 인식 경우  
초록 LED 점등



- 메일 전송

			xogusrla7@gmail.com	intruder alert!!			06.09 18:57
			xogusrla7@gmail.com	intruder alert!!			06.09 18:57
			xogusrla7@gmail.com	intruder alert!!			06.09 18:54
			xogusrla7@gmail.com	intruder alert!!			06.09 18:54
			xogusrla7@gmail.com	intruder alert!!			06.09 18:54

## 작품 구현 - 얼굴 인식 모드

- 얼굴 인식 시

```
if len(faces):
    is_detected = True
    cnt_record = max_cnt_record
    if not is_mail_sent:
        msg = MIMEText('intruder alert!!')
        msg['Subject'] = 'intruder alert!!'
        msg['To'] = 'xogusrla0097@naver.com'
        smtp.sendmail('xogusrla7@gmail.com', 'xogusrla0097@naver.com', msg.as_string())
        is_mail_sent = True
        GPIO.output(24, True)
        GPIO.output(23, False)
```

- 얼굴 미 인식 시

```
else:
    cnt_record -= 1
    if cnt_record == 0:
        is_detected = False
        is_mail_sent = False
        GPIO.output(24, False)
        GPIO.output(23, True)
```

## 마무리

### ● 보완점

- 메일 한 번만 보내기
- 서보 모터 원활한 제어

### ● 추가 할 점

- 영상 녹화 기능
- 진동 센서 활용