

# 컴퓨터네트워크

- 08. Philips Hue -

제 출 일	2018.11.09
학 과	컴퓨터공학과
학 번	201402391
이 름	이 병 만

## ※ 과제 목표 및 해결 방법

### · 과제 목표

- 이전에 구현한 HTTP Server + Hue 제어 웹 서비스 개발
  - ▶ 클라이언트 -> 서버 : hue\_controller.html 요청
  - ▶ 서버 : 사용자 입력에 따라 Hue 제어
  - ▶ 브라우저 접근이 되어야함
  - ▶ 실행 화면을 영상으로 찍어서 youtube에 업로드 → 링크를 보고서에 첨부

### · 해결 방법

#### 1. Server

```
from socket import *
import os
import sys
from threading import Thread
import traceback
from phue import Bridge
```

[그림1. use\_module]

→ 구현에 필요한 모듈을 import한다. philips hue를 사용하기 위해서 모듈을 추가해주었다.

```
def server_excute(clnt_sock):
    data = clnt_sock.recv(1500)
    url = (data.decode()).split(' ')
    file_name = url[1]
    print(file_name)

    if len(file_name.split('?')) > 1 :
        query = file_name.split('?')[1]
        data = query.split('&')
        hue_num = data[0].split('=')[0]
        light = data[1].split('=')[1]
        color_x = data[2].split('=')[1]
        color_y = data[3].split('=')[1]
        hue(hue_num, light, color_x, color_y)

    # if method = GET
    if url[0] == 'GET':
        if os.path.isfile('.'+file_name):
            f = open('.' + file_name, 'rb')
            read_data = f.read(1024)
            file_size = str(os.path.getsize('.' + file_name)).encode() + b'\n\n'
            clnt_sock.send(is_file + file_size + read_data)

            while True:
                read_data = f.read(1024)
                if not read_data:
                    break
                clnt_sock.send(read_data)
            f.close()
            clnt_sock.close()
```

[그림2. server\_excute function]

→ 서버가 실행하는 함수이다. 소켓을 생성할 때마다 소켓을 인자로 받아서 처리해준다. 클라이언트로부터 받은 url을 파싱하는데 '?'를 기준으로 파싱했을 때 배열의 크기가 1보다 클 경우에는 쿼리문도 있다는 것으로 파악해서 쿼리문을 파싱해줄 수 있도록 구현하였다. 파싱한 데이터를 저장해서 hue함수를 호출한다.

```
def hue(hue_num, light, color_x, color_y):
    b = Bridge('192.168.1.139')
    b.connect()

    lights = b.lights
    print(lights)
    light = int(light)
    color_x = float(color_x)
    color_y = float(color_y)

    if hue_num == 'on1':
        lights[0].on = True
        lights[0].brightness = light
        lights[0].xy = [color_x, color_y]
    elif hue_num == 'off1':
        lights[0].on = False

    if hue_num == 'on2':
        lights[1].on = True
        lights[1].brightness = light
        lights[1].xy = [color_x, color_y]
    elif hue_num == 'off2':
        lights[1].on = False

    if hue_num == 'on3':
        lights[2].on = True
        lights[2].brightness = light
        lights[2].xy = [color_x, color_y]
    elif hue_num == 'off3':
        lights[2].on = False
```

[그림3. hue]

➔ hue를 작동시키는 함수이다. 브릿지로 공유기의 IP를 연결해주고 인자로 받은 값으로 hue를 컨트롤한다.

#### ※ 실행 결과

실행 결과는 링크 첨부로 대체하였습니다.

<https://www.youtube.com/watch?v=36XAtC6ldD8&feature=youtu.be>