

2021-1 Capstone Design

12주차 캡스톤 진행 상황

DeePLY

Capstone Design Team : 5

17011885 박세정

17011869 이혜인

17011757 박미희

프로젝트 목적

1

WebRTC의 P2P 연결의 네트워크 이해하기

로봇의 웹캠을 통한 영상을 WebRTC를
이용하여 실시간으로
web 내에서 스트리밍 구현하기

2

3

실시간 영상 딜레이 시간 축소하기

Web 내에서 로봇의 동작 제어하기

4

프로젝트 목적 & 전체 진행상황

1. P2P 연결의 네트워크 이해하기

Web Real-Time Communication (WebRTC)



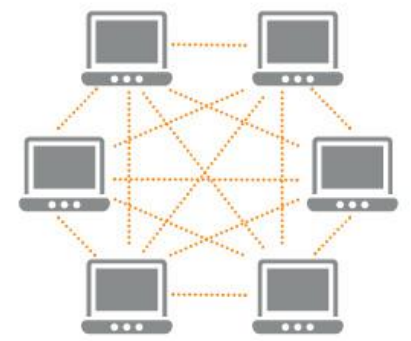
웹 애플리케이션과 사이트가 중간자 없이 브라우저 간에 오디오나 영상 미디어를 포착하고 통신할 수 있도록 설계된 API

P2P (Peer to Peer)

- 개인 컴퓨터(Peer)들이 서로 연결되어 파일을 전송하는 시스템
- 개인 컴퓨터들이 서버이자 클라이언트가 되는 것
- 멀티미디어 데이터를 실시간으로 교환



Server-Based



P2P

프로젝트 목적 & 전체 진행상황

2. 로봇의 웹캠을 통한 영상을 WebRTC를 이용하여 실시간으로 Web 내에서 스트리밍 구현하기

- UV4L을 통해 실시간 영상 스트리밍 진행 (진행 완료)
- WebRTC Datachannel을 통해 실시간 스트리밍 영상을 다른 디바이스의 브라우저로 전송

3. 실시간 영상 딜레이 시간 축소하기

- UV4L 서버를 열때 WebCam의 UVC디바이스의 고유 id를 직접 지정(진행 완료)

프로젝트 목적 & 전체 진행상황

4. web 내에서 로봇의 동작 제어하기

- RTCBot API 이용하여 서로 다른 디바이스와 브라우저로 로봇에게 보내는 명령어 주고 받기
(진행 완료)
- 받은 명령어를 통해 Python GPIO라이브러리를 사용하여 라즈베리파이의 GPIO 핀 원격 제어
(12주차 진행 완료)
- UV4L socket을 이용한 datachannel 구현하여 로봇 제어(12 주차 진행완료)

12주차 목표

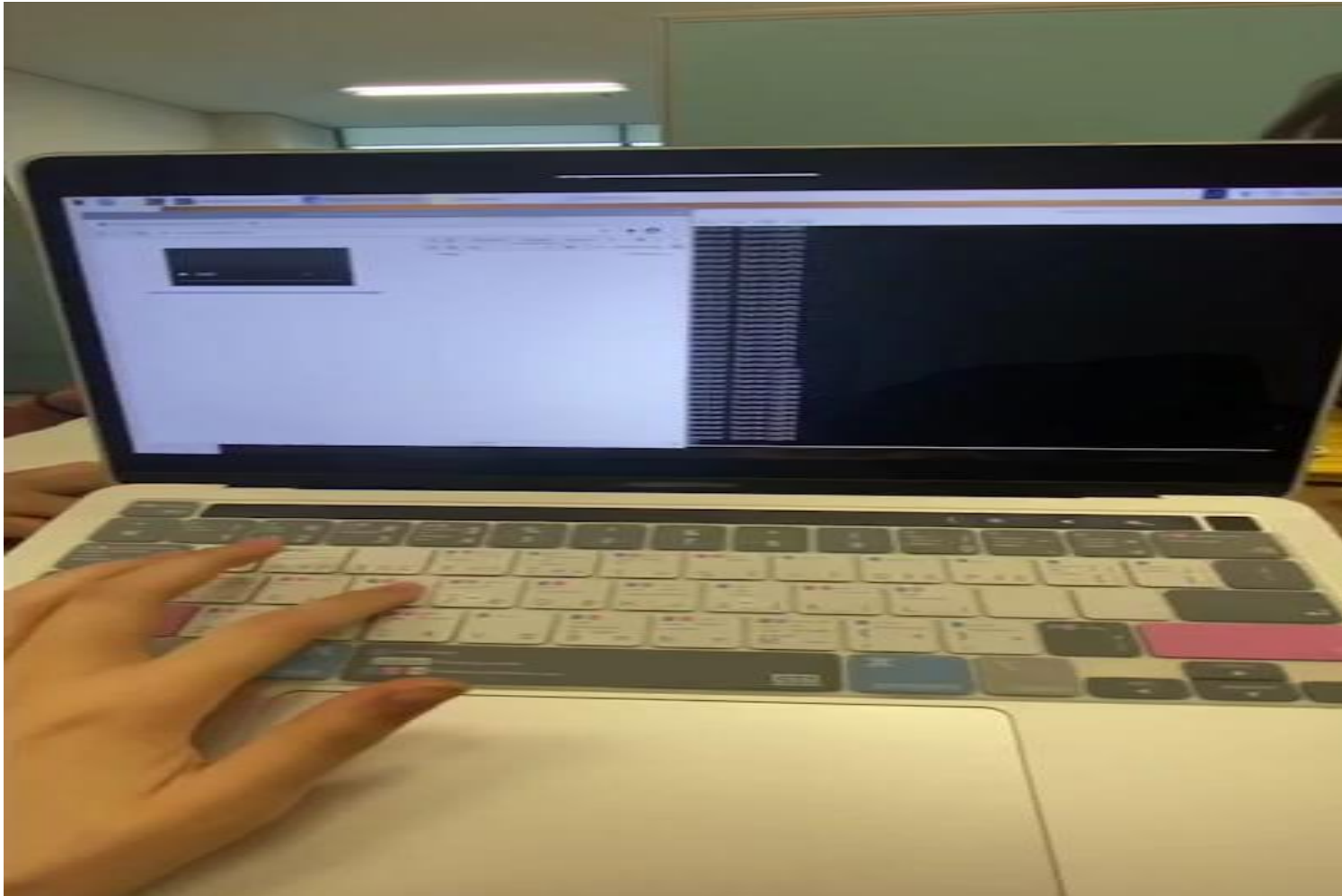
1. 받은 명령어를 통해 로봇 제어하기
2. Custom web app 개선해 영상 스트리밍



1. 받은 명령어를 통해 로봇 제어하기 ✓
2. UV4L socket을 이용한 datachannel 구현하여 로봇 제어 ✓
3. Custom web app 개선해 영상 스트리밍 !

지난 11주차 진행 과정

WebRTC를 통해 두 개의 디바이스로 data channel 구현



- RTCBot API 이용 (로봇 원격 제어 가능)
- RPi - Browser WebRTC 연결
python, javascript

12주차 진행 과정

Data channel로 받은 명령어를 통해 로봇 원격 제어

- RTCBot API 이용하여 받은 명령어를 통해 로봇을 각 명령어에 맞게 제어하기 위한 과정
- **Python GPIO라이브러리**를 사용하여 라즈베리파이의 GPIO 핀 제어

```
9 conn = RTCConnection()
10 conn.video.putSubscription(cam)
11
12 ### 참고: deeply_robot.c
13 import RPi.GPIO as gpio
14 gpio.setmode(gpio.BCM)
15 # gpio.setwarnings(False)
16 import move_func as move
17
18 ## BCM number
19 IN1 = 19 # 오른쪽 앞으로
20 IN2 = 16 # 오른쪽 뒤로
21 IN3 = 21 # 왼쪽 앞으로
22 IN4 = 26 # 왼쪽 뒤로
23
24 gpio.setup(IN1, gpio.OUT, initial=0)
25 gpio.setup(IN2, gpio.OUT, initial=0)
26 gpio.setup(IN3, gpio.OUT, initial=0)
27 gpio.setup(IN4, gpio.OUT, initial=0)
28
29 ###
30 def Motor_Forward():
31     print('GO FORWARD')
32     gpio.output(IN1, 1)
33     gpio.output(IN2, 0)
34     gpio.output(IN3, 1)
35     gpio.output(IN4, 0)
36     ###
37
38 def Motor_Backward():
39     print('GO BACK')
40     gpio.output(IN1, 0)
41     gpio.output(IN2, 1)
42     gpio.output(IN3, 0)
43     gpio.output(IN4, 1)
44
45 def Motor_Left():
46     print('GO LEFT')
47     gpio.output(IN1, 1)
48     gpio.output(IN2, 0)
49     gpio.output(IN3, 0)
50     gpio.output(IN4, 1)
51
52 def Motor_Right():
53     print('GO RIGHT')
54     gpio.output(IN1, 0)
55     gpio.output(IN2, 1)
56     gpio.output(IN3, 1)
57     gpio.output(IN4, 0)
58
59 def Motor_Stop():
60     print('STOP')
61     gpio.output(IN1, 0)
62     gpio.output(IN2, 0)
63     gpio.output(IN3, 0)
64     gpio.output(IN4, 0)
```


12주차 진행 과정

Data channel로 받은 명령어를 통해 로봇 원격 제어



- RTCBot API 이용
-로봇에게 반양
- RPi - Browser WebRTC 연결
(python, javascript)
- **로봇 제어 연결(python)**

12주차 진행 과정

UV4L socket을 이용한 datachannel 구현하여 로봇 제어

```
파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T)
on JS server.js keyboard_robot.py 4 uv4l.py
C: > Users > USER > Desktop > uv4l.py > ...
1 import os
2 import time
3 import RPi.GPIO as GPIO
4 import json
5
6 import socket
7 socket_path='/tmp/uv4l.socket'
8 try:
9     os.unlink(socket_path)
10 except OSError:
11     if os.path.exists(socket_path):
12         raise
13
14 s=socket.socket(socket.AF_UNIX, socket.SOCK_SEQPACKET)
15
16 #####
17 GPIO.setmode(GPIO.BCM) #MOTOR
```

```
File Edit Tabs Help
pi@raspberrypi:~ $ uv4l -k --sched-rr --mem-lock --config-file=
C.conf --driver uvc --driver-config-file=/etc/uv4l/uv4l-uvc.conf
--editable-config-file=/etc/uv4l/uv4l-uvc.conf --device-id 046
<notice> [core] Trying to loading driver 'uvc' from built-in dr
<notice> [core] Loading driver 'uvc' from external plug-in's...
<notice> [driver] Video functionality '' recognized at 046d:082
<notice> [core] Device detected!
modprobe: ERROR: could not insert 'cuse': Operation not permissi
<notice> [core] Registering device node /dev/video0
pi@raspberrypi:~ $ cd /boot/
pi@raspberrypi:/boot $ cd control_deeplly.py
bash: cd: control_deeplly.py: No such file or directory
pi@raspberrypi:/boot $ python control_deeplly.py
socket_path: /tmp/uv4l.socket
awaiting connection....
client_address
('established connection with', '')
I

67 52080 typ host gener
Mid": "", "sdpMLineIndex": 2}
candidate: 3707591233 1 udp webrtc:169
candidate: "candidate:3707591233 1 udp DVLY network-id 3
67 55235 typ host generation 0 ufrag
Mid": "", "sdpMLineIndex": 1}
candidate: 2457773745 1 tcp webrtc:169
67 32809 typ host tcptype passive generation 0 ufrag
work-cost 50", "sdpMid": "", "sdpMLineIndex": 0}
candidate: 2457773745 1 tcp webrtc:169
67 33835 typ host tcptype passive generation 0 ufrag
work-cost 50", "sdpMid": "", "sdpMLineIndex": 2}
candidate: 2457773745 1 tcp webrtc:169
67 52221 typ host tcptype passive generation 0 ufrag
work-cost 50", "sdpMid": "", "sdpMLineIndex": 1}
data: {"type": "answer", "sdp": "v=0\r\no=- 90361041867078
000 webrtc-datachannel 1024\r\n"}
webrtc:309
webrtc:357
webrtc:161
webrtc:124
webrtc:128
webrtc:137
webrtc:707
webrtc:707
webrtc:137
webrtc:707
webrtc:137
webrtc:137
webrtc:137
webrtc:137
```

13 주차 진행 목표

Custom web app 개선해 영상 스트리밍 구현

1. WebRTC API 중 **RTCPeerConnection**(Peer간 스트림(음성, 영상)전송) 사용하여 UV4L WebRTC 브라우저 스트리밍 영상 가져오기
2. 라즈베리파이 USB webcam에서 **Opencv** 사용하여 스트리밍 영상 가져오기