

2021-1 Capstone Design

5주차 캡스톤 진행 상황

DeePLY

Capstone Design Team : 5

17011885 박세정

17011869 이혜인

17011757 박미희

4주차 진행 상황에서 문제점

1. UV4L Streaming server 내에서 웹캠 영상을 실행할 때, Time delay와 lag이 심함.
2. 영상 streaming을 라즈베리파이 컴퓨터 내에서만 실행함.



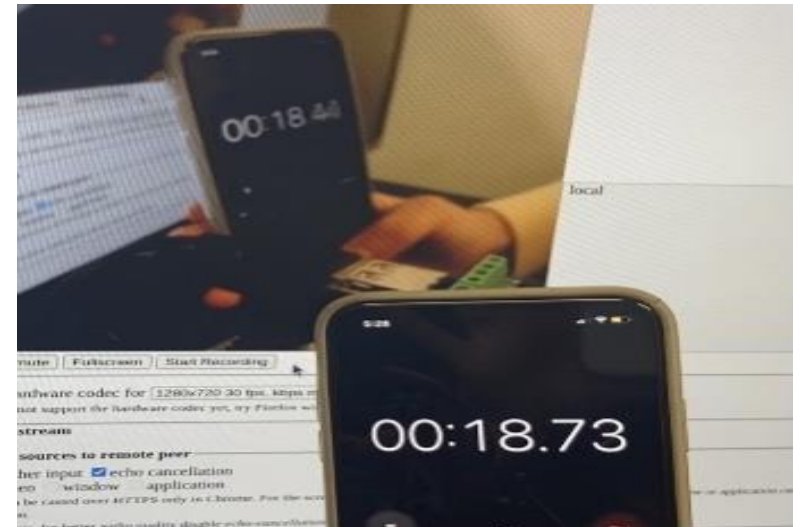
5주차 진행 목표

1. 웹캠 영상의 time delay와 time lag 문제 해결
2. 외부 컴퓨터에서 UV4L Streaming server 연결
3. http -> https 변경
4. WebRTC tutorial 공부

5주차 진행 과정

웹캠 영상의 Time delay와 Time lag 문제 해결하기 위해 시도

1. h264 로 코덱 변경
2. 새로운 공유기 변경
3. 라즈베리파이 변경
4. 해상도 낮추기
5. 공유기 랜선으로 연결하여 streaming test
6. Webrtc server option 변경
7. 650*480 30 fps 에서 가장 최적
8. 웹캠 확인 => 노트북에 연결해본 결과, webcam 문제는 아니었다.



라즈베리파이 변경 이후 time delay는 개선된 듯 하나, 긴 시간동안 streaming 할 경우 time lag 현상은 여전함
=> 6주차 때 개선 방향 더 찾아볼 것

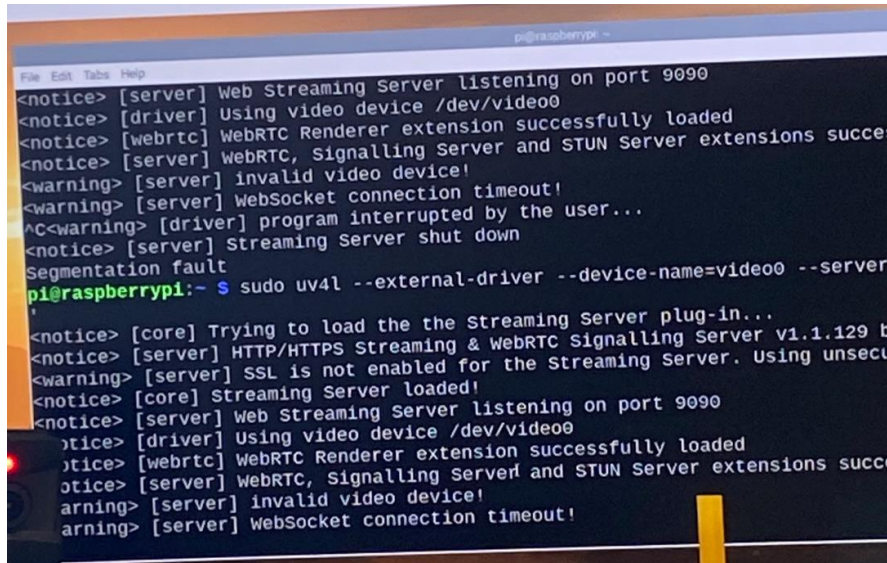
5주차 진행 과정

외부 컴퓨터에서 UV4L Streaming server 연결

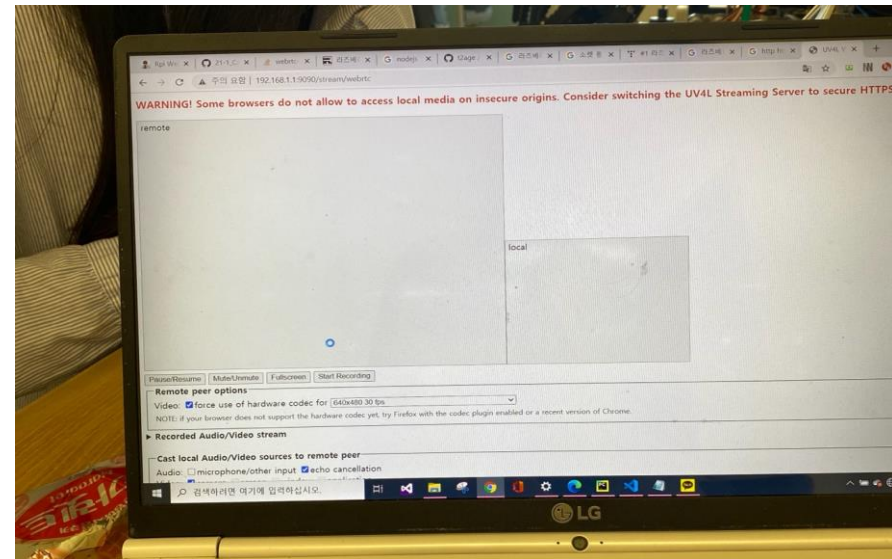
서버 열기 성공 => 기존 서버 열 때 사용하는 명령어에 external 옵션 추가

```
$ uv4l --external-device
```

➡ UV4L 서버는 열렸으나, invalid video device 로 비디오 스트리밍이 불러와지지 않았음



```
pi@raspberrypi:~$ sudo uv4l --external-driver --device-name=video0 --server
<notice> [server] Web Streaming Server listening on port 9090
<notice> [driver] Using video device /dev/video0
<notice> [webrtc] WebRTC Renderer extension successfully loaded
<notice> [server] WebRTC, Signalling Server and STUN Server extensions success
<warning> [server] invalid video device!
<warning> [server] WebSocket connection timeout!
^C<warning> [driver] program interrupted by the user...
<notice> [server] Streaming Server shut down
Segmentation fault
pi@raspberrypi:~$ sudo uv4l --external-driver --device-name=video0 --server
<notice> [core] Trying to load the the Streaming Server plug-in...
<notice> [server] HTTP/HTTPS Streaming & WebRTC Signalling Server v1.1.129 b
<warning> [server] SSL is not enabled for the Streaming Server. Using unsecu
<notice> [core] Streaming Server loaded!
<notice> [server] Web Streaming Server listening on port 9090
<notice> [driver] Using video device /dev/video0
<notice> [webrtc] WebRTC Renderer extension successfully loaded
<notice> [server] WebRTC, Signalling Server and STUN Server extensions succ
<warning> [server] invalid video device!
<warning> [server] WebSocket connection timeout!
```



5주차 진행 과정

http -> https 변경 과정

1. 인증키, 인증서 생성

```
$ openssl genrsa -out selfsign.key 2048 && openssl req -new -x509 -key selfsign.key -out selfsign.crt -sha256
```

2. UV4L-UVC Config 파일 내의 HTTPS 옵션 변경

```
/// HTTPS options:  
server-option = --use-ssl=yes  
server-option = --ssl-private-key-file=/home/pi/selfsign.key  
server-option = --ssl-certificate-file=/hime/pi/selfsign.crt
```

3. Reboot 후 서버 연결

```
uv4l --driver raspicam --server-option '--use-ssl=yes'
```



실패 : NET::ERR_CERT_AUTHORITY_INVALID 에러

5주차 진행 과정

WebRTC tutorial 공부

Google codelab에서 제공하는 webrtc 튜토리얼 보고 코드 이해

<https://codelabs.developers.google.com/codelabs/webrtc-web#3>

Real time communication with WebRTC

31 mins remaining

1 Introduction

2 Overview

3 Get the sample code

4 Stream video from your webcam

5 Stream video with RTCPeerConnection

6 Use RTCDataChannel to exchange data

7 Set up a signaling service to exchange messages

8 Combine peer connection and signaling

9 Take a photo and share it via a data channel

10 Congratulations

Report a mistake

Back

4. Stream video from your webcam

What you'll learn

- Get a video stream from your webcam.
- Manipulate stream playback.
- Use CSS and SVG to manipulate video.

A complete version of this step is in the **step-01** folder.

A dash of HTML...

Add a `video` element and a `script` element to `index.html` in your work directory:

```
<!DOCTYPE html>
<html>

<head>

<title>Realtime communication with WebRTC</title>

<link rel="stylesheet" href="css/main.css" />

</head>

<body>
```

Next

step-02 > js > JS main.js > ...

```
494 // Define action buttons.
205 // 버튼에 action 추가하기
206 const startButton = document.getElementById('startButton');
207 const callButton = document.getElementById('callButton');
208 const hangupButton = document.getElementById('hangupButton');
209
210 // Set up initial action buttons status: disable call and hangup.
211 // 초기 수행 단계 설정: 통화 및 끊기 사용안함
212 callButton.disabled = true;
213 hangupButton.disabled = true;
214
215
216 // Handles start button action: creates local MediaStream.
217 // 시간 단축 작업 처리: 로컬 MediaStream 만들기
218 function startAction() {
219   startButton.disabled = true;
220   navigator.mediaDevices.getUserMedia(mediaStreamConstraints)
221     .then(getLocalMediaStream).catch(handleLocalMediaStreamError);
222   trace('Requesting local stream.');
```