

2021-1 Capstone Design

# UV4L - DataChannel

*DeePLY*

Capstone Design Team : 5

17011885 박세정

17011869 이혜인

17011757 박미희

UV4L streaming server 실행 -> port 접속 -> 실행된 web에서 WebRTC

```
raspberrypi ~ $ uv4l --auto-video_nr --driver raspicam --encoding mjpeg --server-option '--port=9000' --server-option '--admin-password=myp4ssw0rd'
```

참고 링크 : <https://www.linux-projects.org/uv4l/tutorials/webrtc-data-channels/>

즉, UV4L로 가능해진 데이터 흐름은 다음 파이프 라인으로 표현할 수 있습니다.

(로봇)

앱 <- (유닉스 도메인 소켓) -> UV4L <- (WebRTC 데이터 채널) -> 원격 피어

UV4L Tutorial에서 제공된 datachannel\_server.cpp 파일을 라즈베리파이에 생성.

\$ sudo nano /etc/uv4l/uv4l-uvic.conf (raspicam일 경우 uv4l-raspicam.conf)

Config 파일(구성파일)에 들어가서 datachannel 연결 부분, 수정

```
--server-option=--enable-webrtc-datachannels=yes  
--server-option=--webrtc-datachannel-socket=/tmp/uv4l.socket
```

Cmd에서 코드 입력

```
$ datachannel_server /tmp/uv4l.socket
```

Datachannel\_server.cpp 저장한 폴더에서 uv4l.socke파일이 정상적으로 생성되었는지 확인

정상적으로 생성되었다면, \$ datachannel\_server /tmp/uv4l.socket를 입력하여 실행된 상태로

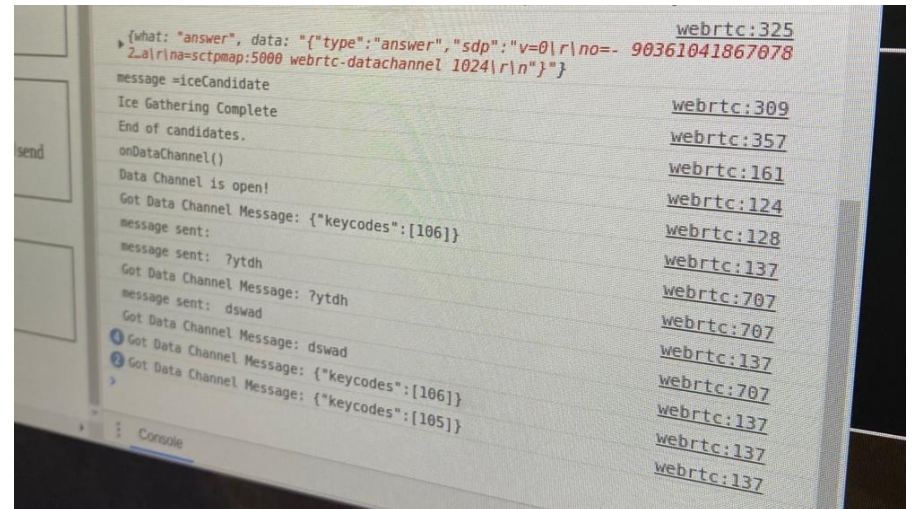
#### ▼ Data Channels

message:   received:   
☐ send device orientation angles alpha, beta, gamma ☐ send key codes (US layout) ☐ send mouse events

Call을 눌러 disable되어 있었던 Data Channel 창이 입력 가능한 상태로 바뀌는 지 확인

(UV4L Streaming Server 웹의 개발자 도구에서 Datachannel 메시지를 직접적으로 보는 것이 더 이해가 빠릅니다.)

onDataChannel()과 Data Channel is open!  
이 출력되면 uv4l streaming serve과  
원격 피어간의 datachannel은 정상적으로 작동합니다.



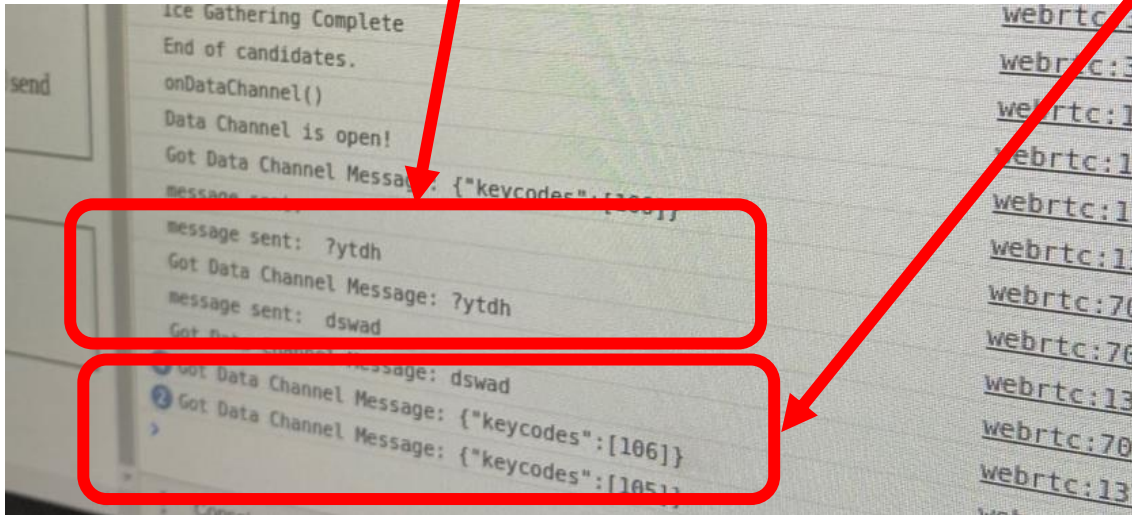
Datachannel이 열려있는 상태에서, 보내는 message 형태

Datachannel이 열릴 시 received에 메시지 출력

▼ Data Channels

message:   received:

☐ send device orientation angles alpha, beta, gamma ☐ send key codes (US layout) ☐ send mouse events



Message를 주고 받는 것과, send key codes 부분 Test

Message의 경우에는 string 형태로 datachannel에 전송

Send key code는 JSON파일로 변경되어 datachannel에 전송

즉, UV4L로 가능해진 데이터 흐름은 다음 파이프라인으로 표현할 수 있습니다.

앱 ← (유닉스 도메인 소켓) → UV4L ← (WebRTC 데이터 채널) → 원격 피어

완료

그렇다면, 어떻게 keycode를 제어하고 메시지를 로봇에 보낼 수 있는가?

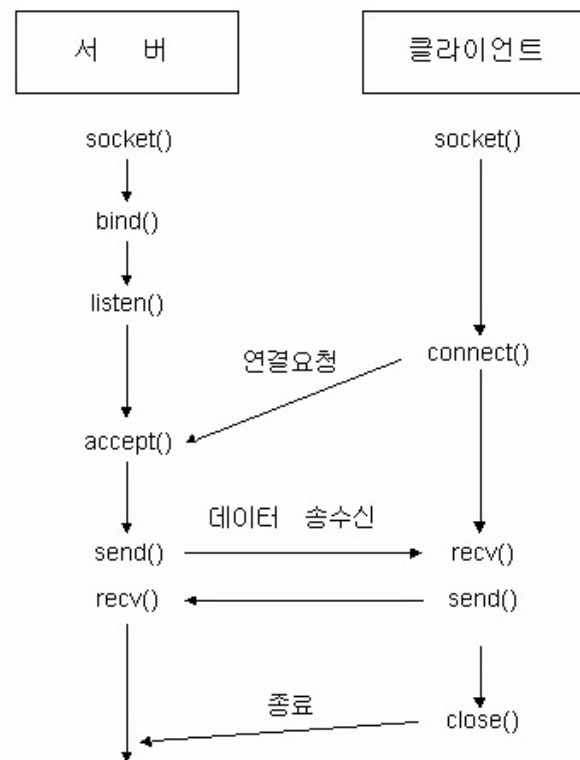
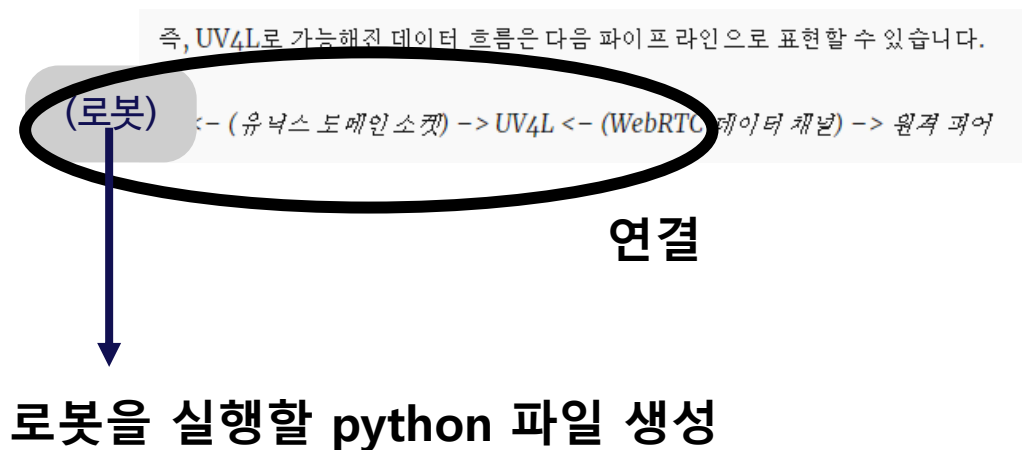
# UNIX SOCKET

참고링크 :

- (1) <https://stackoverflow.com/questions/45364877/interpreting-keypresses-sent-to-raspberry-pi-through-uv4l-webrtc-datachannel#>
- (2) <https://webrtcchacks.com/aiy-vision-kit-uv4l-web-server/>

UNIX\_SOCKET에 대한 이해를 돕기 위한 reference :

- (1) <https://nowonbun.tistory.com/668>
- (2) datachannel\_server.cpp 파일



참고 링크의 (1)드론 제어 python 코드를 이해 후, motor제어가 가능하도록 변경한 control코드

```
1  #coding:utf-8
2
3  import os
4  import time
5  import RPi.GPIO as GPIO
6
7  import socket
8  socket_path='/tmp/uv4l.socket'
9  try:
10     os.unlink(socket_path)
11 except OSError:
12     if os.path.exists(socket_path):
13         raise
14
15 s=socket.socket(socket.AF_UNIX, socket.SOCK_SEQPACKET)
16
17 #####
18 GPIO.setmode(GPIO.BCM) #MOTOR
19
20 ###
21 ENA=13 #좌?
22 ENB=20 #우?
23 IN1=19 #좌-앞
24 IN2=16 #좌-뒤
25 IN3=21 #우-앞
26 IN4=26 #우-뒤
27
28 ###
29 NONE=0
30 LEFT_ARROW=1 #왼쪽
31 RIGHT_ARROW=2 #오른쪽
32 UP_ARROW=3 #앞으로
33 DOWN_ARROW=4 #뒤로
34 STOP_ARROW=5 #멈춤
35
```

```
###
print('socket_path: %s' % socket_path)
s.bind(socket_path) #소켓 매핑
s.listen(1) # 연결 요청 대기 상태 설정
```

(모터 제어 부분 코드 생략)

```
while True:
    print('awaiting connection...')
    connection, client_address = s.accept() # 연결 승낙 후 실제 통신 소켓 반환
    print('client_address %s' % client_address)
    try:
        print('established connection with', client_address)

        data=connection.recv(20) #소켓 데이터 수신
        print('received message"%s"' % data)

        time.sleep(0.01)
        key=getch(int(data))

        #key=getch(data["keycodes"])
        print('key message "%d"' % key)

        if key == UP_ARROW:
            print('UP')
            Motor_Forward(key)
            time.sleep(1)

        elif key == DOWN_ARROW:
            print('DOWN')
            Motor_Backward(key)
            time.sleep(1)
```

```
elif key == STOP_ARROW:
    print('STOP')
    Motor_Stop(key)
    time.sleep(1)

if data:
    print('echo data to client')
    connection.sendall(data) # 데이터 송신

else:
    print('no more data from', client_address)
    break
```

```
finally:
    cleanup()
    connection.close()
```

Cmd에서 이 코드 대신, \$ datachannel\_server /tmp/uv4l.socket 제작한 python 코드 실행

Cmd에서 python저장해둔 경로에서 \$ python3 [파일이름].py

```
print('awaiting connection....')
connection, client_address = s.accept() # 연결 승낙 후 실제 통신 소켓 반환
print('client_address %s' %client_address)
try:
    print('established connection with', client_address)
```

이 부분이 cmd에서 print되면, 정상적으로 실행됨.

즉, UV4L로 가능해진 데이터 흐름은 다음 파이프라인으로 표현할 수 있습니다.

(로봇) <- (유닉스 도메인소켓) -> UV4L <- (WebRTC 데이터 채널) -> 원격 제어

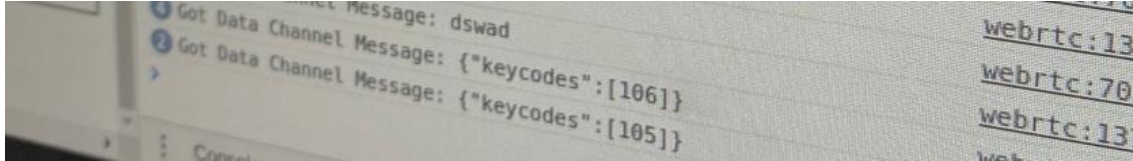
연결 성공

UV4L streaming serve의 datachannel에서 message로 숫자를 입력하면, cmd창에 실행시켜둔 python에서 입력이 들어가는 것을 확인할 수 있다.

Python 파일에서 socket으로 받은 string data를 int로 변환시켜준 후 작성해둔 모터 제어코드와 연결 -> 로봇 작동

키보드의 방향키로도 로봇의 모터를 제어하자

참고 링크 : [\[Python\] websocket으로 json 송신, 수신 하기 \(tistory.com\)](https://tistory.com)



Python 파일에서 JSON string 형태로 data 수신 => JSON 파싱

Python코드에 json 라이브러리를 불러오는 import json 을 추가하고, data를 받아와 JSON 파싱

```
jsondata=json.loads(data)
json_string=jsondata["keycodes"]
```

json\_string은 dict 형태이기 때문에 json\_string[0]을 int형으로 변환한 후 motor에 설정한 key값과 연결

**키보드의 방향키로도 로봇 바퀴 제어 성공**



## Custom한 web page에서도 datachannel 추가하기

(uv4l tutorial)Custom Web 참고링크 :

<https://www.linux-projects.org/uv4l/tutorials/custom-webapp-with-face-detection/>

Uv4l을 설치한 라즈베리파이의 경로 : /usr/share/uv4l/demo/facedetection

Demo 페이지 활용 (face-detection 웹 코드에는 datachannel로 값을 받아오지 않는다.)

Signalling.js에서 ondatachannel을  
이렇게 수정해주면 web에서  
javascript를 이용하여 생긴  
event 혹은 data를 datachannel를  
통해 python실행파일(로봇)에 송신

```
//data channel 열어줌
pc.ondatachannel = function (event) {
  console.log("a datachannel has been created");
  datachannel=event.channel; // 변수 선언 해주어야 앞에서 datachannel.send()구문으로 데이터 채널로 event가 들어간다.

  event.channel.onopen = function () {
    console.log("datachannel is open!");
  };

  event.channel.onerror = function (error) {
    console.error("datachannel error:", error);
  };

  event.channel.onmessage = function (event) {
    onDataChannelMsg(event.data);
  };

  event.channel.onclose = function () {
    datachannel=null;
    console.log("the datachannel is closed");
  };
};
```

Example(html-script)

```
<button class="arrow-key" id="left" value="37"><i class="fas fa-angle-left"></i></button>
<button class="arrow-key" id="down" value="40"><i class="fas fa-angle-down"></i></button>
<button class="arrow-key" id="right" value="39"><i class="fas fa-angle-right"></i></button>
<script>
  left.addEventListener('click', function (e) {
    var keycode = {
      keycode: 37
    };
    console.log("keycode BUTTON [left] : "+JSON.stringify(keycode));
    datachannel.send(JSON.stringify(keycode));
  }, false);
  down.addEventListener('click', function (e) {
```

참고 링크 :

[UV4L Streaming Server 웹페이지 프론트 코드]

<https://github.com/Onixaz/uv4l-webrtc-front-end/blob/master/main.js>