

2021-1 Capstone Design

13주차 캡스톤 진행 상황

DeePLY

Capstone Design Team : 5

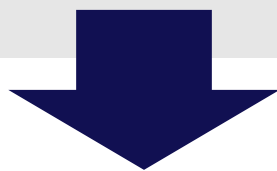
17011885 박세정

17011869 이혜인

17011757 박미희

13주차 목표

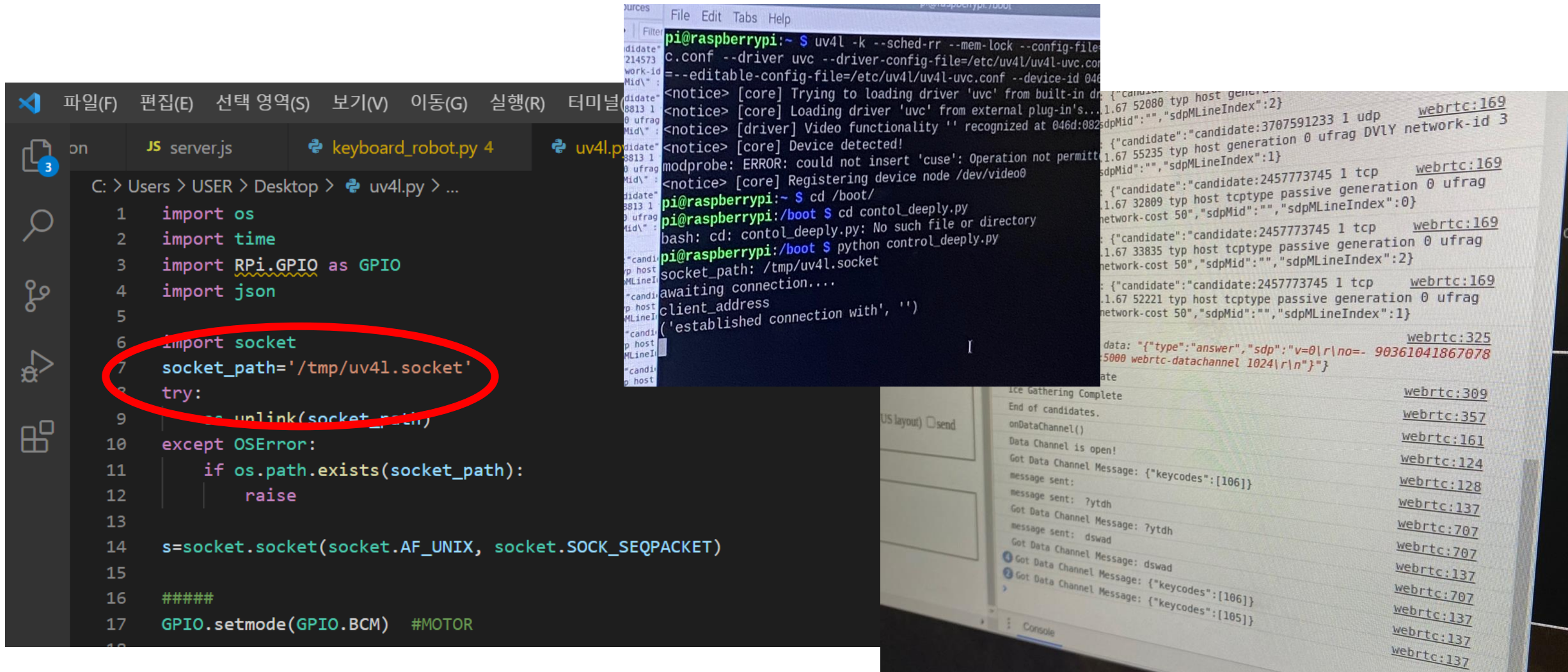
1. WebRTC API 중 **RTCPeerConnection**(Peer간 스트림(음성, 영상)전송) 사용하여 UV4L WebRTC 브라우저 스트리밍 영상 web으로 가져오기
2. 라즈베리파이 USB webcam에서 **OpcenCV** 사용하여 스트리밍 영상 가져오기



1. WebRTC API 중 **RTCPeerConnection**(Peer간 스트림(음성, 영상)전송) 사용하여 UV4L WebRTC 브라우저 스트리밍 영상 web으로 가져오기 ✓
2. 라즈베리파이 USB webcam에서 **OpcenCV** 사용하여 스트리밍 영상 가져오기 ! 로봇 제어에 집중
3. ROBOT ARM 제어 ✓
4. WEB에 로봇 제어 기능 추가하여 로봇과 통신하기 ✓

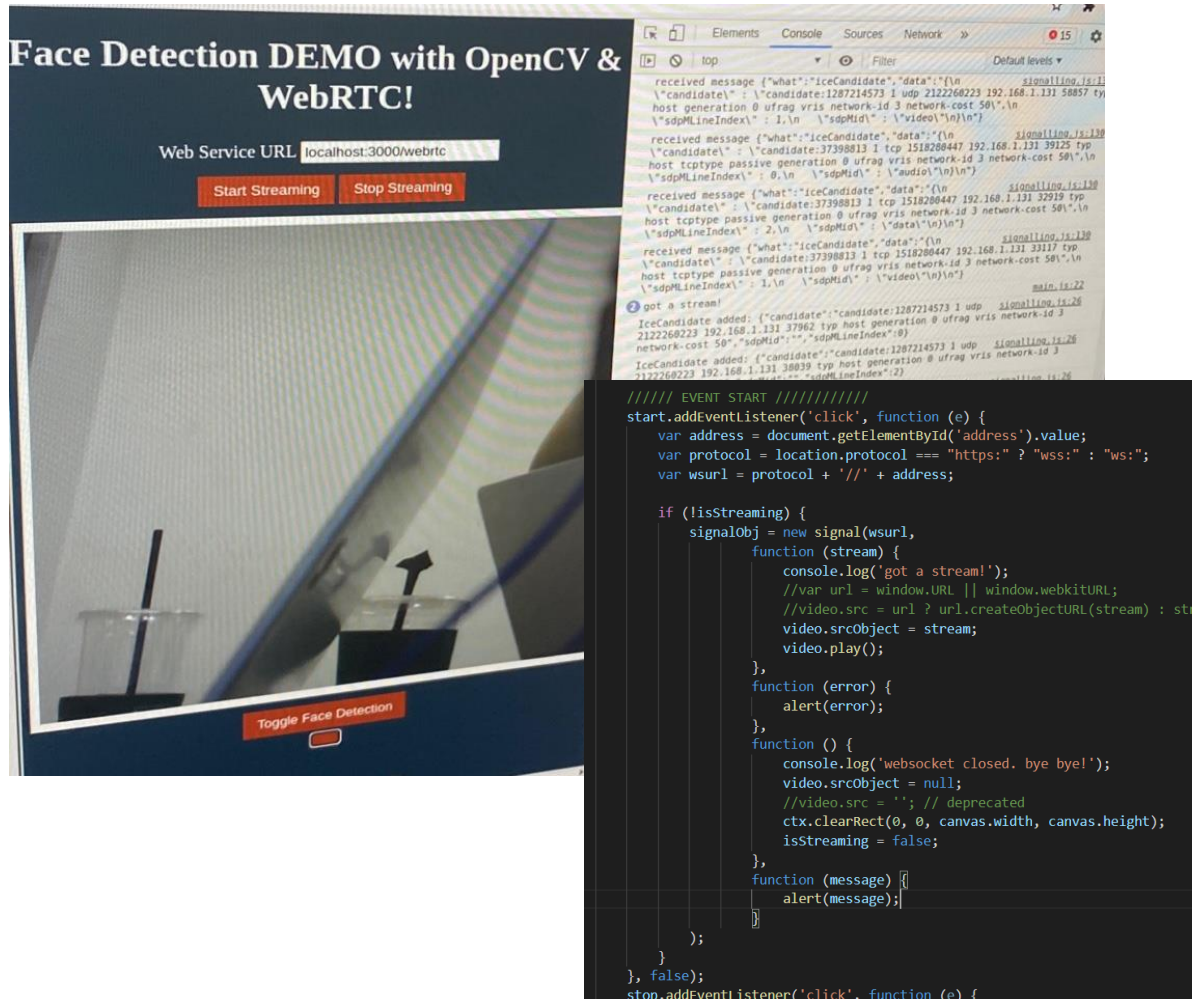
지난 12주차 진행 과정

UV4L socket을 이용한 datachannel 구현하여 로봇 제어



지난 13주차 진행 과정

UV4L WebRTC 브라우저 스트리밍 영상 web으로 가져오기



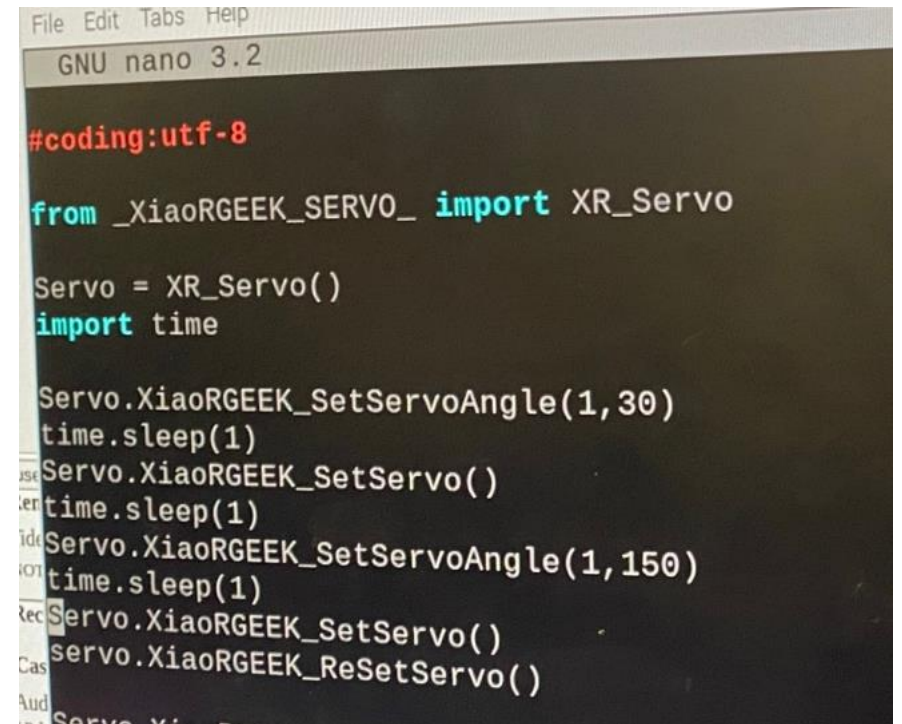
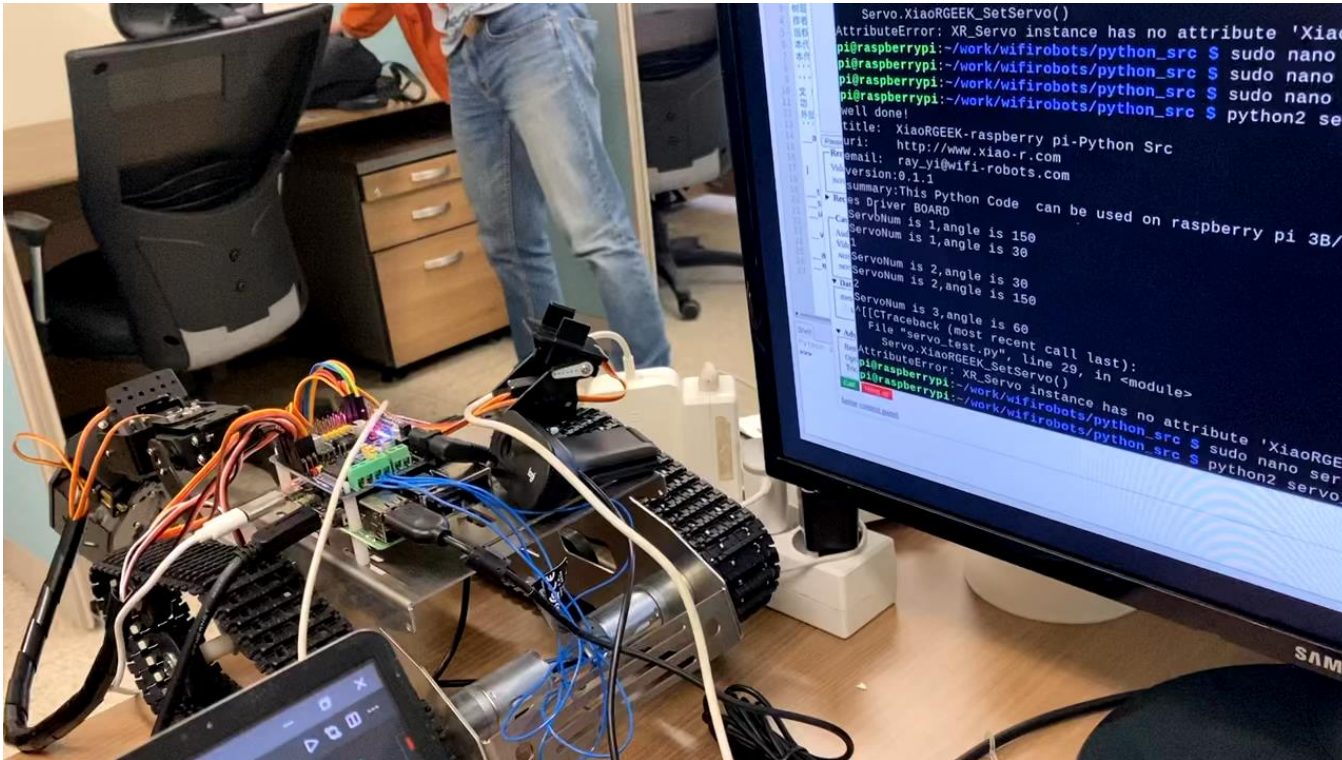
```
JS signalling.js > signal > addIceCandidates\n1 /*\n2  * window.mozRTCPeerConnection, window.mozRTCSessionDescription, window.mozRTCIceCandidate are now deprecated\n3  */\n4\n5 RTCPeerConnection = window.RTCPeerConnection || /*window.mozRTCPeerConnection ||*/ window.webkitRTCPeerConnection;\n6 RTCSessionDescription = /*window.mozRTCSessionDescription ||*/ window.RTCSessionDescription;\n7 RTCIceCandidate = /*window.mozRTCIceCandidate ||*/ window.RTCIceCandidate;\n8\n9 function signal(url, onStream, onError, onClose, onMessage) {\n10   if ("WebSocket" in window) {\n11     console.log("opening web socket: " + url);\n12     var ws = new WebSocket(url);\n13     var pc;\n14     var iceCandidates = [];\n15     var hasRemoteDesc = false;\n16\n17     function addIceCandidates() {\n18       if (hasRemoteDesc) {\n19         iceCandidates.forEach(function (candidate) {\n20           pc.addIceCandidate(candidate,\n21             function () {\n22               console.log("IceCandidate added: " + JSON.stringify(candidate));\n23             },\n24             function (error) {\n25               console.error("addIceCandidate error: " + error);\n26             }\n27           );\n28         }\n29       }\n30       iceCandidates = [];\n31     }\n32\n33     onopen = function () {\n34       /* First we create a peer connection */\n35       var config = {"iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}]};\n36       var options = (optional: []);\n37
```


지난 13주차 진행 과정

ROBOT ARM 제어

Python으로 로봇의 라이브러리를 불러와 ROBOT ARM 제어

사용 파일 : _XiaoRGEEK_SERVO_.so, XiaoRGEEK_about.py, servo_test.py(제작 코드)

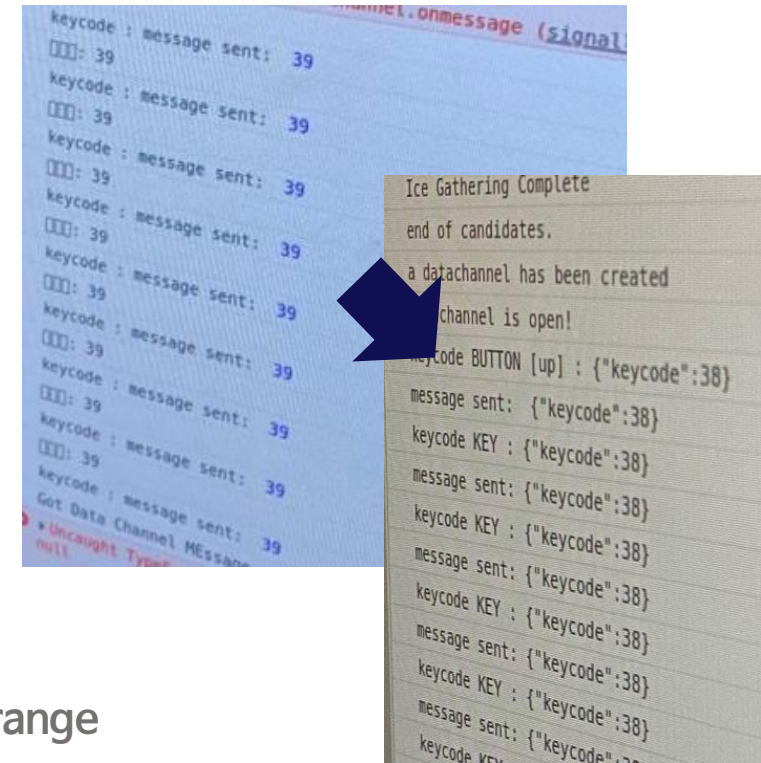


지난 13주차 진행 과정

WEB에서 로봇과 통신하기

1. JSON 파일로 data 변환하기

로봇 : Python 실행파일에서 JSON 파일 파싱



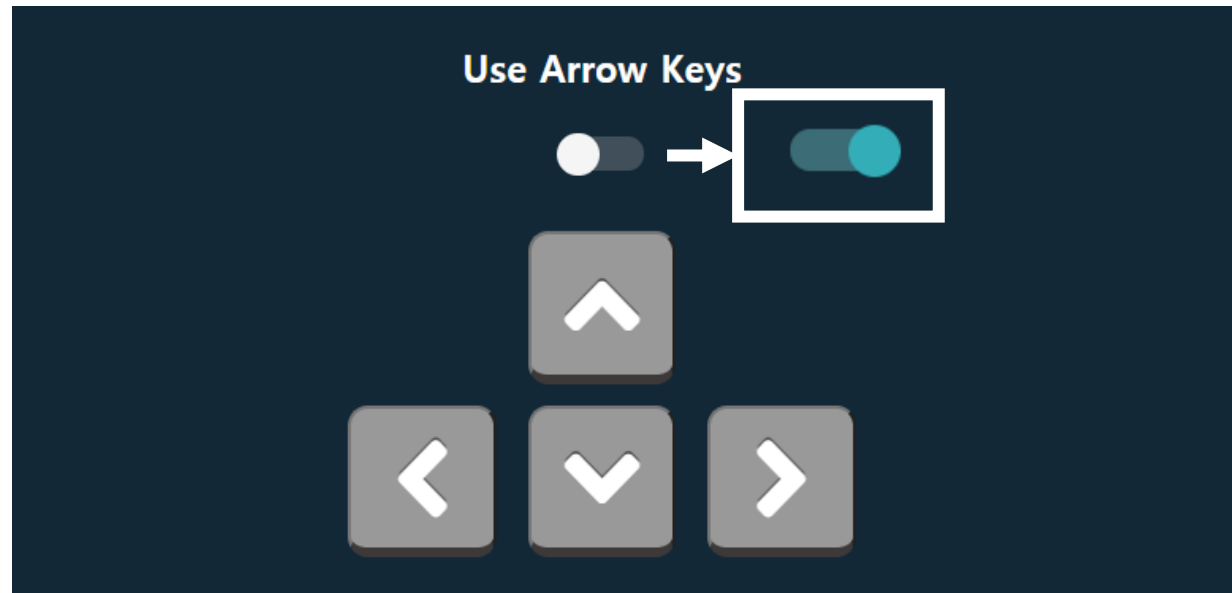
➤ 14주차에 진행 예정

지난 13주차 진행 과정

WEB에서 로봇과 통신하기

2. 방향키 제어하기

- WEB에 버튼을 제작하여 버튼을 누르면, DataChannel을 통해 로봇의 MOTOR 제어
- CheckBox를 켜었을 때, 키보드의 방향키로도 로봇의 모터를 제어할 수 있도록 기능 추가

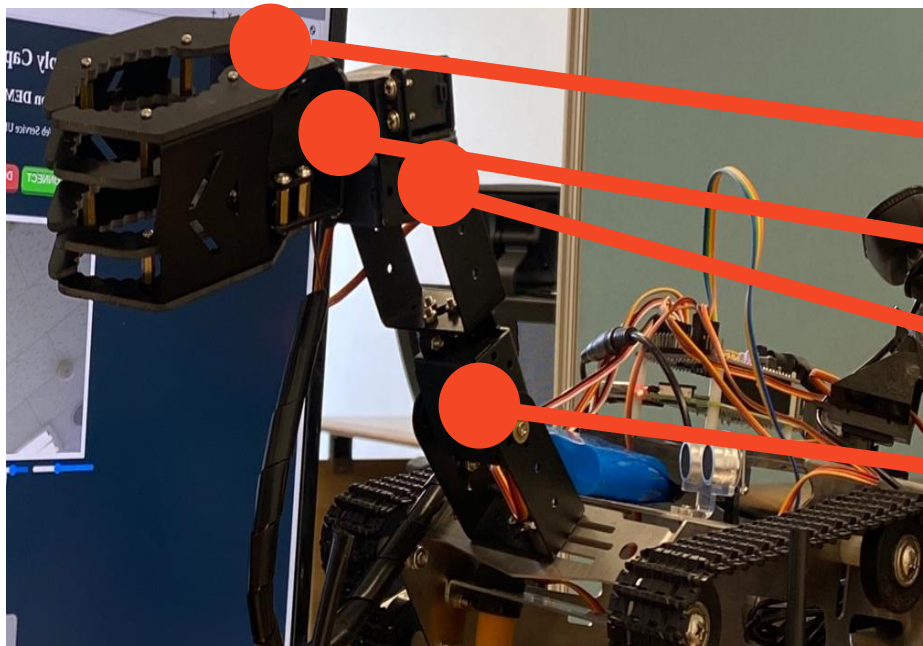


지난 13주차 진행 과정

WEB에서 로봇과 통신하기

3. 로봇 팔 제어하기

- WEB에 input type=range를 제작하여 버튼을 누르면, DataChannel을 통해 로봇의 SERVO MOTOR 제어
- Range를 입력 받아, range에 해당하는 각도만큼 로봇 팔 제어 가능



지난 13주차 진행 과정

WEB에서 로봇과 통신하기

3. Python 실행 코드 수정

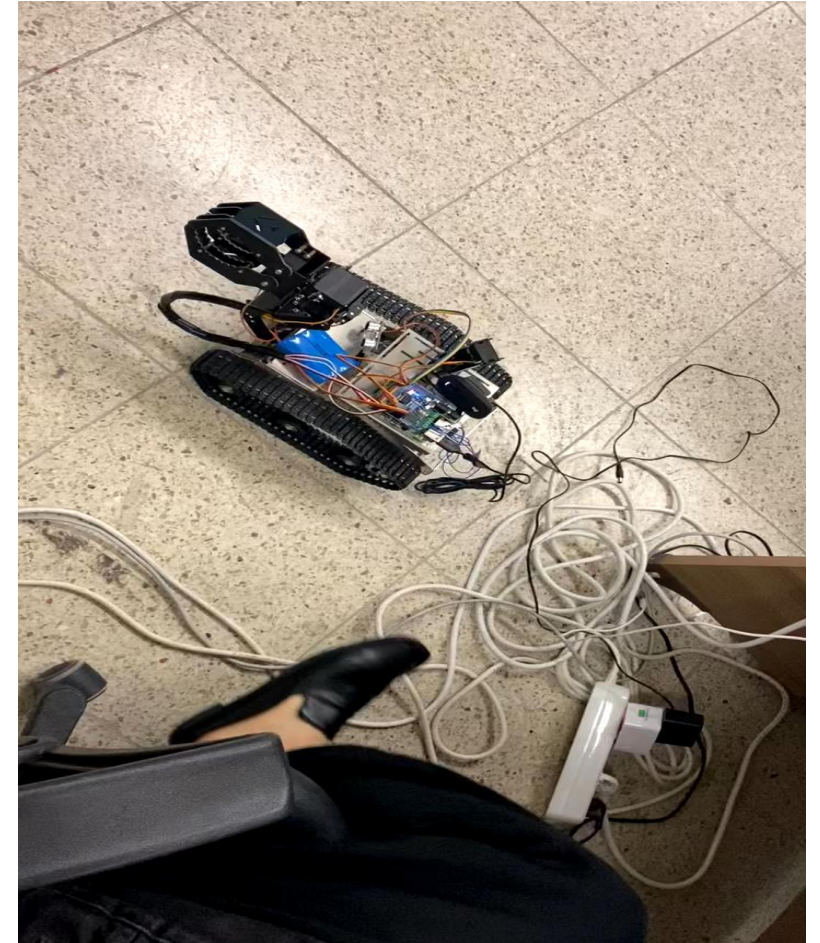
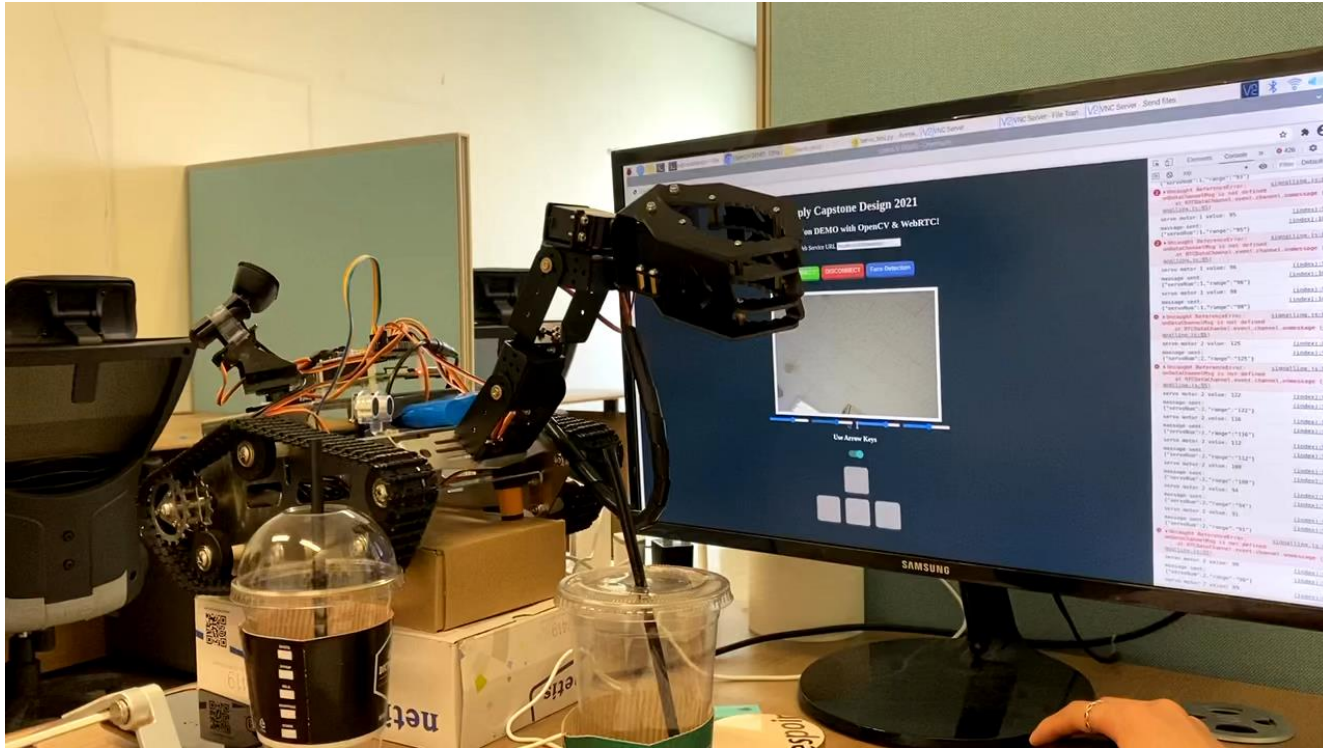
- JSON 파일로 받은 Data를 파싱 하여 필요한 정보를 배열 저장
- 배열을 이용하여 로봇 제어

```
Ice Gathering Complete
end of candidates.
a datachannel has been created
datachannel is open!
keycode BUTTON [up] : {"keycode":38}
message sent: {"keycode":38}
keycode KEY : {"keycode":38}
message sent: {"keycode":38}
keycode KEY : {"keycode":38}
message sent: {"keycode":38}
keycode KEY : {"keycode":38}
message sent: {"keycode":38}
keycode KEY : {"keycode":38}
message sent: {"keycode":38}
keycode KEY : {"keycode":38}
```

```
<notice> [00:00:00]
pi@raspberrypi:~ $ cd /home/pi/Desktop
pi@raspberrypi:~/Desktop $ cd servo_json
pi@raspberrypi:~/Desktop/servo_json $ python2 servo_test.py
well done!
title: XiaoRGEEEK-raspberry pi-Python Src
uri: http://www.xiao-r.com
email: ray_yi@wifi-robots.com
version:0.1.1
summary:This Python Code can be used on raspberry pi 3B/3B+,The d
efinition of GPIO pin is based on XiaoRGEEEK PWR.A53 series Driver
BOARD
socket_path: /tmp/uv4l.socket
awaiting connection....
client_address
('established connection with', '')
received message{"servoNum":4,"range":"121"}
('1 jsondata: ', {u'range': u'121', u'servoNum': 4})
('2 data type:', <type 'str'>)
('3 jsondata type: ', <type 'dict'>)
{"servoNum":4,"range":"121"}
cleanup
```

지난 13주차 진행 과정

결과



지난 14주차 목표

카메라의 SERVO MOTOR 제어 추가

로봇 python 실행 코드 수정

자연스러운 로봇 동작 구성

웹 인터페이스