

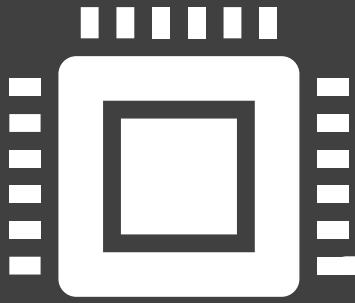


# 군집주행 PJT

조원

안재운 김성욱

최준하 노현진

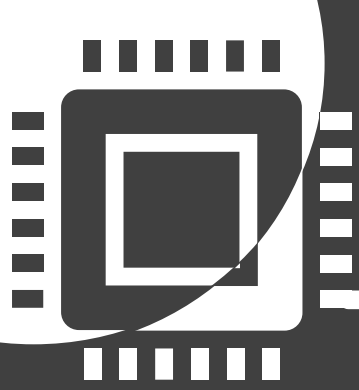


# Contents

역할 분담 소개

구현 기능

어려웠던 점



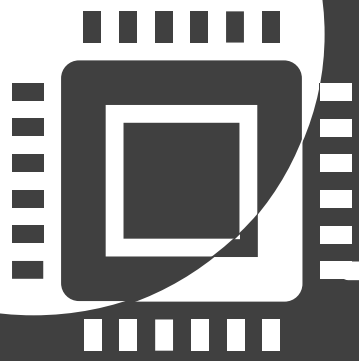
# 역할 분담

김성욱 / 안재운

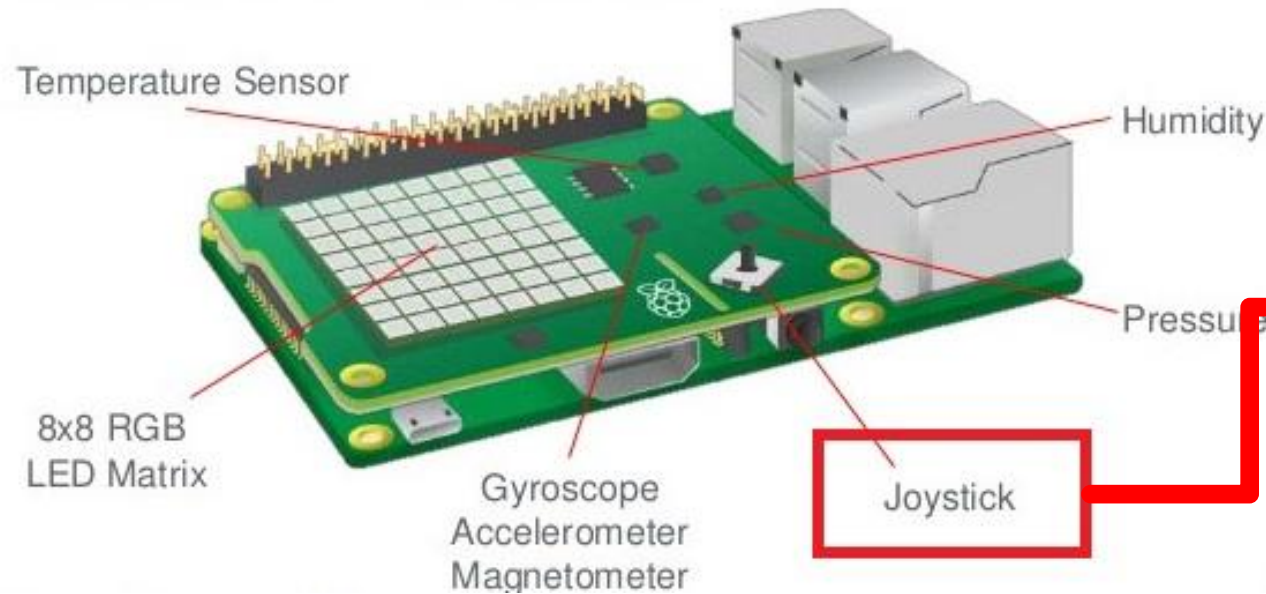
제어 기능 구현

최준하 / 노현진

네트워크 연결 구현



# 제어 기능 구현



SenseHat 을 이용해  
조이스틱 구성

조이스틱의 입력 값을  
AWS를 통해 전송

# 제어 기능 구현

```
# Callback when the subscribed topic receives a message
def on_message_received(topic, payload, dup, qos, retain, **kwargs):
    # 0:F 1:B 2:L 3:R
    # print("Received message from topic '{0}': {1}".format(topic, payload))

    # convert bytes type to string(json)
    msg = payload.decode('utf-8')

    # convert json to dict
    msg = json.loads(msg)
    # print(f"decoded string : {msg}, {type(msg)}")

    # get message
    message = msg
    sign = message["message"]
    global received_count
    received_count += 1

    print(sign)
```

조이 스틱 조작 시  
메시지 수신

```
if sign == 1:
    Speed += Accel
elif sign == 0:
    Speed -= Accel
elif sign == 2:
    Direction -= 50
elif sign == 3:
    Direction += 50
```

```
if Speed > 255:
    Speed = 255
```

```
if Speed < -255:
    Speed = -255
```

```
if Direction >= 400:
    Direction = 400
```

```
if Direction <= 200:
    Direction = 200
```

```
FB = 1
if Speed < 0:
    FB = -1

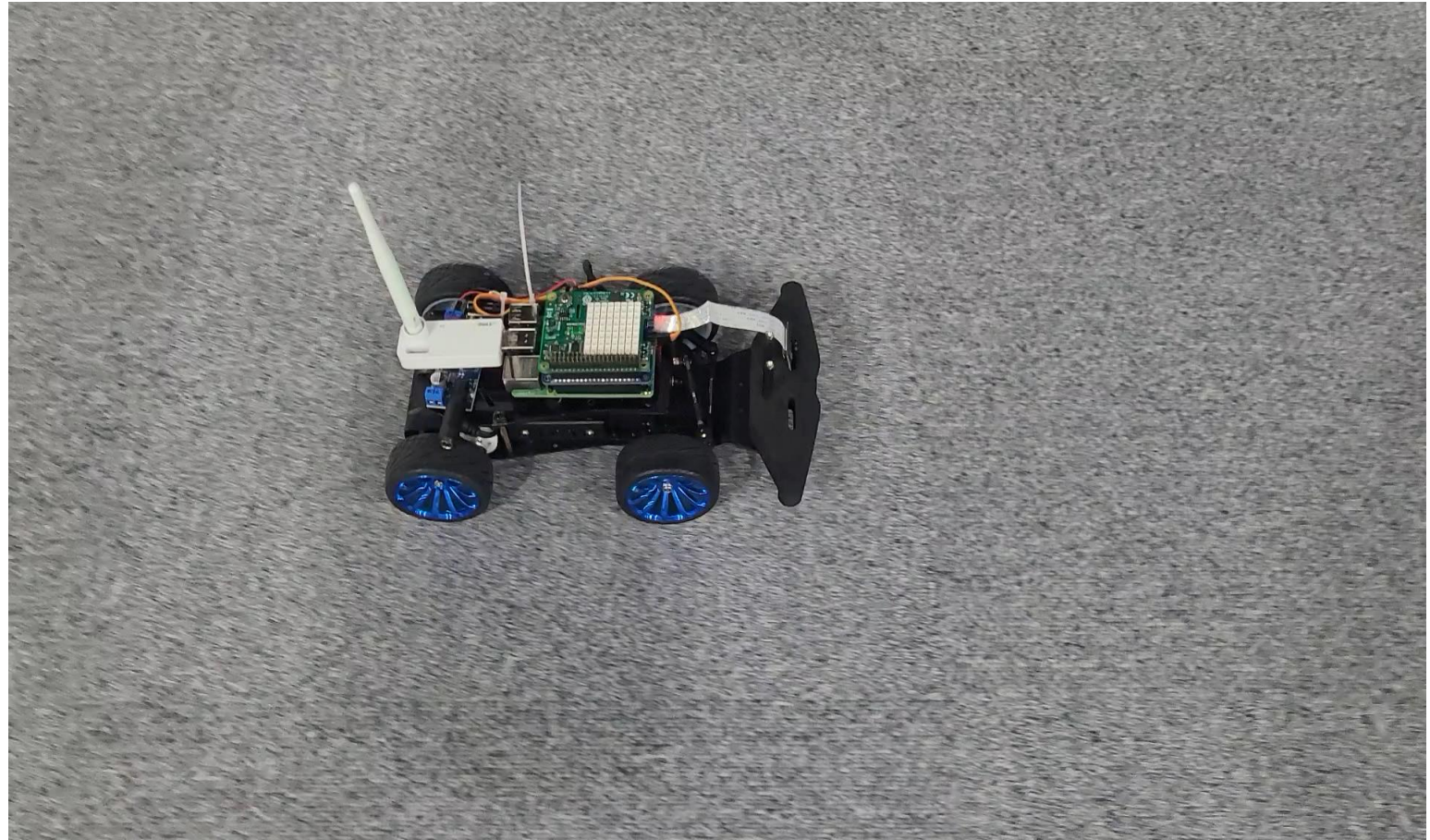
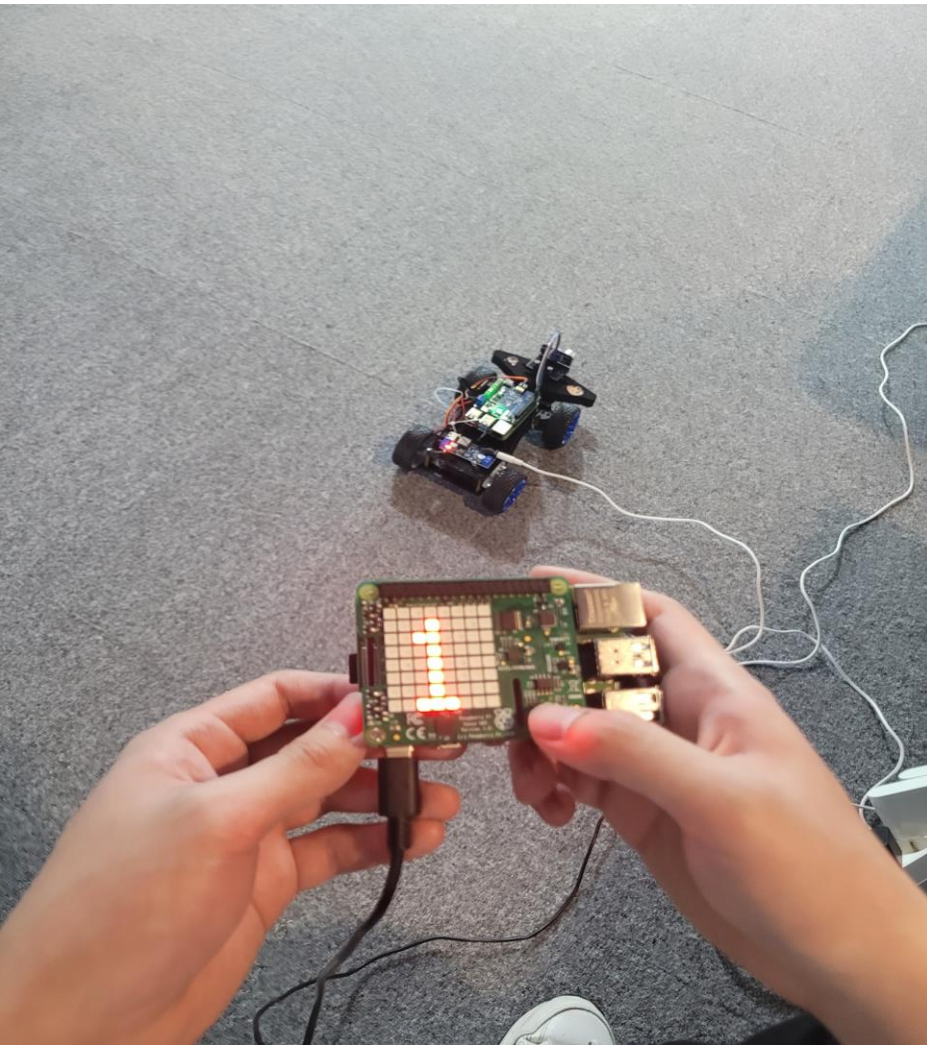
if FB == 1:
    myMotor.run(Raspi_MotorHAT.FORWARD)
elif FB == -1:
    myMotor.run(Raspi_MotorHAT.BACKWARD)

myMotor.setSpeed(Speed * FB)
servo_control.setPWM(0, 0, Direction)
```

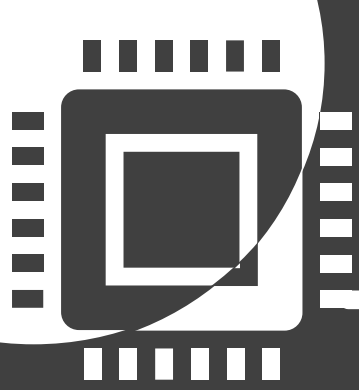
메시지 값에 따라  
속도, 방향 전환

입력한 값을  
하드웨어에 전송

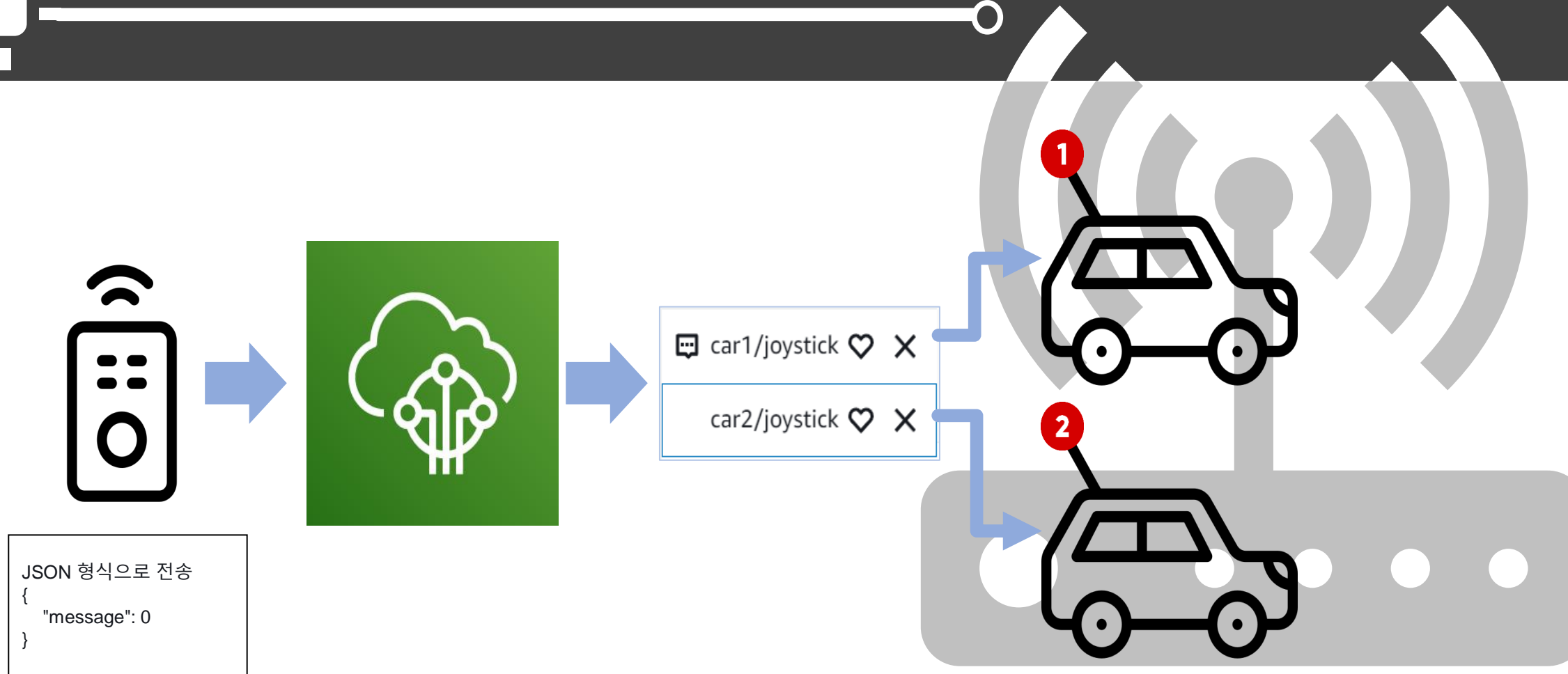
# 조향 및 주행 테스트

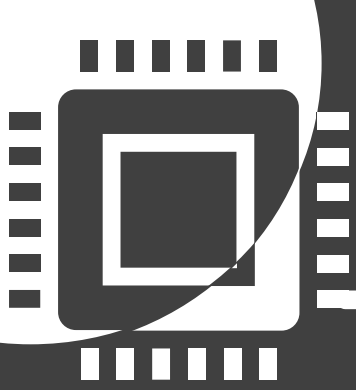




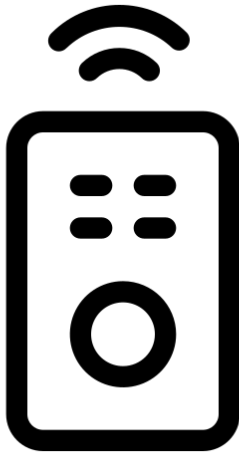


# 네트워크 연결 구현





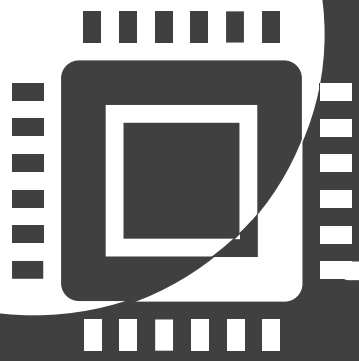
# 네트워크 연결 구현



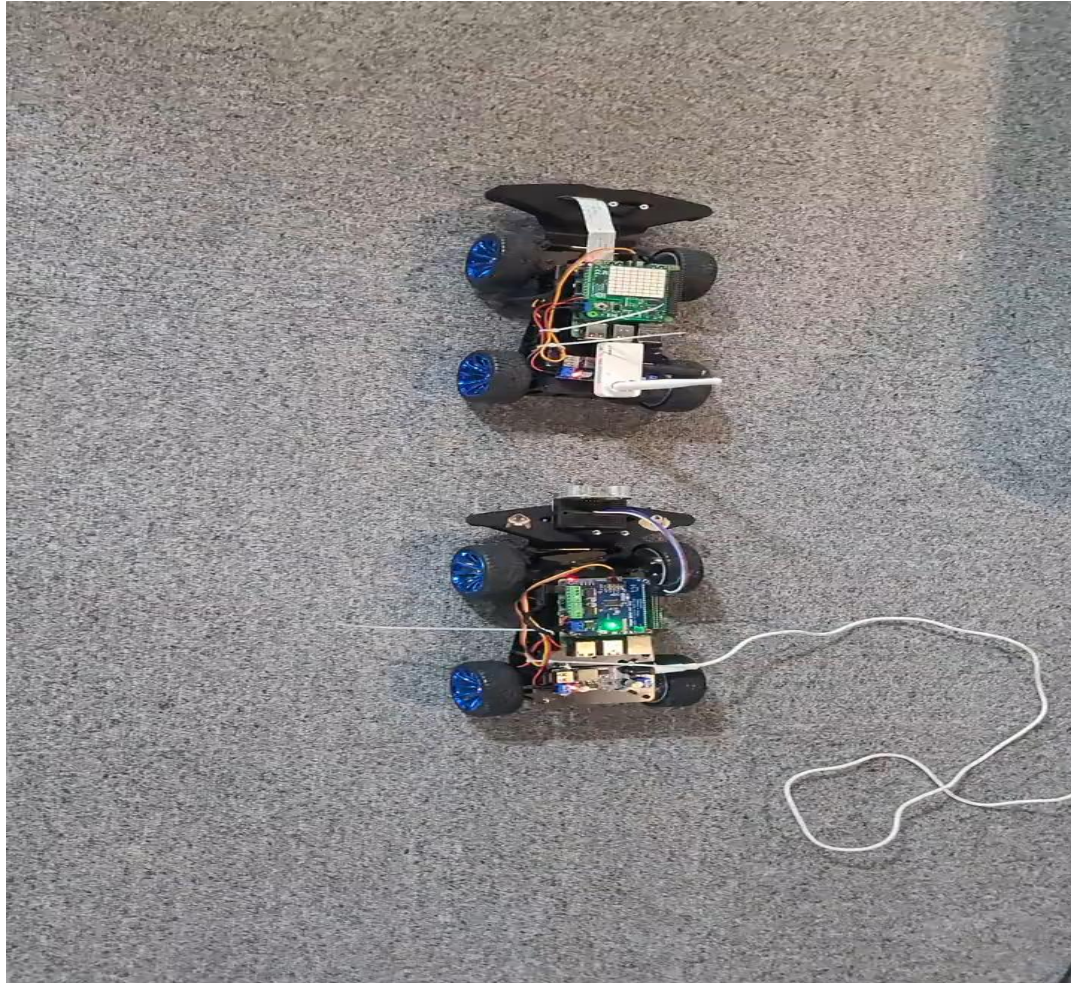
Message Format  
{  
  "message": 0  
}

MQTT 메시지 처리 상세		
Key	Value	Description
"message"	0	전진
	1	후진
	2	좌회전
	3	우회전
	-1	다른 차와 군집 주행 시작 ex) car1 이 -1을 수신했을 경우, car2 뒤를 따라가는 자동 주행 시작





# 군집 주행 영상



# 제어 기능 구현

```
while Finish == 0:
```

```
    constDist = 0.2
    Dist = sensor.distance

    if Dist - constDist > 0.2:

        Accel = (Dist - constDist) * 300
        Speed += int(Accel * FB)

        if Speed > 150:
            Speed = 150

        if Speed < -150:
            Speed = -150

        if Speed > 0:
            FB = 1
```

초음파 센서를 이용  
거리를 측정하면서  
일정 거리 유지

```
elif Dist - constDist < 0:

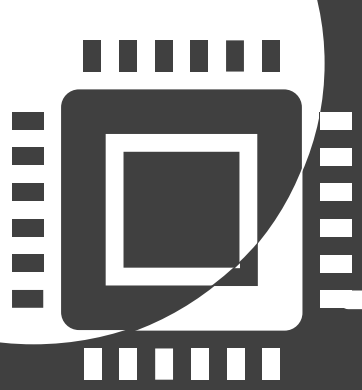
    Accel = (Dist - constDist) * 300
    Speed += int(Accel * FB)

    if Speed < 0:
        FB = -1
    if Speed > 150:
        Speed = 150
    if Speed < -150:
        Speed = -150

else:
    Speed = 0
    FB = 1

if FB == -1:
    myMotor.run(Raspi_MotorHAT.FORWARD)
else:
    myMotor.run(Raspi_MotorHAT.BACKWARD)

myMotor.setspeed(Speed)
print(Dist, 'm', Speed * FB, FB)
sleep(0.1)
```



# 어려웠던 점

- 시행착오의 연속
  - 1. 제어 재설계
    - 컴퓨터 키보드 인터럽트로 제어
      - > 조이스틱
      - > 센스햇 조이스틱
  - 2. 서버 설계
    - Localhost MQTT
      - > AWS IoT MQTT 브로커

