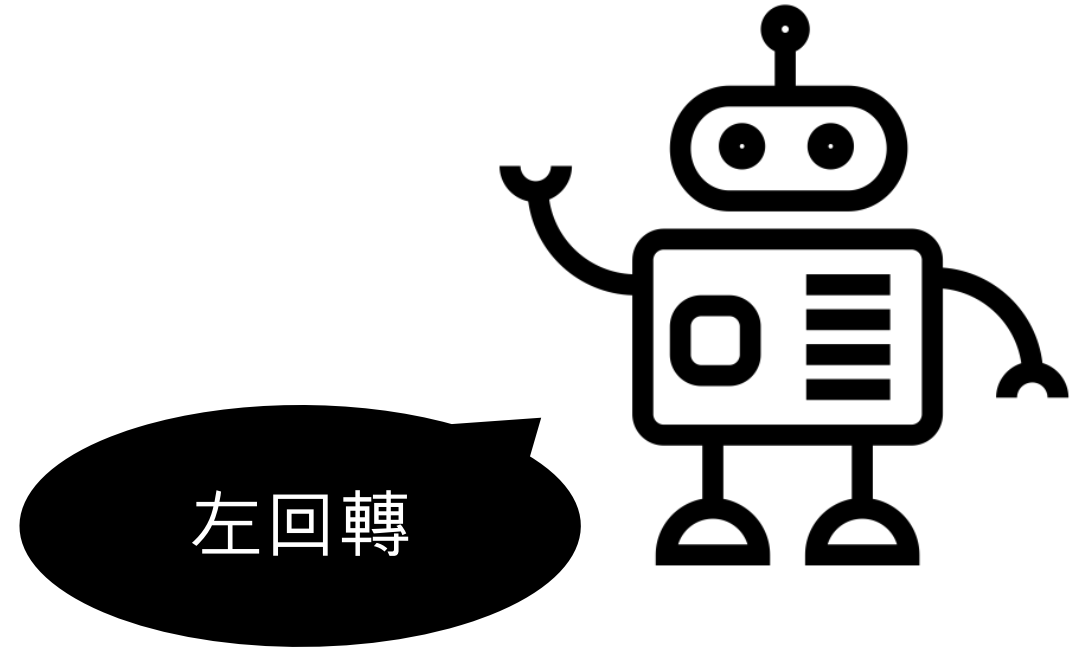
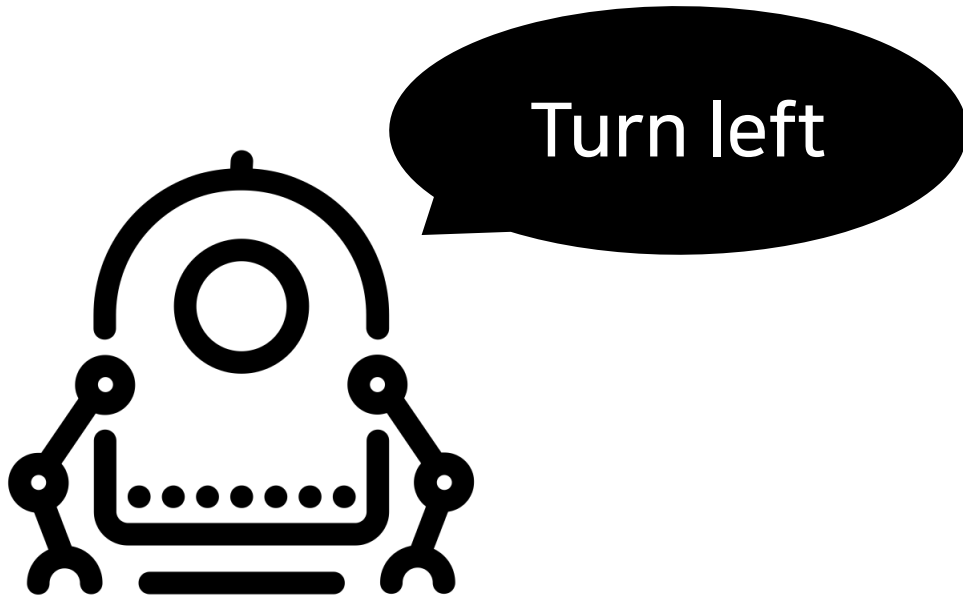


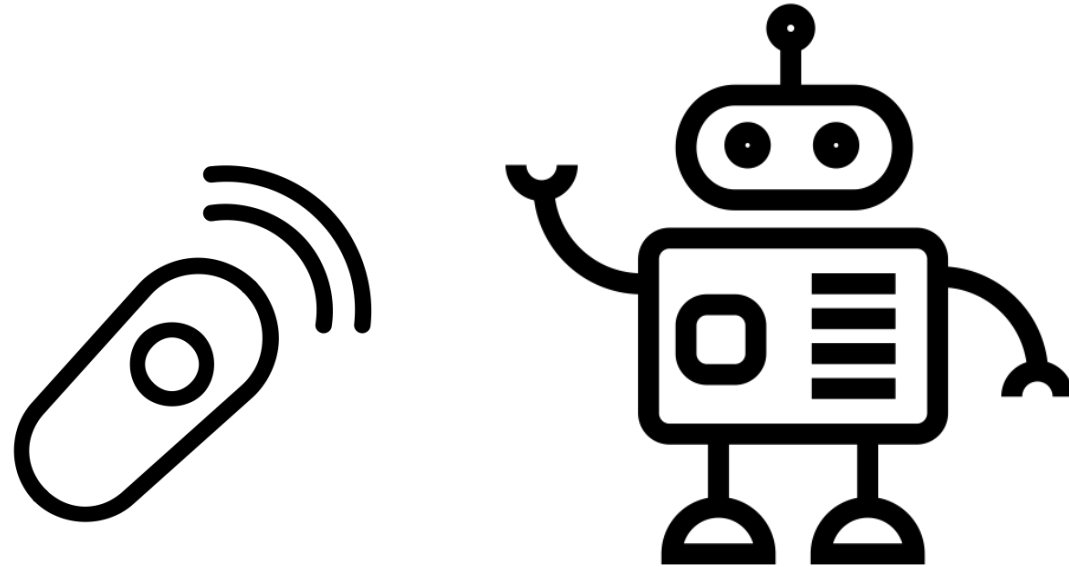
Control + R

로봇 제어를 위한 실행 플랫폼

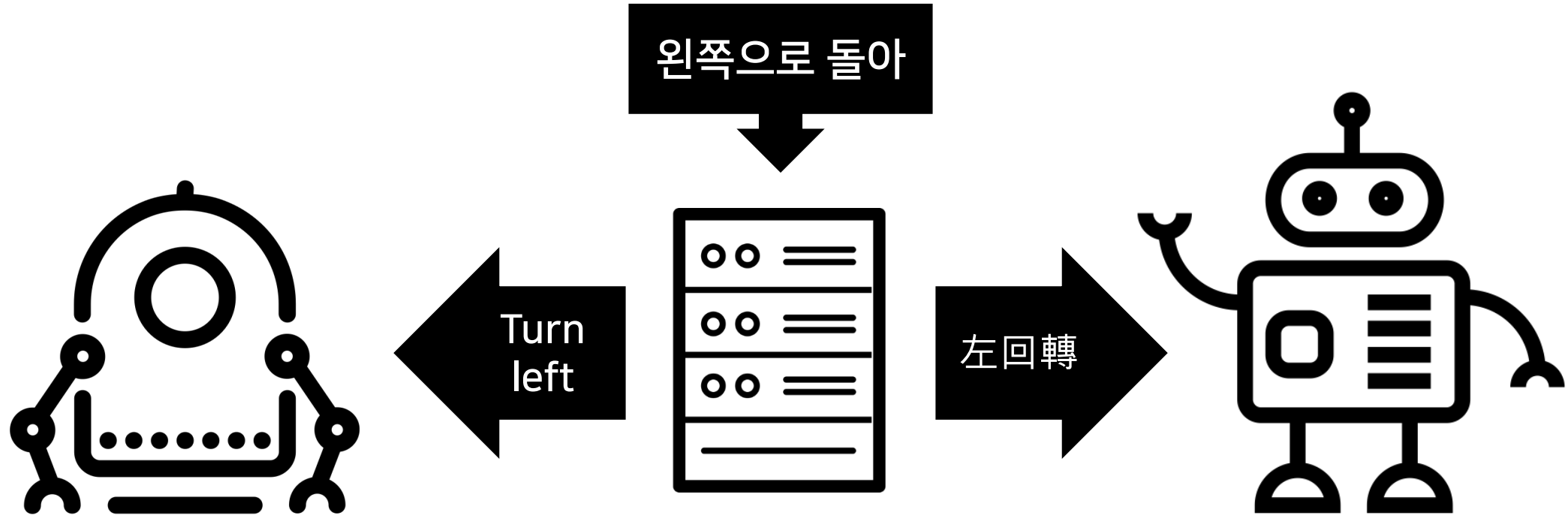
5조 - 이거 끝나면 군대 갈 팀
전자정보공학부 20150452 심규원
전자정보공학부 20160428 김동현



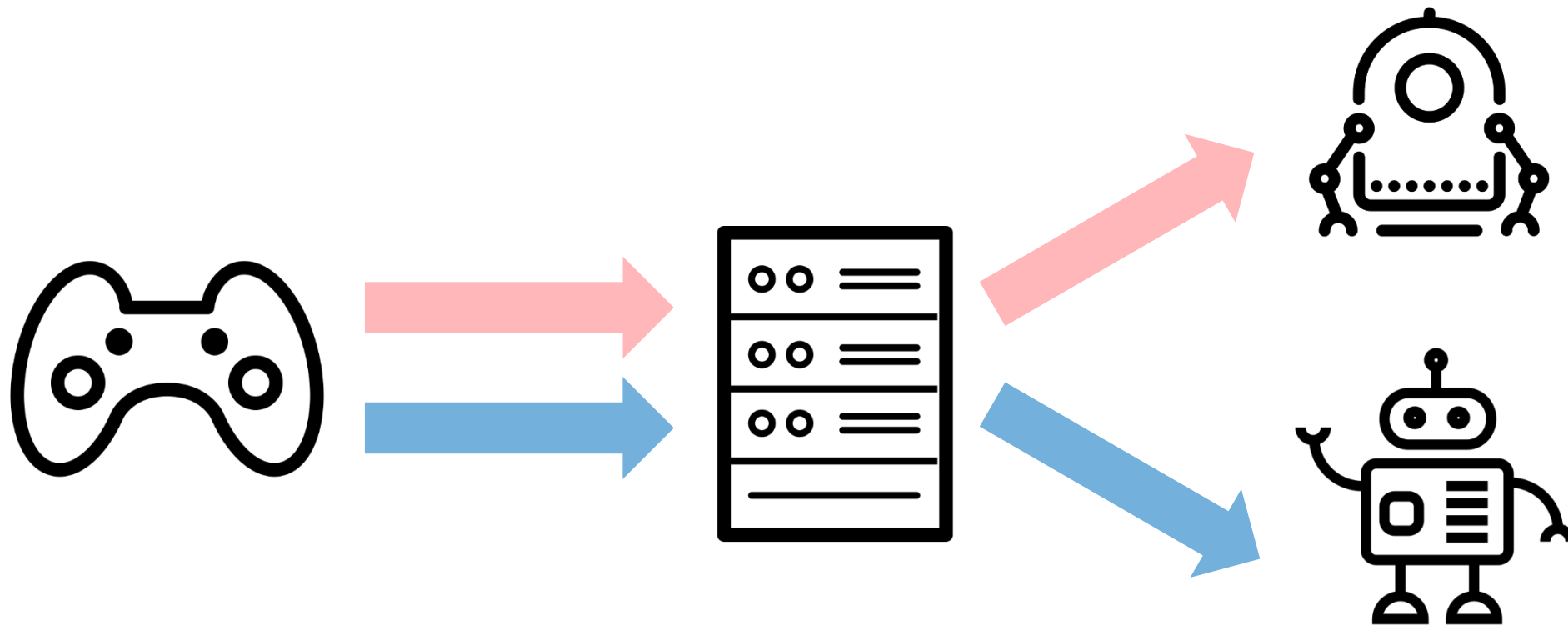
1. 로봇마다 사용하는 프로토콜이 다름



2. 로봇을 제어하는 방법이 한정적임

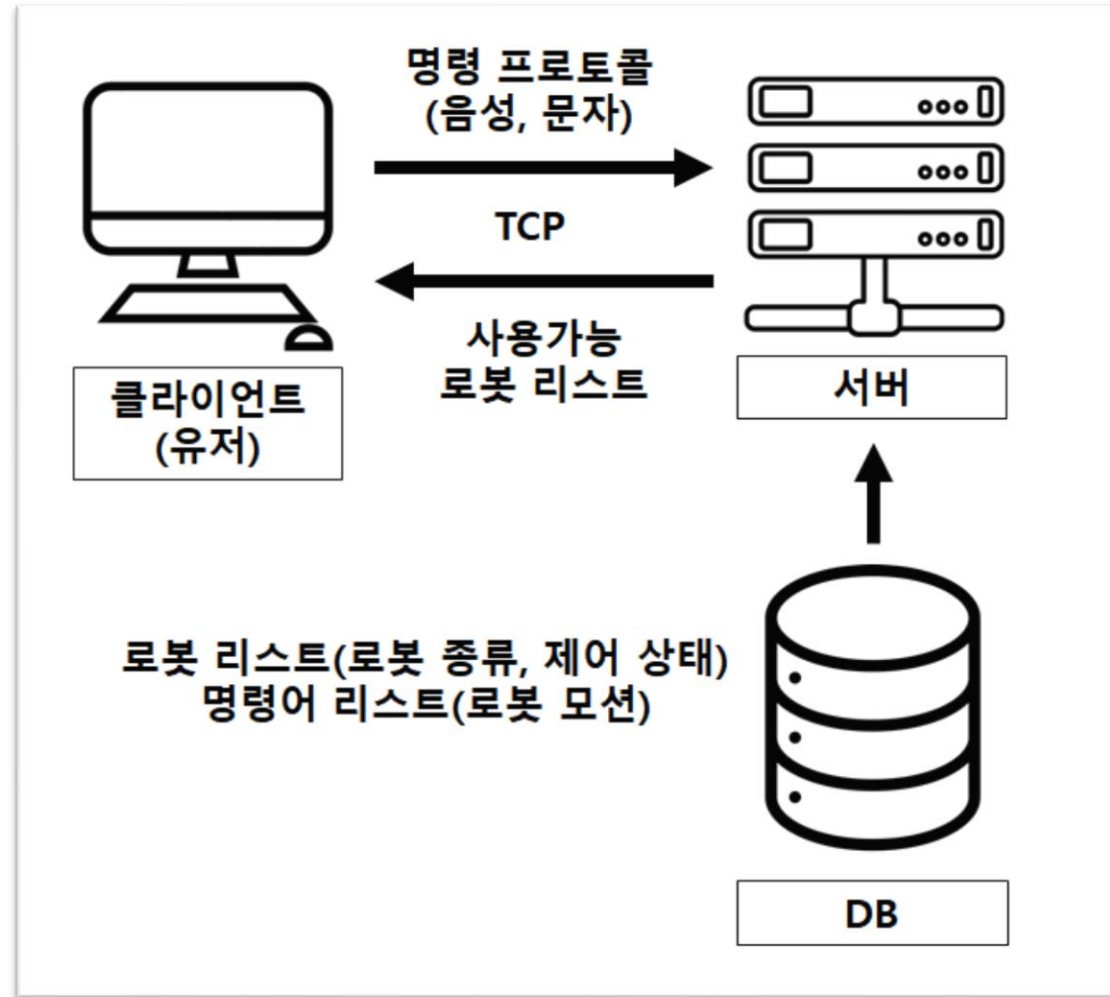


로봇제어를 편리하게 제공하는 플랫폼



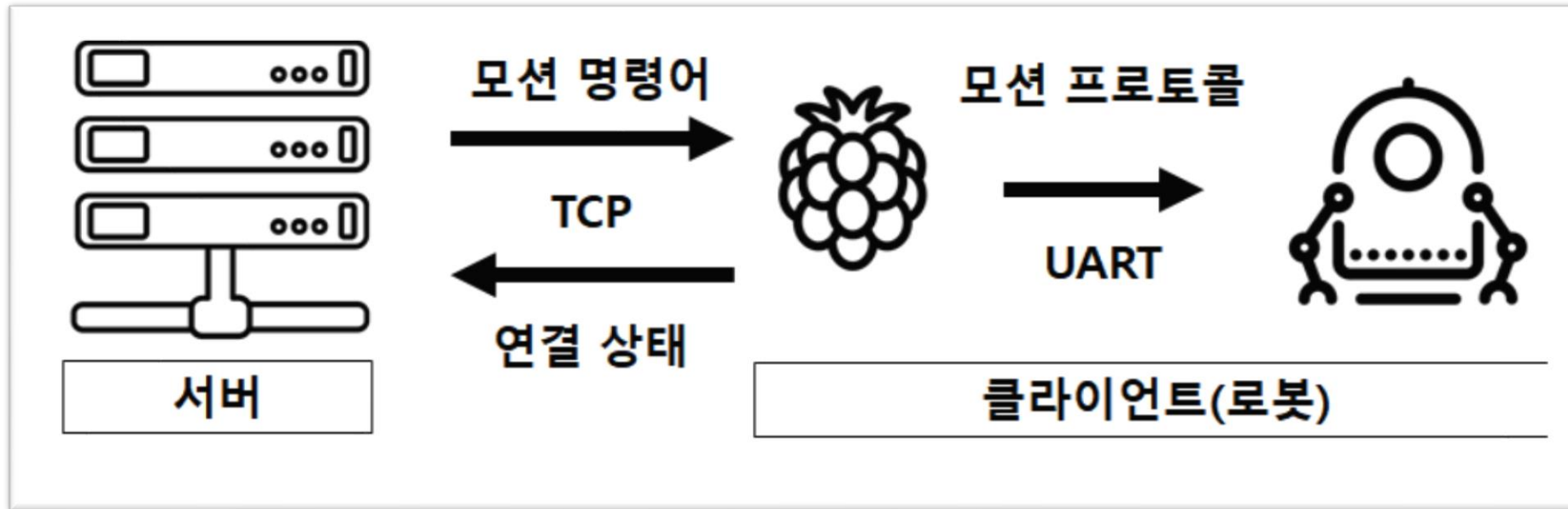
로봇제어를 편리하게 제공하는 플랫폼

TOBE 시스템 기능 구상도



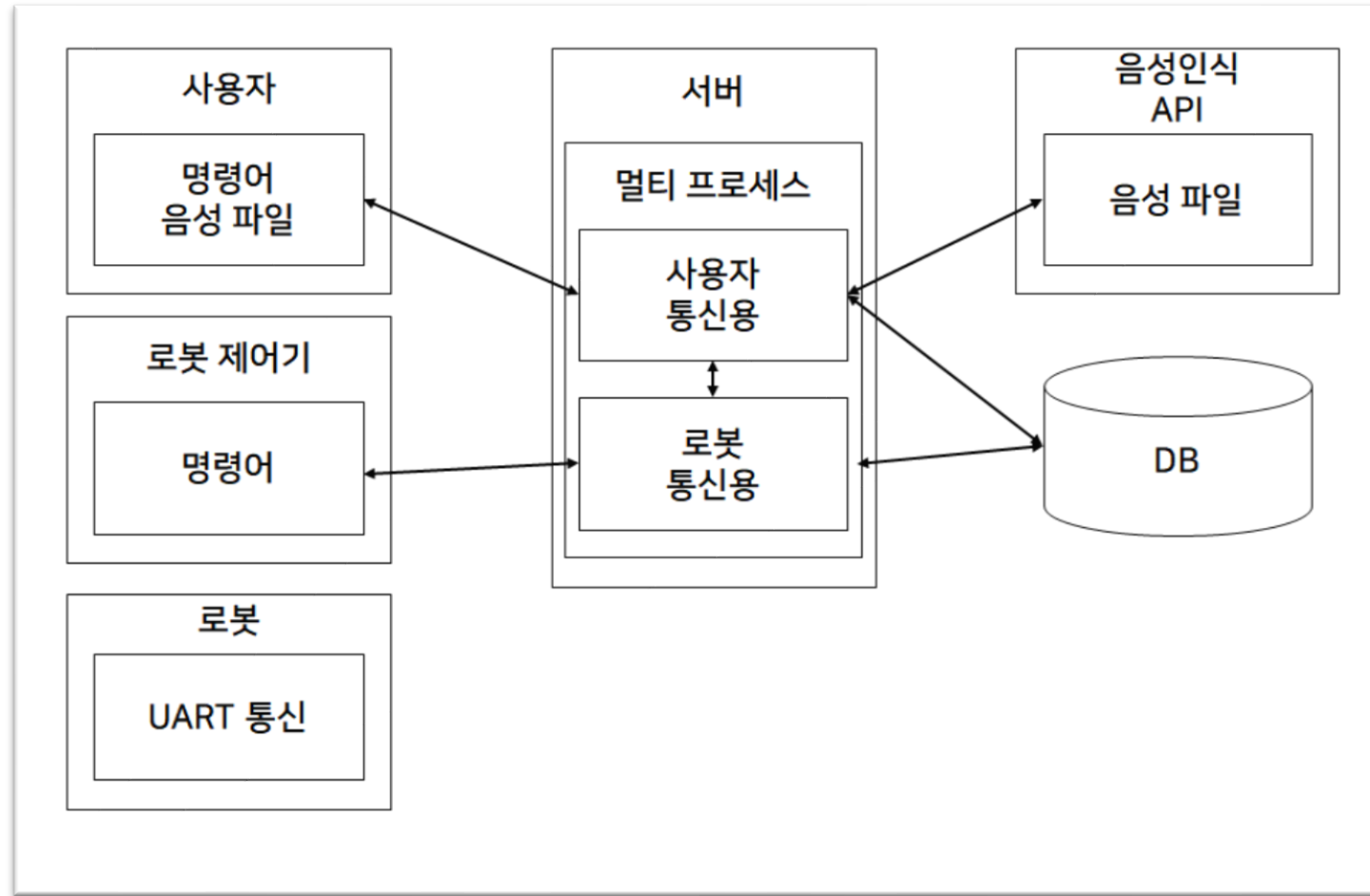
유저 클라이언트 - 서버 간 통신 구조

TOBE 시스템 기능 구상도

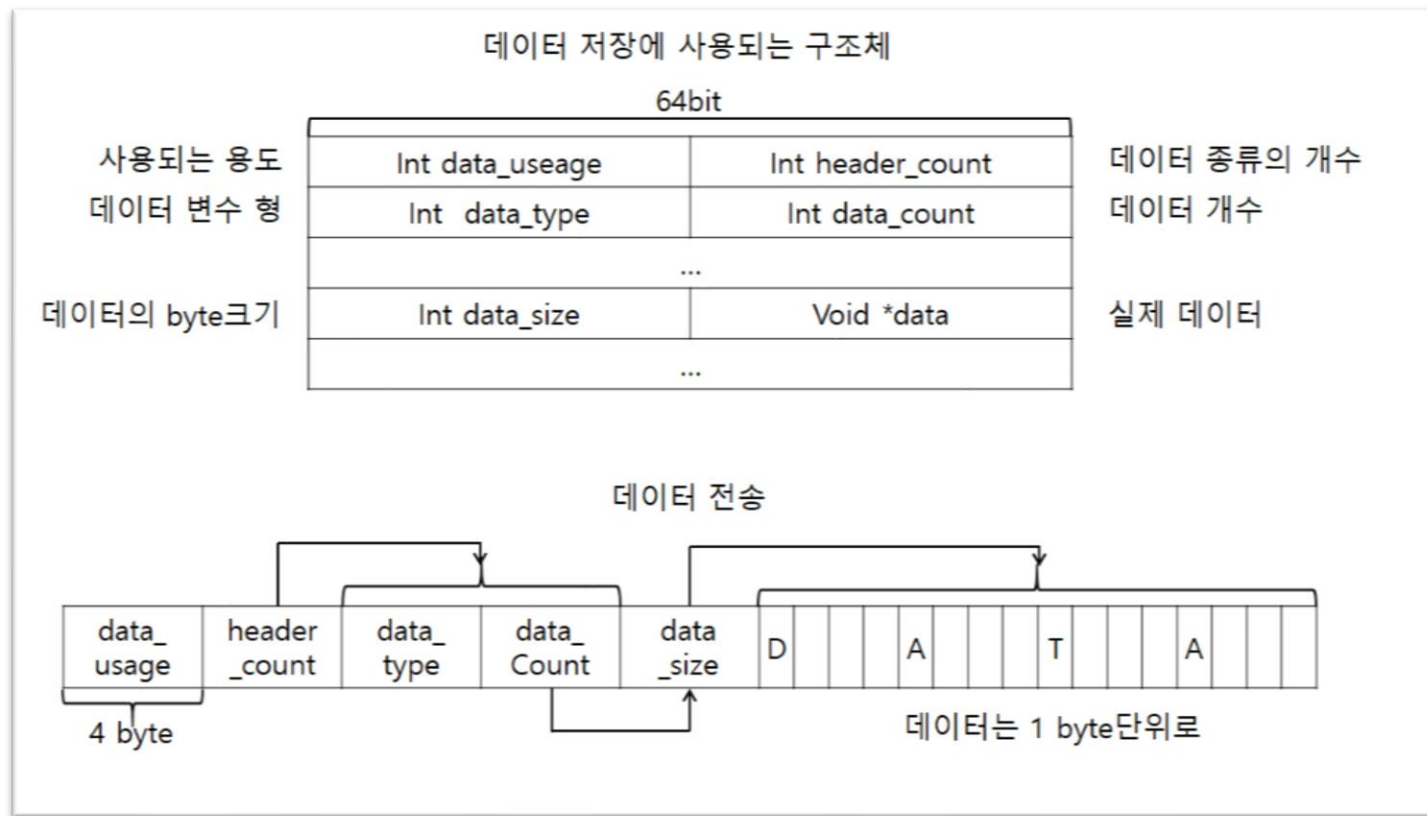


로봇 클라이언트 - 서버 간 구조

TOBE 시스템 기능 구상도



멀티프로세스 기반 서버

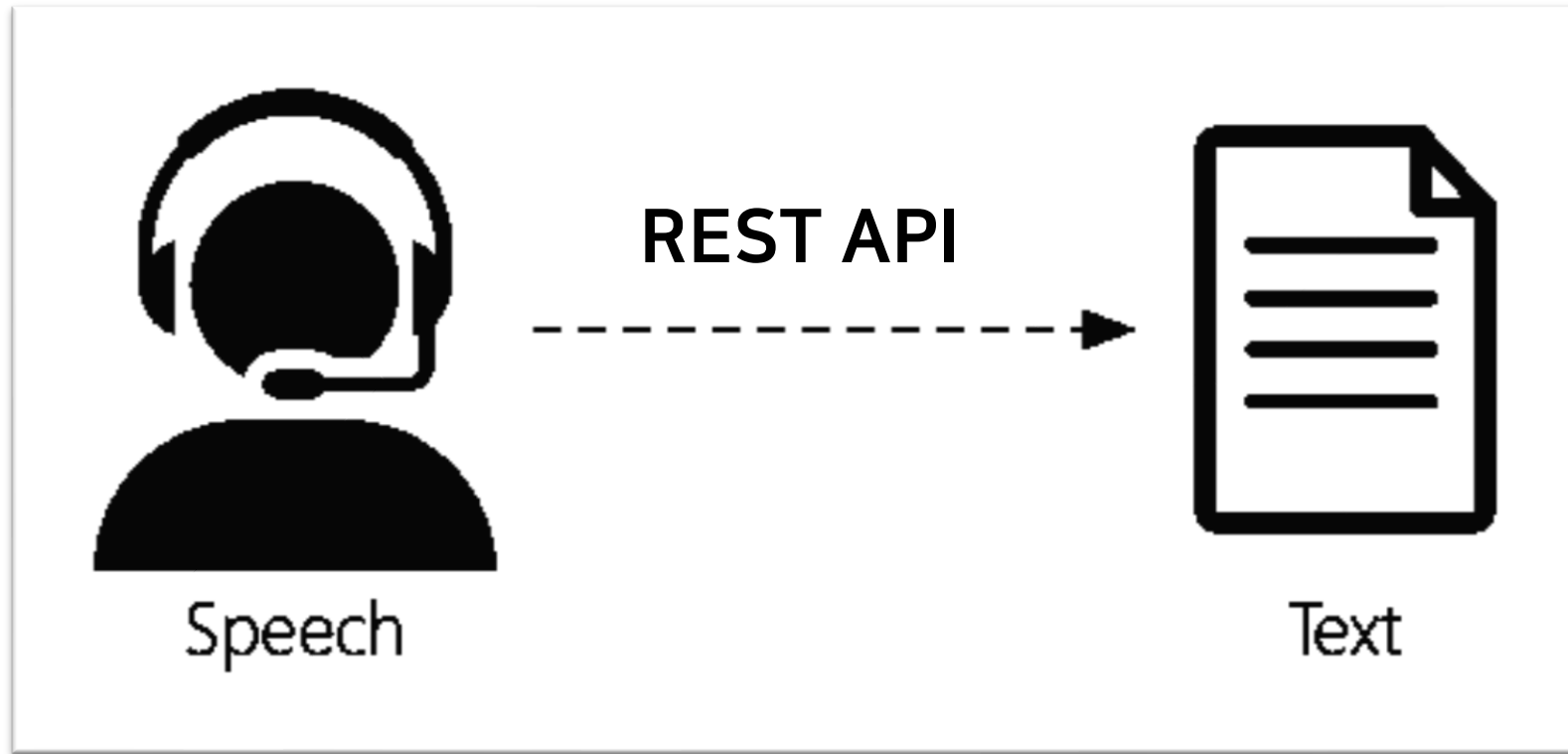


통신에 사용할 **프로토콜** 제작

```
simkyuwon@simkyuwon-VirtualBox: ~/VM_share/NetworkPrograming/netv
0000900: 0100 0000 0000 0000 0000 0000 0000 0000 TTTT .....
0000910: 0000 0000 0000 0000 0000 0000 0100 0000 .....
0000920: 0000 0000 0000 0000 0000 ffff ffff 0000 .....
0000930: ffff 0000 0000 0000 0000 0000 0000 ffff .....
0000940: 0100 0000 0000 ffff 0100 0000 ffff ffff .....
0000950: 0000 ffff 0200 fdff f8ff 0800 d5ff 2700 .....
0000960: 8cff 2f00 61ff b5ff eaff ccfd ce04 7902 ../.a.....y.
0000970: fffc 7f00 a103 b506 b209 f807 24ff d5f9 .....$.
0000980: baf7 baf8 a2ff 0107 c30b 4b0d 1902 6d00 .....K...m.
0000990: 67fa 49f0 84f5 11f6 f1fc 9904 4b0d 6e12 g.I.....K.n.
00009a0: 410c 0508 c805 e1f4 0bea dae3 09e5 c8f2 A.....
00009b0: 31f4 6602 910a 9913 d619 e01c 5f20 5a0a 1.f....._ Z.
00009c0: 890b 3400 dee6 15e3 38dd a0e4 f5ec 3bf1 ..4.....8....;.
00009d0: adfc 560a 0417 a124 2124 911a 801a c616 ..V....$!$.
00009e0: 6e0f bb00 b6fb 8af8 58ef 83fd aff6 3dfe n.....X.....=.
00009f0: 0606 0bf2 e500 31f2 0be7 99e7 02dc 78e9 .....1.....x.
0000a00: 1ced 03f4 b702 4310 ec0c fe14 5623 2b1c .....C.....V#+.
0000a10: 4a18 cf05 8504 24f8 ddf2 3903 0bf4 130d J....$.9....
0000a20: a209 5306 821f e901 930f df09 f9fb 230d ..S.....#.
0000a30: 9dfa e100 7eff 71fb e6ff 9cfb 86fc a402 ....~.q.....
0000a40: 7411 e908 8511 3611 ff04 ac07 60f1 03e9 t....6.....`...
0000a50: a7e5 52e8 c0e8 03e9 bef8 baf9 2411 950f ..R.....$.
0000a60: a401 e812 2f04 9800 f509 00fa aaf4 04f2 ..../.
0000a70: 38d6 cec6 34d0 15c9 3ad4 dae9 57f2 6407 8...4.....W.d.
0000a80: 3c12 5f1e db24 d220 fd16 4709 7900 24ec <...$. ..G.y.$..
0000a90: eff0 7bed 04e1 91f9 27f8 b4f2 6afc 0ded ..{.....'...j...
0000aa0: 23f7 c1fd f6e4 afee dce8 02d5 47e8 6eec #.....G.n.
0000ab0: 54f2 c10b b004 a208 0b0b 63fb cfff 52f5 T.....C...R.
0000ac0: 8d02 6c05 26fc 7c11 a310 6519 b424 6f1f ..l.&|...e..$.
0000ad0: 5f29 892e 7027 8327 bd1a 4112 c317 6f10 _)..p'...'..A...o.
0000ae0: aa08 8306 b200 96f9 e4fa 75ff 8806 350d .....u...5.
0000af0: 0c12 011b cb19 d510 0611 4e0a 6204 1204 .....N.b...
0000b00: e5ff 9ffd 16f9 1cfd 0905 8002 11fb d602 .....

[send way] [Modified] 5891 lines --3%--
```

해당 프로토콜로 파일 전송 및 파싱



Microsoft Azure Cognitive를 이용한 **Speech to Text**

```
y_text_list = list(map(lambda j : j[1],train.nouns()))
vectorizer.fit_transform(x_text_list)
gsvc.fit(vectorizer.transform(x_text_list), y_text_list)
pred_list = gsvc.predict(vectorizer.transform(test_list))
```

```
for now,now2 in zip(pred_list,testW):
    print("예측 : ",motionDict[now],"\t 명령어 : ",now2)
```

| | |
|----------|----------------|
| 예측 : 뒤걸기 | 명령어 : 앞으로 가자 |
| 예측 : 뒤걸기 | 명령어 : 뒤로 가봐 |
| 예측 : 좌걸기 | 명령어 : 왼쪽으로 걸어 |
| 예측 : 좌회전 | 명령어 : 왼쪽으로 회전 |
| 예측 : 좌회전 | 명령어 : 좌 회전 |
| 예측 : 우걸기 | 명령어 : 오른쪽으로 걸어 |
| 예측 : 우회전 | 명령어 : 오른쪽으로 회전 |
| 예측 : 공격 | 명령어 : 앞 때려 |
| 예측 : 공격 | 명령어 : 공격해 |
| 예측 : 좌걸기 | 명령어 : 왼쪽 가즈아 |

KoNLPy를 이용한 **명사, 동사** 추출

```
def NounVerb(rawText):
    kkma=Kkma().pos(rawText)
    tw=Twitter().pos(rawText)
    returnText = ''
    for Kpos,Tpos in zip(kkma,tw):
        if Tpos[1] == 'Noun' or Tpos[1] == 'Verb':
            returnText+=Tpos[0] + ' '
    # if Kpos[1][0] == 'N' or Kpos[1][0] == 'V':
    #     returnText+=Kpos[1][0] + ' '
    return returnText

if __name__ == "__main__":
    print(NounVerb('뿌리 깊은 나무는 바람에 아니 흔들려서 꽃 좋고 열매 많으니'))
    print(NounVerb('울타리 옆으로 맑은 물이 흐른다'))
    print(NounVerb('왼쪽으로 돌아'))
```

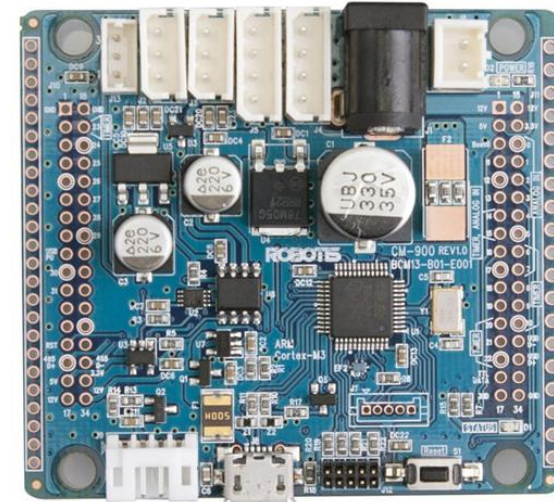
뿌리 나무 바람 흔들려 꽃 열매
울타리 옆 맑은 물이 흐른
왼쪽 돌아

SVM을 이용한 **지도학습**으로 명령어 **예측**

| Pin# | NAME | | NAME | Pin# |
|------|-----------------------|---|-----------------------|------|
| 01 | 3.3v DC Power | ⬢ | DC Power 5v | 02 |
| 03 | GPIO02 (SDA1 , I2C) | ⬢ | DC Power 5v | 04 |
| 05 | GPIO03 (SCL1 , I2C) | ⬢ | Ground | 06 |
| 07 | GPIO04 (GPIO_GCLK) | ⬢ | (TXD0) GPIO14 | 08 |
| 09 | Ground | ⬢ | (RXD0) GPIO15 | 10 |
| 11 | GPIO17 (GPIO_GEN0) | ⬢ | (GPIO_GEN1) GPIO18 | 12 |
| 13 | GPIO27 (GPIO_GEN2) | ⬢ | Ground | 14 |
| 15 | GPIO22 (GPIO_GEN3) | ⬢ | (GPIO_GEN4) GPIO23 | 16 |
| 17 | 3.3v DC Power | ⬢ | (GPIO_GEN5) GPIO24 | 18 |
| 19 | GPIO10 (SPI_MOSI) | ⬢ | Ground | 20 |
| 21 | GPIO09 (SPI_MISO) | ⬢ | (GPIO_GEN6) GPIO25 | 22 |
| 23 | GPIO11 (SPI_CLK) | ⬢ | (SPI_CE0_N) GPIO08 | 24 |
| 25 | Ground | ⬢ | (SPI_CE1_N) GPIO07 | 26 |
| 27 | ID_SD (I2C ID EEPROM) | ⬢ | (I2C ID EEPROM) ID_SC | 28 |
| 29 | GPIO05 | ⬢ | Ground | 30 |
| 31 | GPIO06 | ⬢ | GPIO12 | 32 |
| 33 | GPIO13 | ⬢ | Ground | 34 |
| 35 | GPIO19 | ⬢ | GPIO16 | 36 |
| 37 | GPIO26 | ⬢ | GPIO20 | 38 |
| 39 | Ground | ⬢ | GPIO21 | 40 |

Rev. 1.1
16/07/2014

<http://www.element14.com>



서버에서 받은 **데이터**를 **제어기**(로봇)에 전송(Serial)

- 한 로봇이 여러 명령을 **동시**에 처리할 수 없다
 - ✓ **명령어 큐**를 만들어 순차적으로 처리할 수 있도록 설계한다.
- 음성인식 **API 서비스가 불안정** 할 수 있다
 - ✓ **대체로 사용할** 수 있는 음성인식 API를 추가한다

음성인식 및 발음평가 API 서비스 일시중단 안내



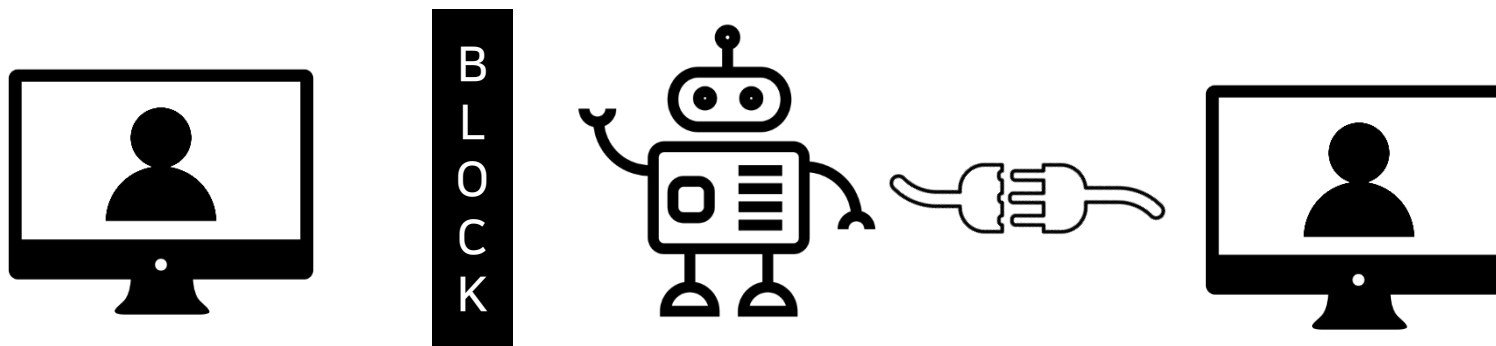
서버 재정비 작업으로 음성인식 및 발음평가 API 서비스를 일시적으로 중단하오니 이용에 참고부탁드립니다.

* 서비스 중단 일시 : 2018년 4월 6일(금) 오후 6시부터

* 서비스 중단 API : 음성인식 및 발음평가 API (언어분석 API는 계속 이용 가능)

* 서비스 재개 : 2018년 6월(예정)

- 로봇이 배터리로 동작하므로 클라이언트 서버가 **강제 종료**되는 일이 잦을 수 있다
 - ✓ **시그널과 timeout**을 통해 클라이언트 **종료를 감지**하여 처리한다.
- 이미 사용 중인 로봇에 대해 **교착상태**가 일어날 수 있다
 - ✓ **Blocking**을 이용해 사용 중인 로봇에 대한 접근을 막는다.



Risk 및 일정 관리

| | 항목 | 4월 | | | | 5월 | | | | 6월 | | | |
|-------|----------------------|----|---|---|------|----|---|---|---|----|---------------------------|---|---|
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 서버 | PC클라이언트-서버간 소켓 통신 구현 | | | | 중간고사 | | | | | | 최종 점검 및 발표 준비 | | |
| | 라즈베리파이-서버간 소켓 통신 구현 | | | | | | | | | | | | |
| | 클라이언트에서 받은 데이터 가공 | | | | | | | | | | | | |
| | DB구현 및 서버 연동 | | | | | | | | | | | | |
| | STT API 연동 | | | | | | | | | | | | |
| | 자연어 처리 알고리즘 구현 | | | | | | | | | | | | |
| 클라이언트 | 머신러닝 알고리즘 구현 | | | | | | | | | | | | |
| | 라즈베리파이 환경 세팅 | | | | | | | | | | | | |
| | 데이터 입력 및 전송 구조 구현 | | | | | | | | | | | | |
| | 로봇 모션 제작 | | | | | | | | | | | | |
| DB | 로봇 클라이언트 제작 | | | | | | | | | | | | |
| | 로봇 리스트 작성 | | | | | | | | | | | | |
| | 모션 종류 및 프로토콜 작성 | | | | | | | | | | | | |

Thank you

Contact

- Github : [inerplat/Wedge-Robot.git](https://github.com/inerplat/Wedge-Robot)
- 심규원 : tlarbvkf@naver.com
- 김동현 : inerplat@gmail.com