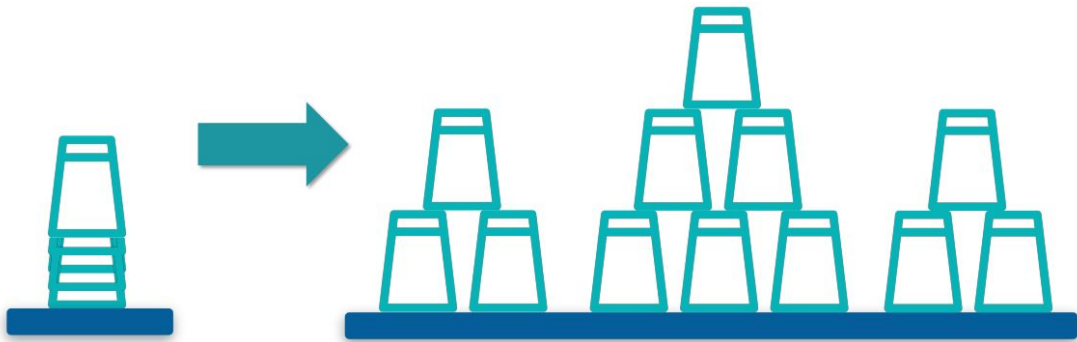


Sports Stacking

: ROS2활용 협동로봇 동작 운용 프로젝트



A-4조

팀원: 류승기, 이선우, 최민호

목차

1. 프로젝트 개요
2. 프로젝트 해결 과제 정의
3. **Core** 알고리즘 및 차별성
4. 문제점 및 해결방법
5. **Lesson Learn**
6. 구성원 역할

Week5: 협동로봇 동작 운영 실습 프로젝트 개요

	실습 항목	
1 ~ 4일차	<ul style="list-style-type: none"> ▪ Doosan Robot ROS 패키지(설치) 및 시스템 구성 설명 ▪ MoveL, MoveJ, Grip, Release, Force, Periodic 실습 	<ul style="list-style-type: none"> ▪ Getting_position, force_control, mode_periodic연습 ▪ 현재 좌표값 모니터링, 힘제어, Periodic 실습
	<ul style="list-style-type: none"> ▪ 실습 <div>     </div> <ul style="list-style-type: none"> ▪ Sports Stacking 평가 과제 공개 	
5일차	<ul style="list-style-type: none"> ▪ Sports Stacking 평가 과제 실습 	<ul style="list-style-type: none"> ▪ 조별 발표 및 평가

Week5: 협동로봇 동작 운영

실습

프로젝트 해결 과제 정의

: Cup Stacking & Reverse Stacking

프로젝트 목표

: 11개의 Cup Stacking을 3분내에 진행

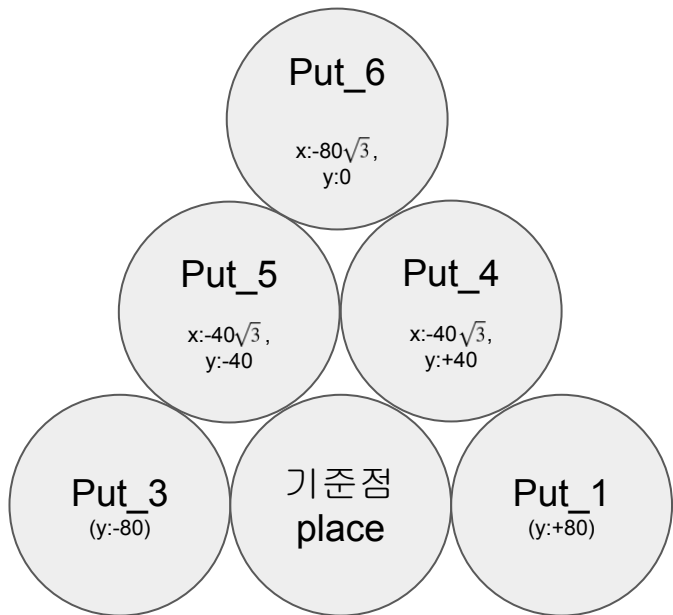
해결과제

- Way Point 설정 및 Teaching
 - 기존 Way Point 설정 후, 상대좌표를 활용하여 로봇의 Way Point를 효율적으로 Teaching
- 정확한 Gripping 제어
 - 최적화된 그리퍼 너비 및 힘을 조정하여 Gripping Point를 정밀하게 제어
- Force Control(힘 제어) 적용
 - Force Control을 활용하여 컵을 정확한 위치에 배치
- 제한 시간 내 Stacking 및 충돌 방지
 - 제한된 시간 내에 Stacking을 완료하고, 컵을 치지 않도록 Movesx 기반의 최적화된 움직임 구현

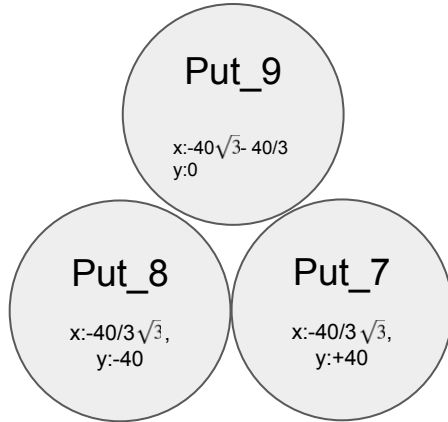
프로젝트 해결 과제 정의

: Cup Stacking & Reverse Stacking

1층: Put_6bottom



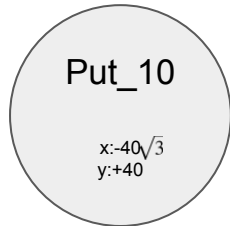
2층: Put_3middle



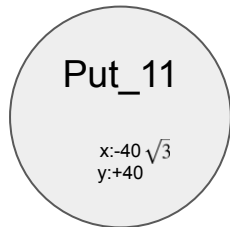
$z = 94$

$z = 189$

3층: Put_top



4층: Put_reverse



$z = 250$

Week5: 협동로봇 동작 운영

실습

Core 알고리즘 :

1. 그리퍼 설정 및 제어

OnRobot WebCLIENT v6.2.4

Devices
Configuration
WebLogic™
Paths
Update
TCP/CoG
Licenses
Applications

OnRobot WebLogic™

This page allows to browse/manage the OnRobot WebLogic™ programs. You can create new program and run it on the Editor tab. (To make your program run automatically on power-on, leave it running while powering the Compute Box off.)

[Browser](#) [Program Editor](#)

doosan Program is running NEW SAVE STOP

Program is running

#0

SELECT DEVICE ▾

SELECT DEVICE ▾

RG2-0 ▾

IN 1 2 3 4 5 6 7 8 AND (No device selected AND No device selected) → Width (50 mm, 20 N) OUT 1 2 3 4 5 6 7 8 ▾

#1

SELECT DEVICE ▾

SELECT DEVICE ▾

RG2-0 ▾

IN 1 2 3 4 5 6 7 8 AND (No device selected AND No device selected) → Width (75 mm, 15 N) OUT 1 2 3 4 5 6 7 8 ▾

#2

RG2-0 ▾

RG2-0 ▾

SELECT DEVICE ▾

IN 1 2 3 4 5 6 7 8 AND (Width <= 65 mm AND Grip == TRUE) → No device selected OUT 1 2 3 4 5 6 7 8 ▾

#3

RG2-0 ▾

SELECT DEVICE ▾

SELECT DEVICE ▾

IN 1 2 3 4 5 6 7 8 OR (Width >= 70 mm AND No device selected) → No device selected OUT 1 2 3 4 5 6 7 8 ▾

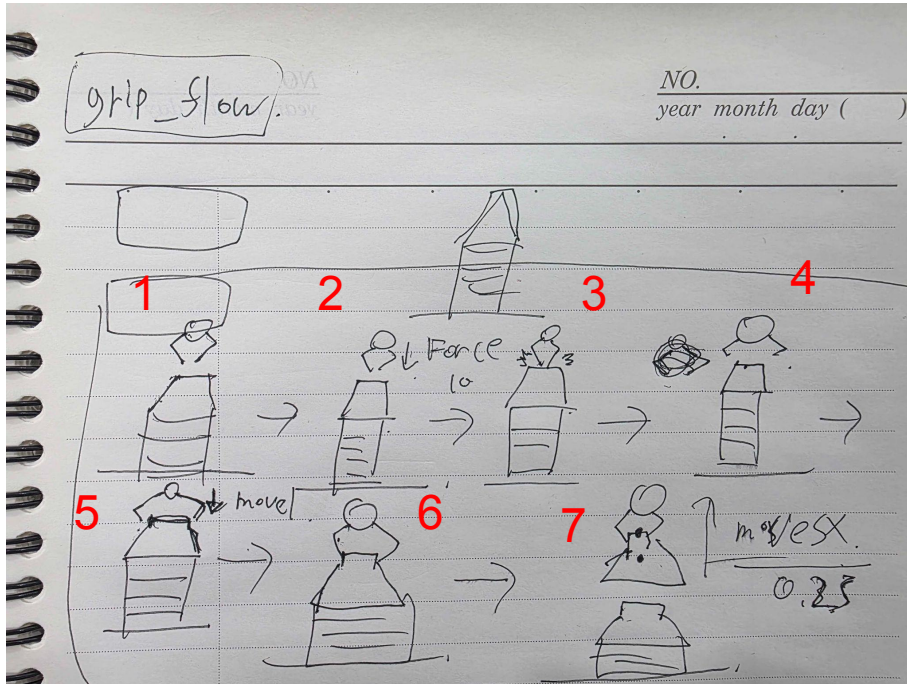
+ Add new conditions and commands

Copyright © 2025 OnRobot A/S

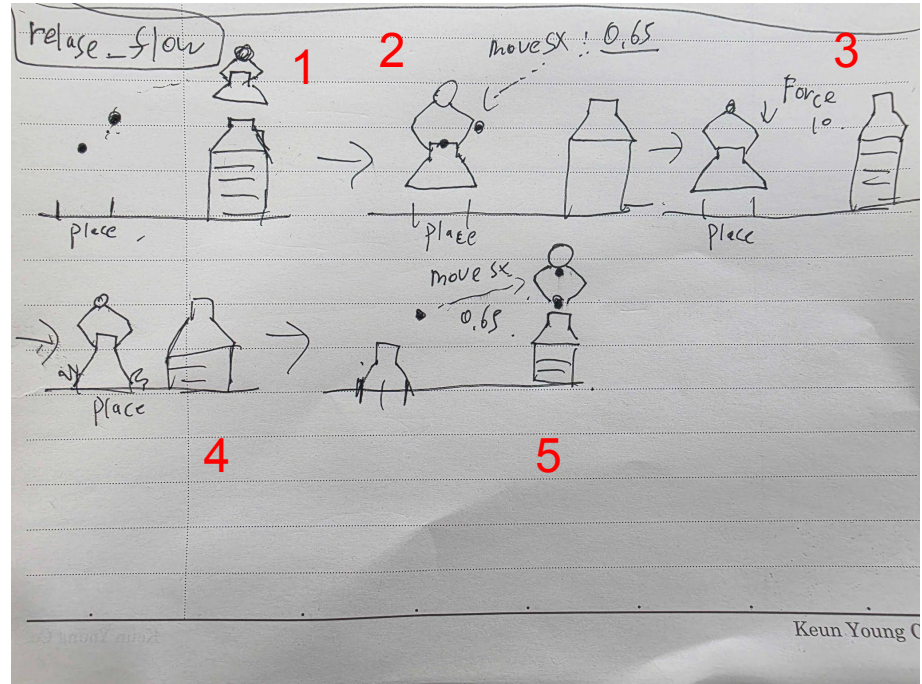
Core 알고리즘:

1. 그리퍼 설정 및 제어

def Grip_flow()



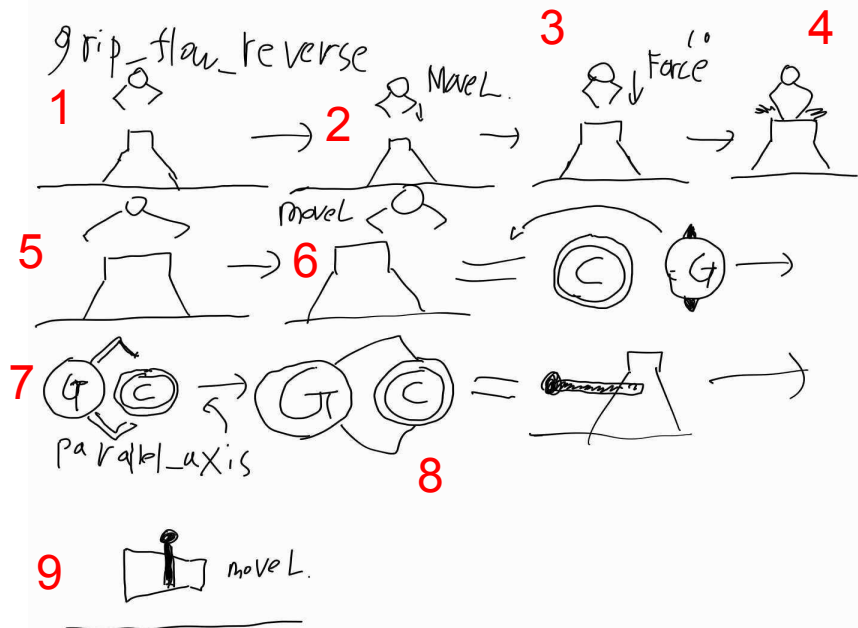
def Release_flow()



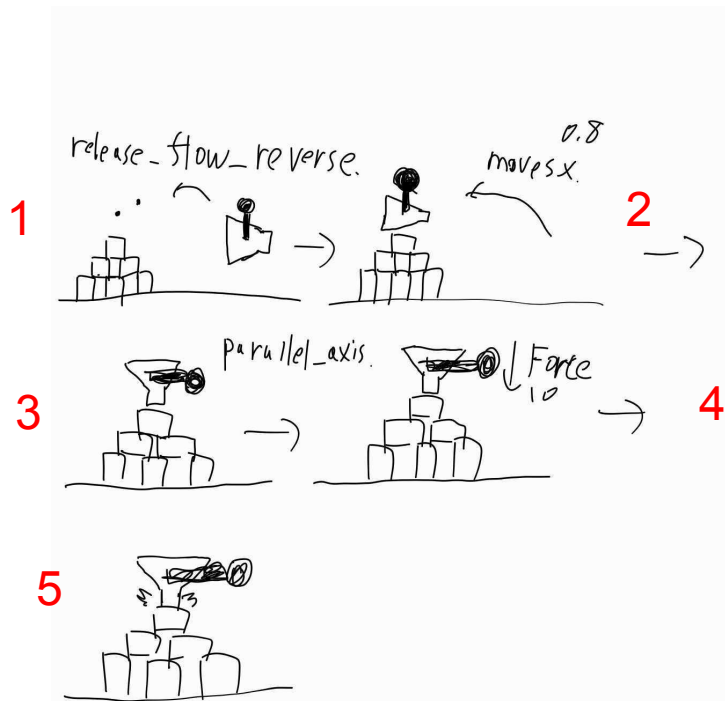
Core 알고리즘:

1. 그리퍼 설정 및 제어

def Grip_flow_reverse()



def Release_flow_reverse()

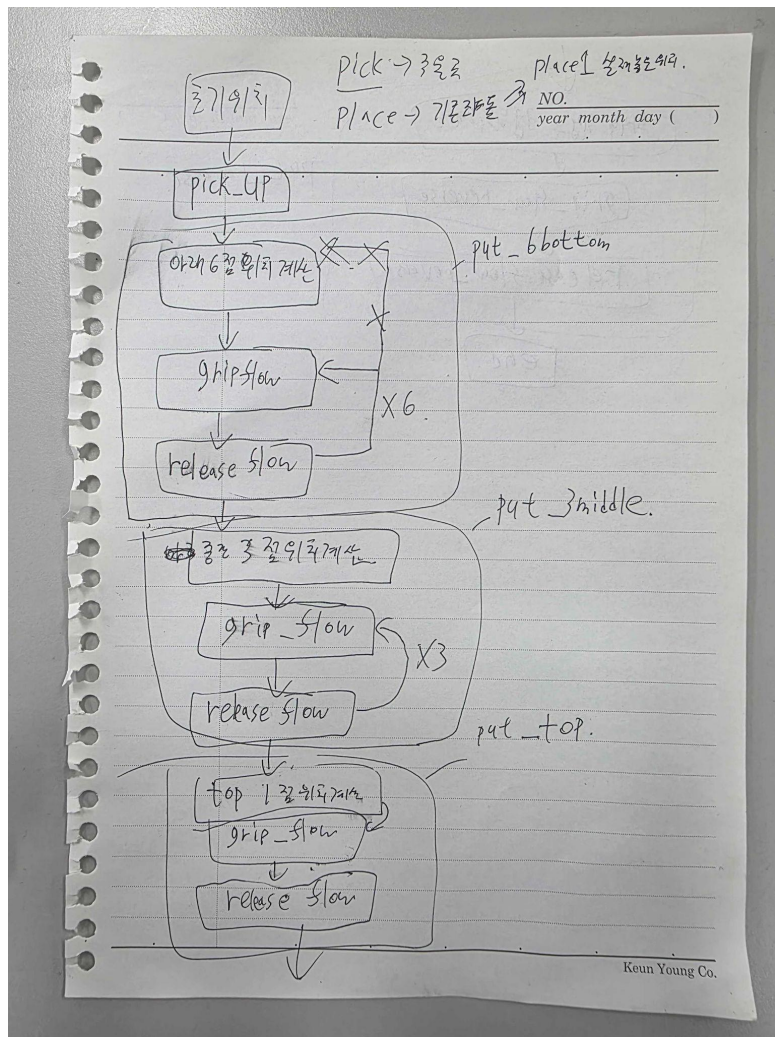


Week5: 협동로봇 동작 운영

실습

Core 알고리즘 :

2. put_6bottom > put3_middle > put_top

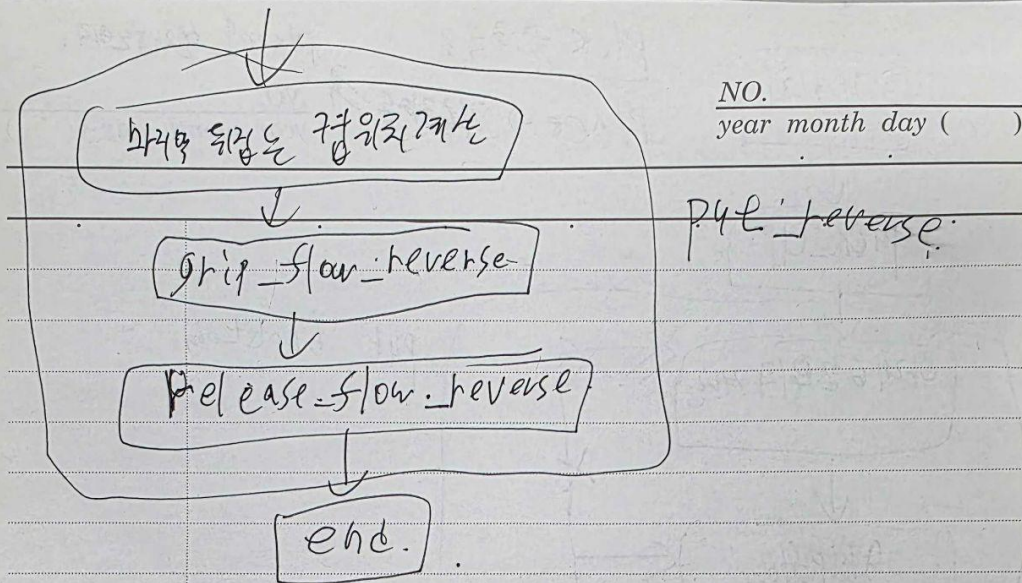


Week5: 협동로봇 동작 운영

실습

Core 알고리즘 :

3. put_reverse > grip_flow_reverse > release_flow_reverse



Week5: 협동로봇 동작 운영

실습

Core 알고리즘 : <차별성>

1. 기능 파일과 알고리즘 파일 분류

10개 탑쌓기 -> put_6bottom > put3_middle > put_top

1개 뒤집어서 쌓기 -> put_reverse

```
cup_stand_11.py M x
ws > src > homework > homework5_Final > cup_stand_11.py > main
16 def main(args=None):
45
46 > except ImportError as e:--
49     set_tool("Tool Weight")
50     set_tcp("GripperDA_v1")
51     # 초기 위치
52     JReady = posx([367.23846435546875, 3.2307586669921875, 220.86367797851562, 82.5703506
53     pick1 = posx([266.2298889160156, 24.937015533447266, 205.596, 141.998, 179.98, 140.07
54     place1 = posx([525.650146484375, -1.451927900314331, 84.67518615722656, 141.998, 179
55     count_cup = 0
56     while rclpy.ok():
57         # 초기 위치로 이동
58         move(JReady, vel=VELOCITY, acc=ACC, ref=DR_BASE)
59         move(pick1, vel=VELOCITY, acc=ACC, ref=DR_BASE)
60
61         count_cup = util_hw5.put_6bottom(place1,pick1,count_cup )
62         count_cup = util_hw5.put_3middle(place1, pick1, count_cup)
63         count_cup = util_hw5.put_top(place1, pick1, count_cup)
64         print(f"end {count_cup}")
65         util_hw5.put_reverse(place1,pick1,count_cup)
66         break
67     rclpy.shutdown()
68     print("end")
69
70
71 if __name__ == "__main__":
72     main()
73
```

```
util_hw5.py x
ws > src > homework > homework5_Final > util_hw5.py > put_6bottom
17 print("start util")
18 > try:--
48     except ImportError as e:
49         print(f"Error importing DSR ROBOT2 : {e}")
50
51
52 > def wait_digital_input(sig_num):--
57
58 > def release():--
62
63 > def grip():--
68
69 > def grip_without_wait():--
73
74
75 > def grip_flow(pick1,count):--
107
108 > def release_flow(place1,size,pick,count):--
132
133 > def put_6bottom(place, pick, count_cup):--
160
161 > def put_3middle(place, pick, count_cup):--
183
184 > def put_top(place, pick, count_cup):--
195
196 > def grip_flow_reverse(pick1,count):--
233
234 > def release_flow_reverse(place1):--
259
260 > def put_reverse(place,pick,count_cup):--
273
270 print("end util")
273
```

Week5: 협동로봇 동작 운영

실습

Core 알고리즘 : <차별성>

2. bash 파일 생성

1. get_position.sh: 현재 로봇의 6개 위치 좌표를 Base 좌표계 기준으로 조회

```
ws > $ get_position.sh
1   ros2 service call /dsr01/aux_control/get_current_posx dsr_msgs2/srv/GetCurrentPosx "{ref: 0}"
```

2. go_home.sh: 현재 로봇의 6개 위치 좌표를 Base 좌표계 기준으로 조회

```
ws > $ go_home.sh
1   ros2 service call /dsr01/motion/move_joint dsr_msgs2/srv/MoveJoint "{pos : [0.0, 0., 90., 0., 90., 0], vel : 30, acc : 100}"
```

3. grip.sh & release.sh: 그리퍼의 close, release를 진행

```
ws > $ grip.sh
1   ros2 service call /dsr01/io/set_ctrl_box_digital_output dsr_msgs2/srv/SetCtrlBoxDigitalOutput "{index: 1, value: 1}"
2   ros2 service call /dsr01/io/set_ctrl_box_digital_output dsr_msgs2/srv/SetCtrlBoxDigitalOutput "{index: 2, value: 0}"
```

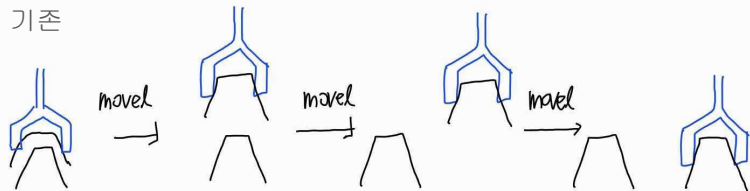
```
ws > $ release.sh
1   ros2 service call /dsr01/io/set_ctrl_box_digital_output dsr_msgs2/srv/SetCtrlBoxDigitalOutput "{index: 1, value: 0}"
2   ros2 service call /dsr01/io/set_ctrl_box_digital_output dsr_msgs2/srv/SetCtrlBoxDigitalOutput "{index: 2, value: 1}"
```

4. z_parrel.sh: 로봇 z축 정렬

```
ws > $ z_parrel.sh
1   ros2 service call /dsr01/force/parallel_axis2 dsr_msgs2/srv/ParallelAxis2 "{vect: [0, 0, -1], axis: 2, ref: 0}"
```


Week5: 협동로봇 동작 운영 실습 문제점 및 해결방법

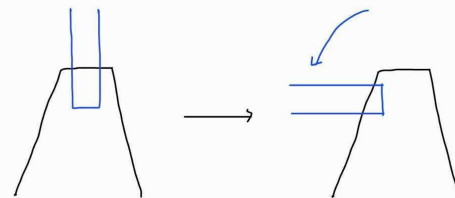
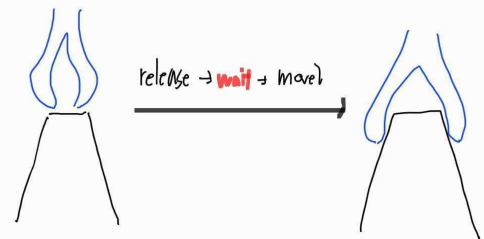
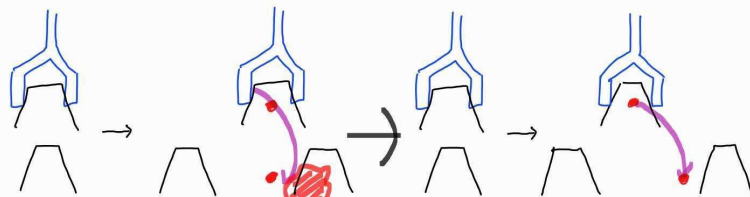
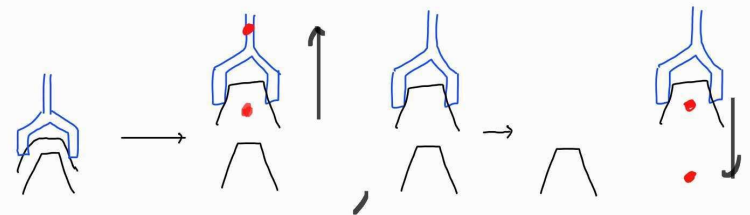
기존



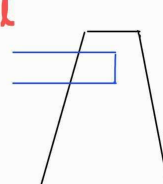
⇒ move1x (time설정)

속도↑

```
move1x(xlist, vel=[VELOCITY, VELOCITY], time=0.8, acc=[ACC, ACC], vel_opt=DR_MVS_VEL_NONE)
```



move1



```
pos_1 = get_current_posx()
pos_1 = pos_1[0]
pos_1[2] -= 20
pos_1[1] += 20
print("current position1 : ", pos_1)
release()
print("current release : ", pos_1)
release_compliance_ctrl()
move1(pos_1, vel=VELOCITY, acc=ACC, ref=DR_BASE)
```

Lesson Learn:

1. moveI보다 movesx 를 통한 시간 단축
2. 힘 제어를 사용할 때, 힘 제어를 푼 후, `get_current_posx()`를 사용하면 안된다.
3. 문제 정의에 대한 최적화를 진행하면서 발생한 여러 문제를 해결해 나아가며 문제해결 능력 향상

구성원 역할:

- 이선우 : 코드작성, 테스트
- 류승기 : 알고리즘 설계, 테스트
- 최민호 : 알고리즘 설계, 테스트