

TETRA-DS V 매뉴얼

TETRA-DS V



HYULIM NETWORKS

자율주행로봇 (AMR)

CONTENTS

Chapter 1. TETRA-DS V (What is TETRA-DS V?)	6
1-1. TETRA-DSV 사양(TETRA-DSV Specifications)	6
1-2. TETRA-DSV구성품 (TETRA-DSV Packages)	7
1-3. 기술지원 (Technical Support).....	8
Chapter 2. 기계적 사양 (Mechanical Hardware Specifications)	9
2-1. 마운트판 (Mount Plate)	11
2-2. 모터및엔코더 (Motors and Encoders).....	12
2-3. 캐스터 (Casters)	12
Chapter 3. 전기적 사양 (Electrical Hardware Specifications).....	13
3-1. LiDAR센서 (SICK TIM571).....	13
3-1-1전방 LiDAR센서(SICK TIM571).....	13
3-1-2후방 LiDAR센서(CYGBOT CygLiDAR D1).....	14
3-2. RGBD카메라 (Intel RealSense D435).....	14
3-3. 자동충전용카메라.....	16
3-4. 전방범퍼	17
3-5. 배터리상태표시 LED (Battery Status LED).....	17
3-6. 좌, 우상태표시 LED (Left, Right Status LED)	18
3-7. 주전원스위치 (Main Power Switch).....	18
3-8. 비상정지버튼 (Emergency Button)	19
3-9. 수동충전커넥터	20
3-10. 자동충전단자	21
3-11. 전원/센서보드 (Power/Sensor Board)	21
3-12. 구동모듈 (Drive Board)	23
3-13. 인터페이스보드전원포트	25
3-14. 인터페이스보드신호포트	26
Chapter 4. 제어 하드웨어 프로토콜 (Control Hardware Protocol)	29
4-1. 통신규약(Rules of Communication).....	29
4-2. 통신 Packet 운영방법 (Packet Communication Methods of Operation)	29
4-3. 전원/센서모듈프로토콜 (Power Sensor Board Protocol)	30
4-3-1Power Sensor BD status Read	31
4-3-2 TETRA Status Read.....	32
4-3-3 Analog Port Data Read	33
4-3-4 Power Sensor BD Regulator Power ON/OFF	34
4-3-5 Single Power Sensor BD Regulator Power ON/OFF.....	35
4-3-6 Wheel Driver BD Power Relay ON/OFF.....	35

4-3-7 Out Port ON/OFF	35
4-3-8 OUT Port Single Control ON/OFF.....	36
4-3-9 IN Port & OUT Port status Read	36
4-3-10 Sonar Sensor Loop Enable.....	36
4-3-11 Sonar Sensor Data Read	37
4-3-12 Power Sensor BD Parameter Data Write	37
4-3-13 Parameter Data Flash Memory Write.....	39
4-3-14 Parameter Data Read.....	39
4-3-15 LED Brightness Toggle Data Write	39
4-3-16 Display Toggle LED Select.....	40
4-3-17 Cumulative current Read	41
4-3-18 Cumulative voltage Read	42
4-3-19 Cumulative buffer erase	43
4-3-20 Cumulative buffer erase	43
4-4. 구동보드프로토콜 (Drive Board Protocol)	43
4-4-1구동보드파라미터일람표 (Drive Board Parameter Chart)	44
4-4-2 Drive status Read.....	45
4-4-3 Encoder Position Read.....	47
4-4-4 Coordinates Read.....	47
4-4-5 Coordinates Change.....	47
4-4-6Motion	48
4-4-7 Velocity Control (Speed Mode).....	49
4-4-8 Velocity Control (Position Mode)	49
4-4-9 Error Reset.....	49
4-4-10 Servo ON/OFF	49
4-4-11 Control Mode Change.....	49
4-4-12Parameter Read	50
4-4-13 Parameter Write	50
4-4-14Parameter Save	50
4-5.프로토콜 실행 예제 (Example of Protocol execution)	51
Ex1) 현재위치에서 1[m] 직진후 90도우회전명령내리기.....	51
Ex2) 위치제어모드에서모터제어하기	53
Ex3) 속도제어모드에서모터제어하기	55
Ex4) 원하는가속도, 밝기, 색상으로 LED 제어하기	57
Ex5) Sonar 제어하기.....	59
Ex6) Bumper 센서동작상태확인하기	61
Ex7) EMS 스위치동작상태확인하기	63
Ex8) I.O PORT를사용하여 RELAY 제어하기.....	65
Ex9) Power Senser Board에공급되는전원제어하기	68
Chapter 5. 전송계층 프로토콜 TCP (Transmission Control Protocol)	72
5-1. TCP/IP모듈프로토콜요청값차트 (TCP/IP Module Protocol Request Chart).....	72
5-2. TCP/IP모듈프로토콜응답값차트(TCP/IP Module Protocol Response Chart)	73
5-3. TCP/IP모듈프로토콜실행예제(TCP/IP Module Protocol Example).....	75
Chapter 6. Software Set up.....	78
6-0. PC Setup.....	78
6-1. ROS Melodic 설치.....	80

6-2. Realsense SDK 설치 (SDK 2.42.0).....	82
6-3. Realsense ROS 설치 (v 2.2.22).....	84
6-4. Cartographer 설치.....	87
6-5. ROS Package 설치.....	89
6-6. USBRule 설정.....	90
6-6-1 usb cam(AR marker) rule 설정	90
6-6-2 IMU 설정	92
Chapter 7. SLAM	94
7-1. Node 실행	94
7-2. JoyStick 조작.....	94
7-3. LandMark 등록	95
7-4. 지도 저장.....	96
Chapter 8. Navigation	98
8-1. Node 실행	98
8-2. 위치 이동.....	99
8-2-1 2D Nav Goal.....	99
8-2-2 Joystick 사용.....	99
8-2-3 서비스 호출.....	100
8-3. 위치 저장.....	101
8-4. 위치 보정.....	103
8-4-1 2D Pose Estimate	103
8-4-2 서비스 호출.....	103
Chapter 9. ROS	105
9-1. TETRA-DSV ROS Service.....	105
9-2. TETRA-DSV ROS TOPIC.....	110
Chapter 10. TETRA DS V 싱글 관제 서비스 사용 가이드.....	113
10-1. TETRA APP 실행	113
10-2. APP 메인화면	114
10-2-1. APP 메인화면 – 지도상표시정보	115
10-2-2. APP 메인화면 – 로봇을특정위치로이동시키기	116
10-3. 설정	118
10-3-1. 설정 – 지도매핑	119
10-3-2. 설정 – 지도변경및관리	120
10-3-3. 설정 – 위치관리 – 위치편집	121
10-3-3 (1). 설정 – 위치관리 – 위치편집 - 클릭해서추가.....	122
10-3-3 (2). 설정 – 위치관리 – 위치편집 – 로봇위치로추가	123

10-3-3 (3). 설정 – 위치관리 – 위치편집 – 위치수정및삭제	124
10-3-3 (4). 설정 – 위치관리 – 감속구간.....	125
10-3-3 (5). 설정 – 위치관리 – 가상벽.....	126
10-3-4. 설정 – 태스크플랜.....	127
10-3-5. 설정 – 태스크매니저.....	128
10-3-5 (1). 설정 – 태스크매니저 – 태스크추가	129
10-3-5 (2). 설정 – 태스크매니저 – 태스크추가 - 플랜생성	130
10-3-5 (3). 설정 – 태스크매니저 – 태스크추가 – 플랜수정및삭제.....	131
10-3-6. 설정 – 로봇설정	132

Chapter 1. TETRA-DS V (What is TETRA-DS V?)

1-1. TETRA-DSV 사양(TETRA-DSV Specifications)

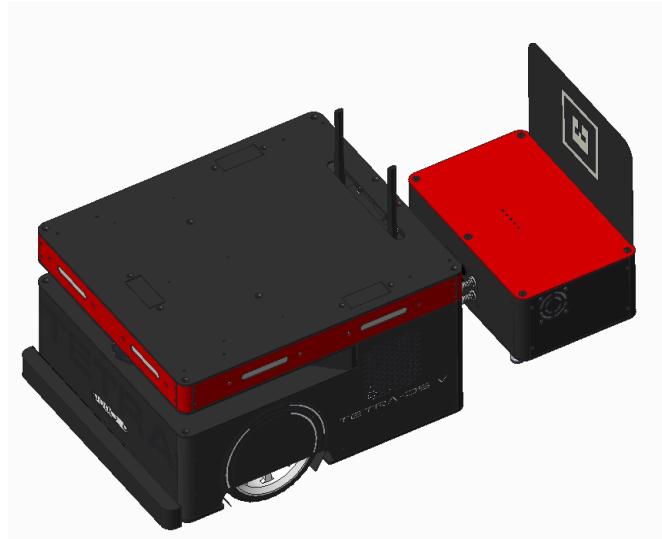
TETRA-DS5에 대한 사양은 표 1-1를 참고하시기 바랍니다.

PARTS	SECTION	CONTENTS
Mechanical	Dimensions /Turning radius	592mm * 490mm * 336mm / 906mm(회전반경)
	Robot weight /Max Payload	About 45kg / 100kg (평지기준)
Main System	Mini PC	Intel i7/32GB DDR4
	OS	Ubuntu 18.04LTS(Linux) / ROS Melodic
Actuator part	Type of Drive	2-Wheel Differential Drive (전방 2Wheel 구동, 후방 Caster[1EA], 후방보조 Caster[2EA])
	Motor	PMSM MOTOR(AC SERVO)
	Motor spec	100Watt / 23V ~ 29V
	Max Speed	1.5M/sec
Sensor Part	LiDAR	SICK TIM571 (256° 감지 Platform 설계)/TOF 시그넷
	Bumper / EMG	전방 1CH (최대 8CH) /EMG SW 1CH
	USB CAM	Camera for automatic charging induction
	RGBD Camera	INTEL D435
Power Part	Battery	Lithium ion7S12P 35Ah
	Run Time	35Ah 8시간 (PC 사양, 주행조건에 따라 변동)
	Charging method	Automatic Charge/10A, Manual Charge/10A
	Charge Time	35Ah: 200분 (85%까지 충전시간) 35Ah : 80% 180분

<표 1-1> Specifications of TETRA-DSV

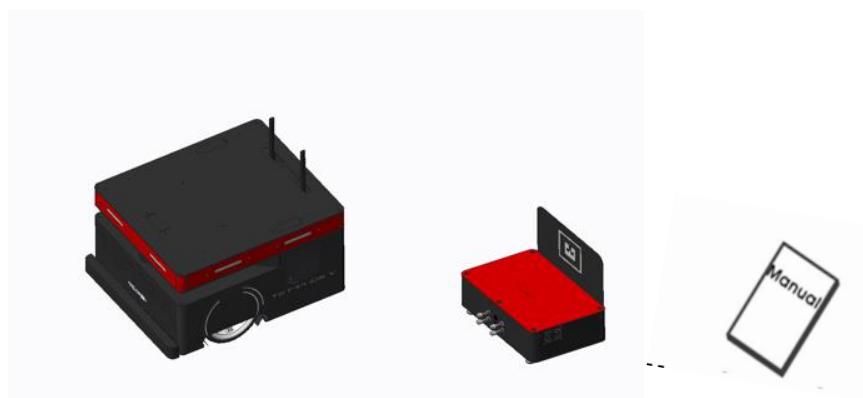
1-2. TETRA-DSV 구성품 (TETRA-DSV Packages)

TETRA-DSV는 실내환경에서 이동로봇의 자율주행 기술 개발을 위해 활용할 수 있는 고성능 이동로봇 플랫폼의 고유명칭입니다. 실내용 이동로봇의 자율주행 소프트웨어 개발에 활용할 수 있는 플랫폼의 이미지는 아래 그림 1-1에 나타난 바와 같습니다.



<그림 1-1> Image of TETRA-DSV(Basic Model)

TETRA-DSV의 구성품은 아래 그림 1-2에 나타난 바와 같으며, 사용자는 초기 개봉 후 구성품들이 제대로 들어있는지 바로 확인하시기 바랍니다. 플랫폼의 구성품 중 일부가 누락된 경우에는 바로 당사 고객지원센터로 문의하시기 바랍니다. 그림에 나타나 있는 구성품들은 모델 및 제작 상황에 따라 변경될 수도 있습니다.

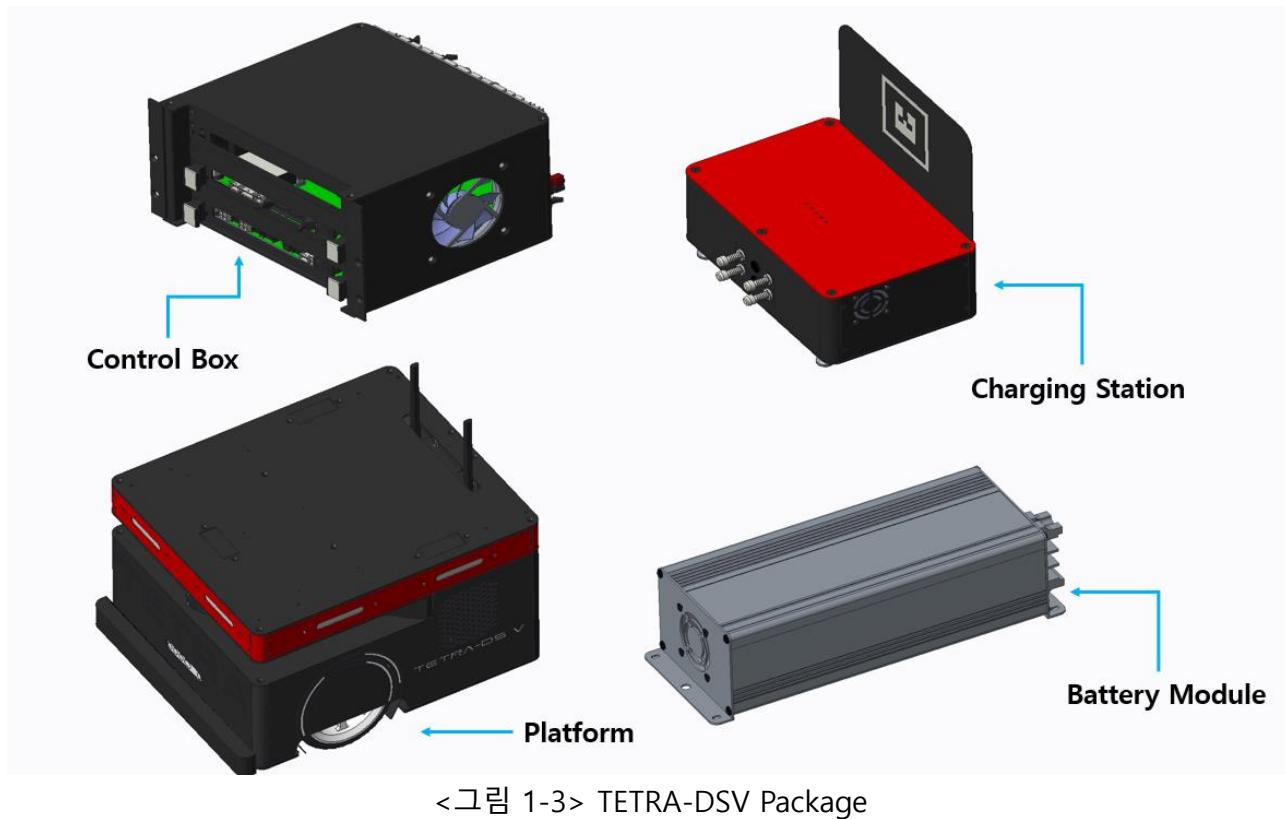


Platform Charging Station Operation Manuals

<그림 1-2> TETRA-DSV Package

- TETRA-DSV 본체 (Platform)
- 자동충전기 (Charging Station)
- 운용 매뉴얼 (Operation Manuals)

TETRA-DSV를 구성하는 모듈은 그림 1-3에 나타난 바와 같이 크게 본체와, 컨트롤박스, 배터리 모듈, 충전스테이션으로 구분되며, TETRA-DSV의 향후 업그레이드 및 손쉬운 유지보수를 위해 각각의 모듈화로 설계되어 있습니다.



1-3. 기술지원 (Technical Support)

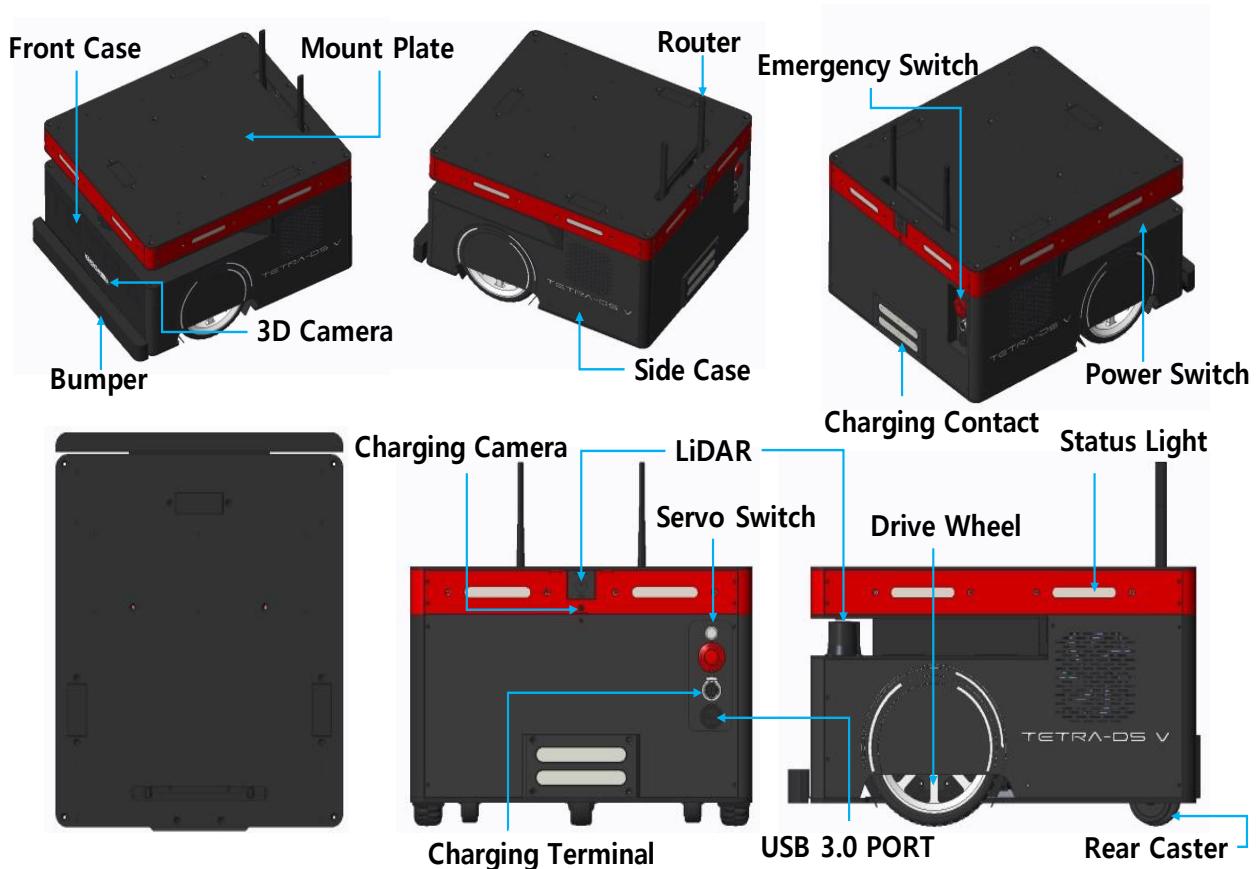
사용 상의 문제가 있으신가요? 제공된 운영 매뉴얼 상에서 해답을 찾을 수 없으신가요? 또는 당사 플랫폼의 성능 개선점을 발견하셨나요? 아래 온라인 웹사이트에서 당신의 생각과 의문점을 공유해 주시기 바랍니다.

<http://hyulimnetworks.com/contactus> 또는, 전화로 문의해 주시기 바랍니다.

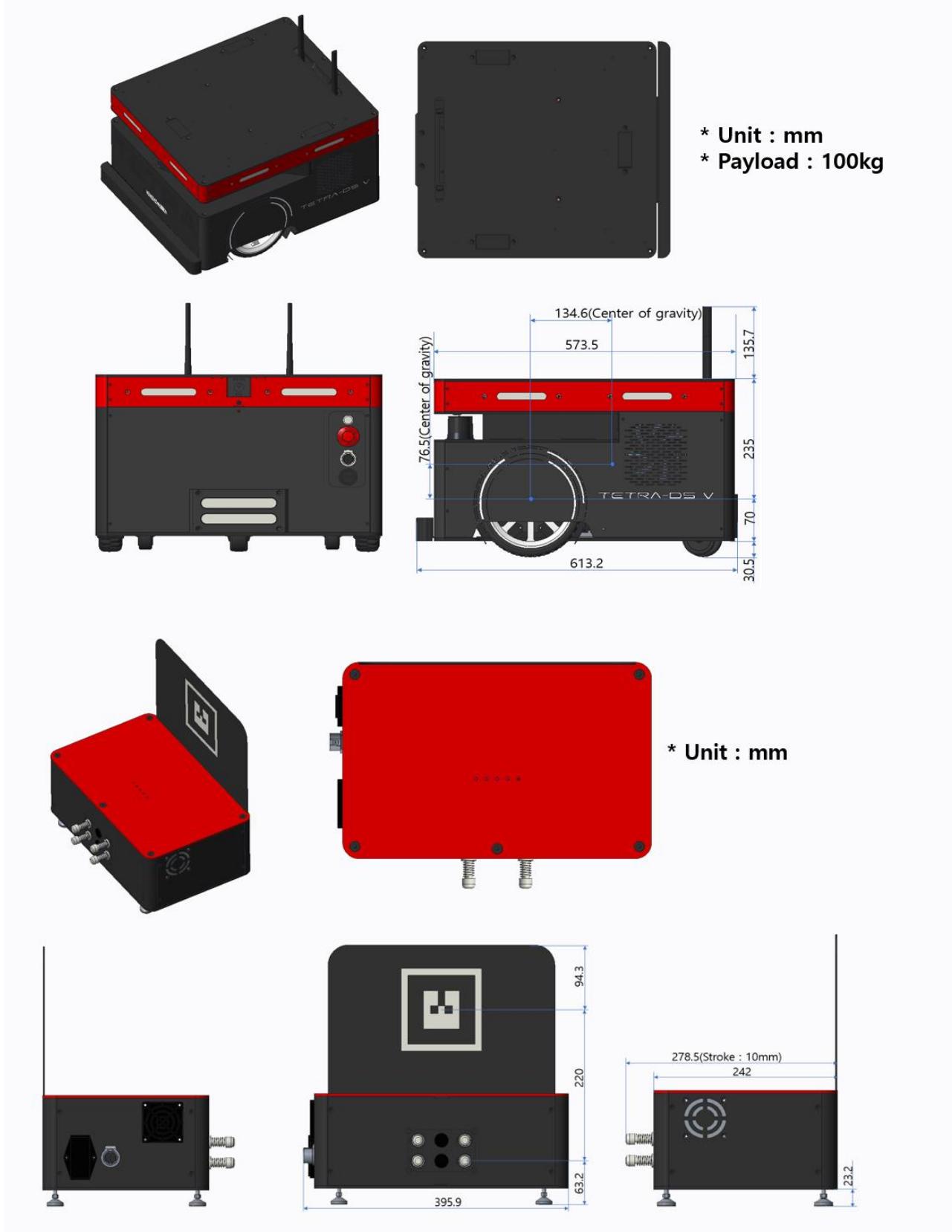
T. +82-70-4339-0935

Chapter 2. 기계적 사양(Mechanical Hardware Specifications)

TETRA-DSV를 구성하는 구성품들(Components)의 배치 정보는 그림 2-1에 나타난 바와 같습니다. TETRA-DSV의 구동방식은 차륜구동(Differential Drive) 방식으로 설계되어 있으며, TETRA-DSV의 구동모터로는 PMSM 모터 : 고성능 AC 서보 모터(Servo Motor)가 장착되어 있어 주행속도 및 가반하중(Payload)측면에서 성능이 우수합니다.



<그림 2-1> Components Layout of TETRA-DSV



<그림 2-2> Physical Dimensions of TETRA-DSV

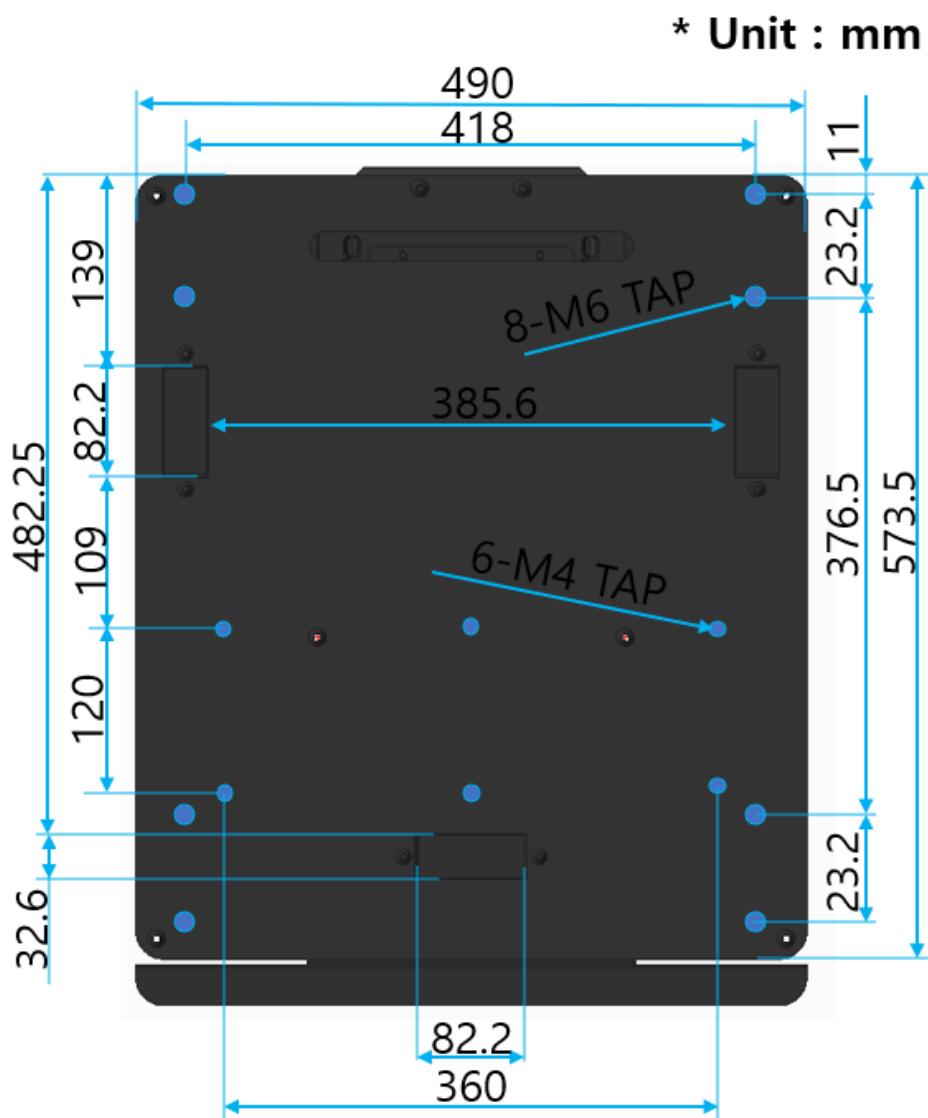
컴포넌트(Components)

- 마운트 판 (Mount Plate)

- 모터 및 엔코더 (Motors and Encoders)
- 캐스터 (Caster)

2-1. 마운트 판 (Mount Plate)

TETRA-DSV의 상판인 마운트 판(Mount Plate)에는 이동로봇의 자율주행 S/W 기술개발에 활용되는 다양한 센서 – 레이저 스캐너, 위치인식용 센서들이 장착될 수 있도록 다수의 취부용 탭(Mount Tap)이 제공되고 있습니다. TETRA-DSV은 자율주행 S/W 기술 개발에 주로 활용되는 몇 가지 센서들의 취부용 브라켓Bracket)을 추가품목으로 제공하여 판매하고 있습니다. 마운트 판에 뚫려있는 취부용 탭들의 치수는 아래 그림 2-3에 나타난 바와 같이 구성되어 있습니다.



<그림 2-3> Dimensions of Taps on Mount Plate

2-2. 모터 및 엔코더 (Motors and Encoders)

TETRA-DSV의 구동 시스템은 고속, 고토크의 특성을 갖는 AC Servo Motor를 채용하고 있으며, 각각의 구동모터에는 정밀한 속도 및 위치 검출을 통해 진보한 'Dead-Reckoning'이 가능한 고정밀 광학식 엔코더가 장착되어 있습니다. 구동 시스템을 구성하고 있는 구동모터, 감속기, 엔코더에 대한 사양은 표 2-1를 참고하시기 바랍니다.

ITEM	UNIT	SPECIFICATION
Nominal Power	W	100
Driving Voltage	Vdc	24
Nominal Speed	r/min	3000
Max. Speed	r/min	3000
Encoder	PPR	2500
Pulse per revolution	PPR	10000 (Quadratic)
Reduction Ratio	-	20 : 1

<표 2-1> Motor, Reducer & Encoder Specifications of TETRA-DSV

2-3. 캐스터 (Casters)

TETRA-DSV의 후방에는 자유로운 360도 회전이 가능한 캐스터가 장착되어 있어 직진, 선회, 등 TETRA-DSV의 안정적인 구동을 가능하게 합니다. 후방 캐스터는 메인 캐스터와 보조 캐스터, 두 종류로 구성되어 있으며, 양 옆의 보조 캐스터를 통해 로봇 전복을 방지할 수 있습니다. 후방 캐스터에 장착된 룰러는 플라스틱 소재로 제작되어 있어 장기간 사용할 경우 마모가 될 수 있으며, 이러한 경우에는 당사 고객지원센터를 통해 교체해 주시기 바랍니다.



<그림 2-4> Dimensions of Rear Caster

Chapter 3. 전기적사양 (Electrical Hardware Specifications)

컴포넌트(Components)

- LiDAR센서 (전방 : SICK TIM571 / 후방 : CYGBOT CygLiDAR D1)
- RGBD 카메라 (INTEL RealSense D455)
- 자동충전용 카메라
- 전방 범퍼
- 배터리 상태표시 LED
- 좌,우 상태표시 LED
- 주 전원 스위치
- 비상정지 버튼
- 수동충전 커넥터
- 자동충전 단자
- 전원/센서 보드 (Control BOX)
- 구동 보드 (Control BOX)
- 인터페이스 보드 (Control BOX)
- 충전스테이션

3-1. LiDAR센서

3-1-1 전방 LiDAR센서(SICK TIM571)

TETRA-DSV의 전방 주행센서는 SICK사의 TIM571 실외용 2D LiDAR를 장착하였으며 그림 3-1에 나타나 있는 바와 같이 플랫폼의 전면에 배치되었습니다. 주요 사양은 표 3-1에 나타난 바와 같습니다.



<그림 3-1>2D LiDAR센서

SECTION	CONTENTS
측정원리	HDDM
사용 분야	Indoor / Outdoor
광원	적외선(850nm)
레이저 등급	1(IEC 60825-1:2014, EN 60825-1:2014)
측정범위	TETRA-DS V 장착 256° (센서자체 270°)
스캔 주파수	15Hz
각 분해능	0.33°
작업 구역	0.05m ~ 25m

<표 3-1>SICK TIM571 사양

3-1-2 후방 LiDAR센서(CYGBOT CygLiDAR D1)

TETRA-DSV의 전방 주행센서는 CYGBOT사의 CygLiDAR D1 TOF LiDAR를 장착하였으며 2D와 3D 거리 데이터를 동시에 측정할 수 있습니다. 그림 3-2에 나타나 있는 바와 같이 플랫폼의 후면에 배치되었습니다. 주요 사양은 표 3-2에 나타난 바와 같습니다.



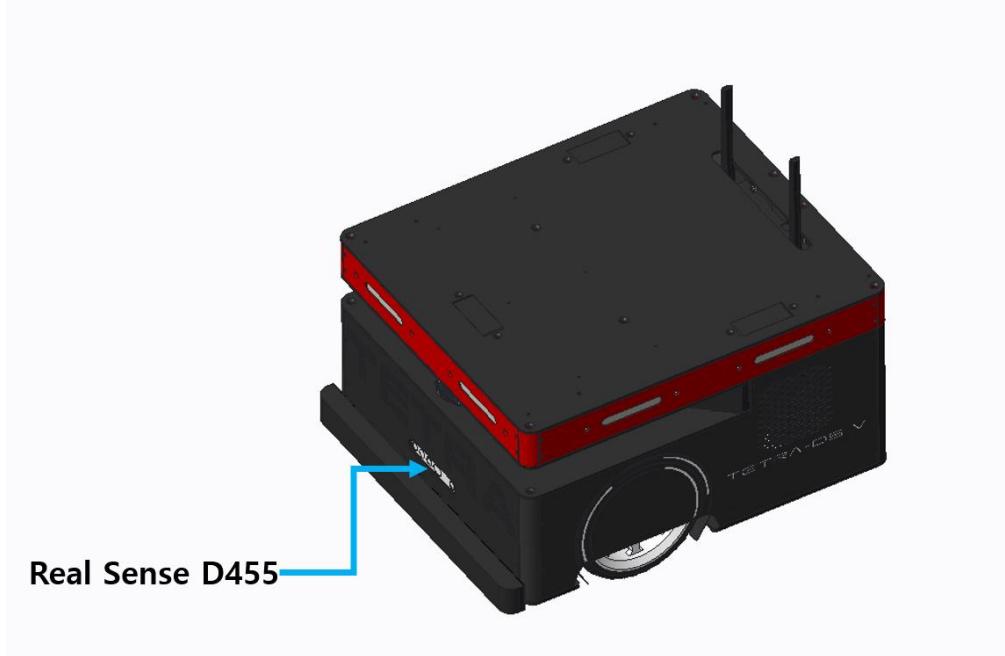
<그림 3-2> 2D/3D LiDAR센서

SECTION	CONTENTS
오차 범위	±1%
시야 각	2D/3D Horizontal : 120° 3D Vertical : 65°
파장	Laser Diode : NIR 808nm LED : NIR 808nm
통신 규격	UART TTL 3.3V 3,000,000 bps
측정 범위	2D : 200mm ~ 8,000mm 3D : 50mm ~ 2,000mm
정격 입력 전압	5V, 500mA
각 분해능	2D : 0.75° (Angle) 3D : 160 x 60 (Pixel)
작동 온도	-20°C ~ 60°C

<표 3-2> CYGBOT CygLiDAR D1 사양

3-2. RGBD카메라 (Intel RealSense D455)

TETRA-DS5의 주 주행보조 카메라는 Intel사의 Real Sense D435를 장착하였으며 그림 3-2에 나타나 있는 바와 같이 플랫폼의 전면 하단에 1개가 배치되었습니다. 주요 사양은 표 3-2에 나타난 바와 같습니다.



<그림 3-2>RGBD카메라

SECTION	CONTENTS
뎁스 기술	Active IR Stereo
작동 범위	0.4m ~ 6m
뎁스 해상도 및 FPS	1280 X 720 30fps 848 X 480 90fps
뎁스 시야 (H x V)	85.2° x 57° (+/- 3°)
치수	124mm X 26mm X 29mm
시스템 인터페이스 유형	USB 3.1 Gen 1

<표 3-2>Real sense D455 사양

3-3. 자동충전용 카메라

TETRA-DSV는 자동충전을 위한 USB카메라를 장착하였으며 그림 3-3에 나타나 있는 바와 같이 TETRA-DSV의 후면에 배치되었습니다. 주요 사양은 표 3-3에 나타난 바와 같습니다.



<그림 3-3> 자동충전용 USB카메라

SECTION	CONTENTS
제품 칩	OV9726
해상도	1920 X 720P 30fps
시야	74°
치수	60mm X 8mm X 4mm

<표 3-3>USB카메라 사양

3-4. 전방 범퍼

TETRA-DSV는 충돌 감지 및 충돌 시 충격을 감소시키기 위하여 그림 3-4에 나타나 있는 바와 같이 플랫폼의 전면 하단에 배치되었습니다. 범퍼 신호는 구동보드에서 감지하며 신호가 감지되면 상위제어기의 명령에 의해서도 전진방향으로는 구동이 되지 않습니다.

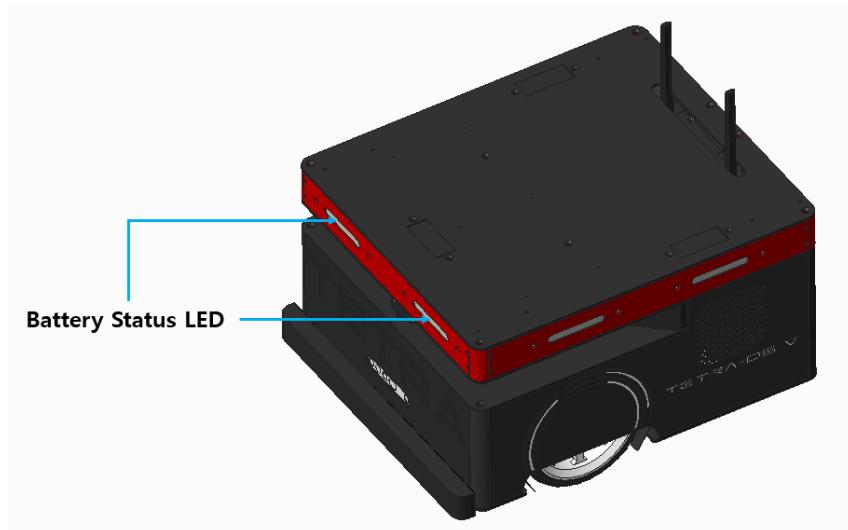


<그림 3-4>전방 범퍼

3-5. 배터리 상태 표시 LED (Battery Status LED)

그림3-5에 나타나 있는 바와 같이 TETRA-DSV의 전면에는 TETRA-DS5에 내장된 배터리의 잔량 및

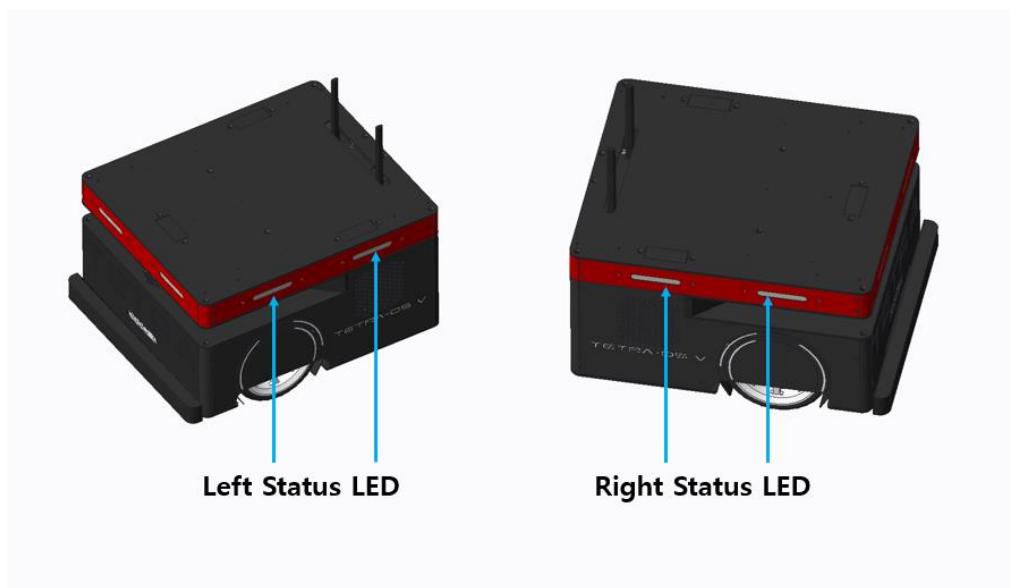
충전상태를 표시하는 LED창이 있습니다. 배터리 전압이 24.5V이상 에서는 GREEN이 Battery전압에 따라 점멸합니다. 24.5V이하에서는 RED가 Battery전압에 따라 점멸합니다. 자동충전단자를 통한 충전 중에는 RED가 충전 양에 따라 점멸합니다. 충전스테이션과 도킹 되어있는 상태에서 충전이 완료가 되면 ORANGE(GREEN,RED)가 점멸합니다.



<그림 3-5> Battery Status LED

3-6. 좌,우 상태 표시 LED (Left, Right Status LED)

TETRA-DSV의 좌측과 우측에 상태를 표현할 수 있는 LED창이 그림3-6에 나타나 있는 바와 같이 설치되어 있습니다. LED창에는 3Color(GREEN,RED,BLUE)LED를 프로토콜에 의해 각각 제어할 수 있습니다.



<그림3-6>Side Status LED

3-7. 주 전원 스위치 (Main Power Switch)

TETRA-DSV의 전원을 켜기 위한 주 전원 스위치(Main Power Switch)는 그림 3-7에 나타난 바와 같이 TETRA-DSV의 우 측면에서 접근 가능한 상부에 장착되어 있습니다. 주 전원 스위치는 파워/

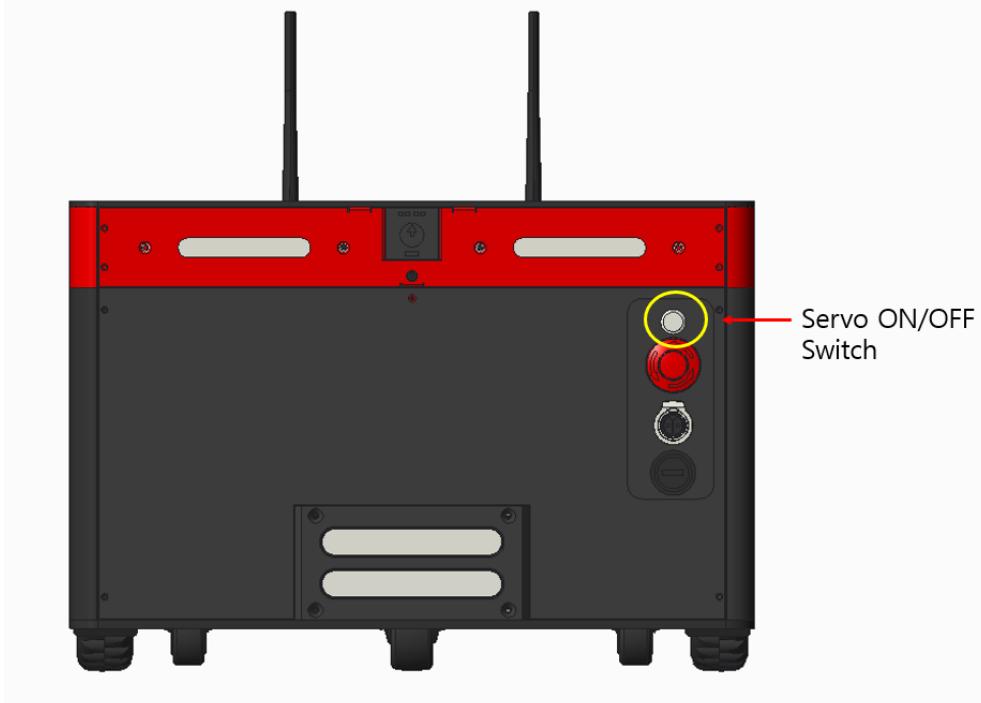
센서보드(후면커버 내부 Control BOX 장착)의 전원스위치와 Logical OR로 구성되어 있어 TETRA-DS5 주 전원스위치를 사용하기 위해서는 파워/센서보드 전원 스위치를 OFF로 설정해야 합니다.



<그림 3-7> Main Power Switch

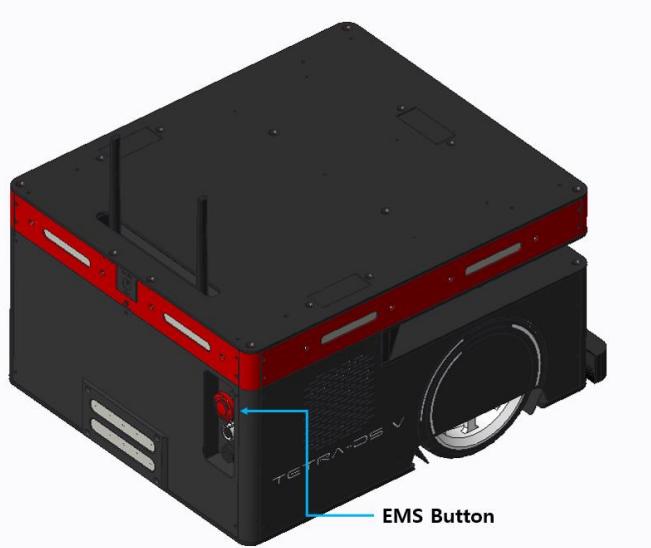
3-8. 서보 모터 전원 스위치(Servo ON/OFF Switch)

그림 3-8에 나타나 있는 바와 같이 TETRA-DSV의 후면에 장착되어 있는 '서보 모터 전원 스위치(Servo ON/OFF Switch)'는 기기를 수동 이동 시 사용하는 역할을 합니다. Servo OFF 작동 시, 제어 루프가 해제되어 프리-런(Free-Run) 상태로 전환되며, 이동할 수 있습니다. 추가적으로 Servo ON 작동 시 알람 발생 Clear 하는 기능도 포함되어 있습니다. 예를 들어 기동 중 Over current에 의한 알람이 발생할 경우, Servo 스위치를 통해 OFF 후 ON 할 경우 알람을 해제할 수 있습니다. 지속적인 알람 발생 시, 당사로 문의 바랍니다.



3-9. 비상정지 버튼 (Emergency Button)

그림 3-8에 나타나 있는 바와 같이 TETRA-DSV의 후면에 장착되어 있는 '비상정지 버튼(Emergency Button)' TETRA-DSV의 오작동이 발생한 상황이나 구동모터를 정지시킨 상태에서 행하여지는 각종 시험 시에 활용될 수 있습니다. 비상정지 버튼이 눌려진 경우, TETRA-DSV은 '강제정지상태'가 되며 구동모터들은 강제적으로 정지하게 되어 어떠한 제어명령으로도 구동모터를 구동할 수 없게 됩니다. 구동모터를 정상적으로 제어하기 위해서는 비상정지 버튼을 시계방향으로 회전시켜 강제정지 상태를 해제하셔야 합니다.

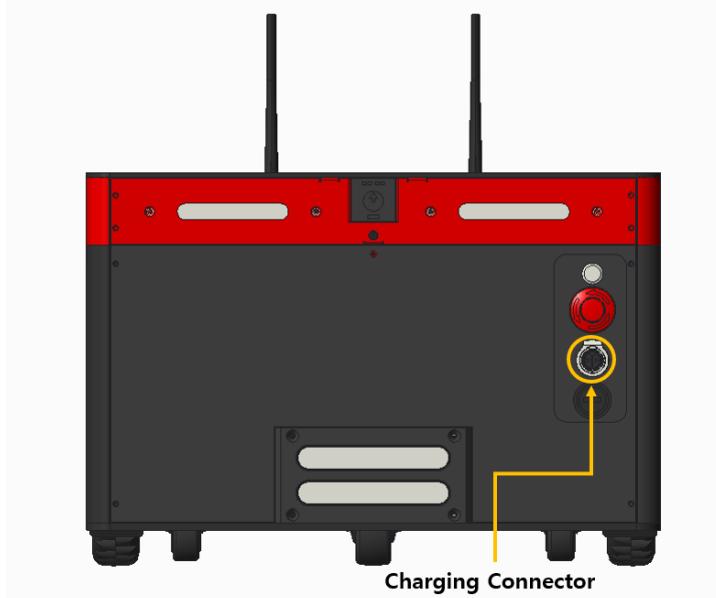


<그림 3-8> Emergency Button

3-9. 수동 충전 커넥터

TETRA-DS5의 충전을 위한 커넥터이며 TETRA-DSV의 배터리와 연결되어 있습니다.

배터리 전압이 20V이하의 경우(방전상태) 충전스테이션의 수동충전 단자에 연결하여야 충전이 가능합니다. TETRA-DS5의 수동 충전 단자의 위치는 그림3-9에 나타난 바와 같습니다.



<그림 3-9> Charging Connector

3-10. 자동 충전 단자

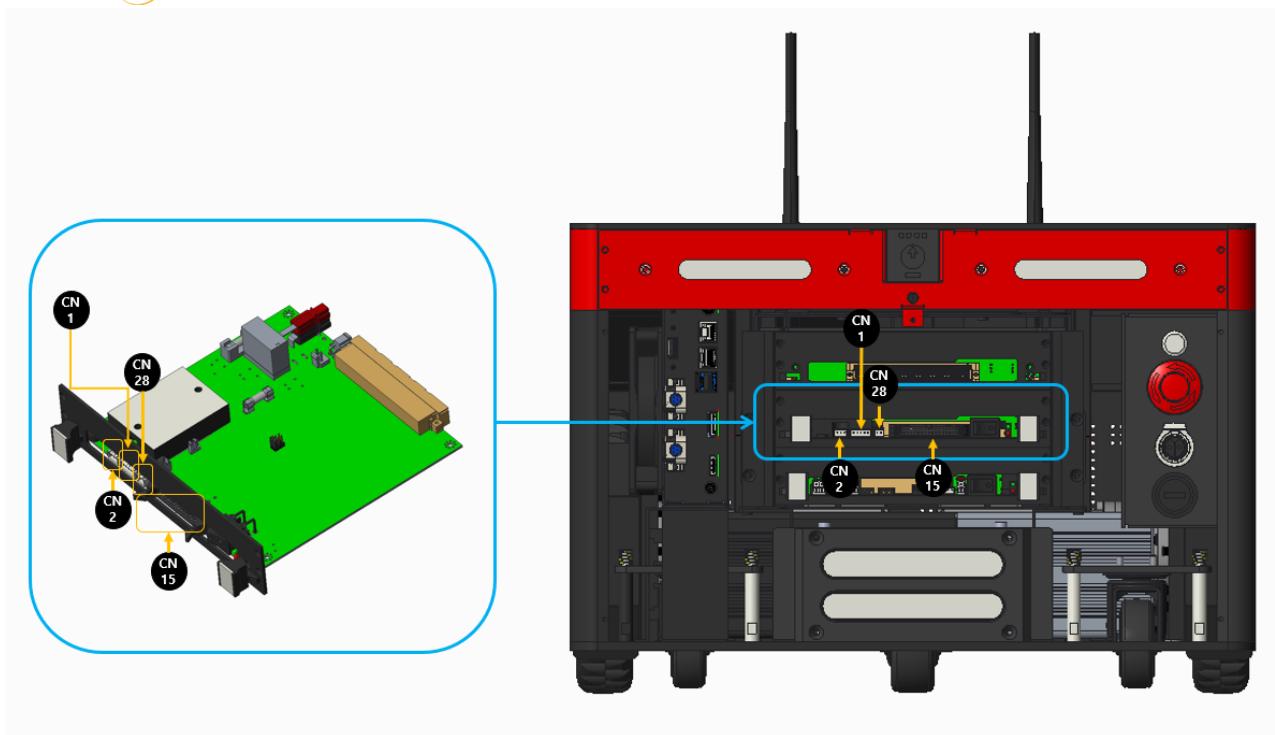
TETRA-DS5는 자동충전을 위한 단자가 플랫폼의 후면 하단에 그림3-10에 나타난 바와 같이 배치되어 있습니다. TETRA-DSV는 충전 +단자와 -단자를 통하여 충전스테이션의 도킹상태, 충전상태, 분리상태를 인식하며 정상적으로 도킹이 이루어 졌을 때 전원/센서보드 내부 충전 릴레이를 ON시킴으로써 충전이 이루어 집니다.



<그림 3-10>Auto Charging Terminal

3-11. 전원/센서 보드 (Power/Sensor Board)

전원/센서 보드는 TETRA-DSV에 장착된 각종 장치들에 필요한 전원을 분배하는 기능, 구동 보드의 전원을 ON/OFF하는 기능, 사용전력을 모니터링하는 기능, 등 전원을 관리하는 기능, 뿐만 아니라 초음파 센서의 정보를 수집, 범용 IN PORT, OUT PORT, ADC PORT 제어 및 모니터링 기능을 포함하고 있습니다. 전원/센서 보드의 CN2,CN1,CN28,CN15 의 사양은 표 3-4에 나타난 바와 같습니다. 그림 3-11은 전원/센서 보드를 나타내고 있습니다.

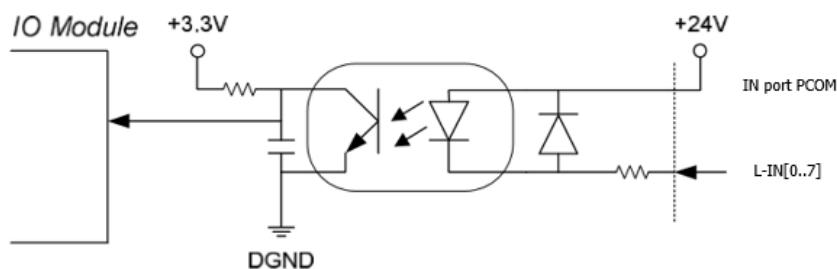


<그림3-11> Components Layout of Power Sensor Board

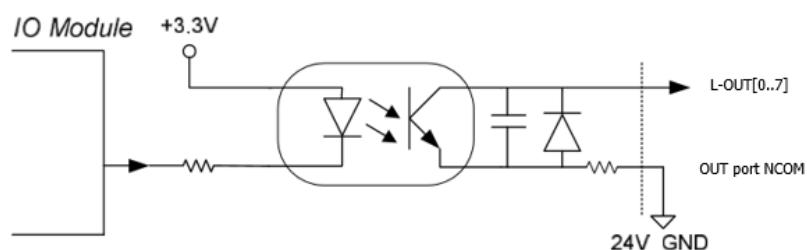
SECTION	NO	CONNECTOR	Pin No.	Pin DESCRIPTION	SPECIFICATION	REMARK
Power Sensor Board	CN2	SMAW250-03 (Yeonho)	1	RXD	UART(RS-232C) Comm. port 115200bps	Direct Control using User PC
			2	TXD		
			3	GND		
	CN1	SMAW250-05 (Yeonho)	1	P_MCLR		
			2	3.3V		
			3	GND		
			4	PGED2		
			5	PGEC2		
	CN28	SMAW250-02 (Yeonho)	1	NTC Thermister IN (10K)	온도센서 입력포트 1	
			2	GND		
	CN15	HIF3-30PA- 2.54DS (HIROSE)	1	IN PORT0	IN PORT 회로 그림 3-14 참조	
			3	IN PORT1		
			5	IN PORT2		
			7	IN PORT3		
			9	IN PORT4		
			11	IN PORT5		
			13	IN PORT6		
			15	IN PORT7		
			2	OUT PORT0	OUT PORT 회로 그림 3-15 참조	
			4	OUT PORT1		
			6	OUT PORT2		
			8	OUT PORT3		
			10	OUT PORT4		
			12	OUT PORT5		
			14	OUT PORT6		
			16	OUT PORT7		
			17	PCOM	POWER1 : 24V	J11 Select

		18	PCOM	POWER7 : 12V POWER5 : 5V	
		19	NCOM(GND)	GND	
		20	NCOM(GND)		
		21	ADC PORT0		
		22	ADC PORT1		
		23	ADC PORT2	Voltage Range : 0V~5V Resolution : 10bit	Caution! Do not supply over 5V
		24	ADC PORT3		
		25	ADC PORT4		
		26	ADC PORT5		
		27	ADC PORT6		
		28	ADC PORT7		
		29	AVCC	POWER6 : 5V	
		30	GND		

<표 3-4> Pin Map Information of Power/Sensor Board CN2, CN15



<그림3-12>Circuit of IN Port



<그림3-13>Circuit of OUT Port

3-12. 구동 모듈 (Wheel Drive Board)

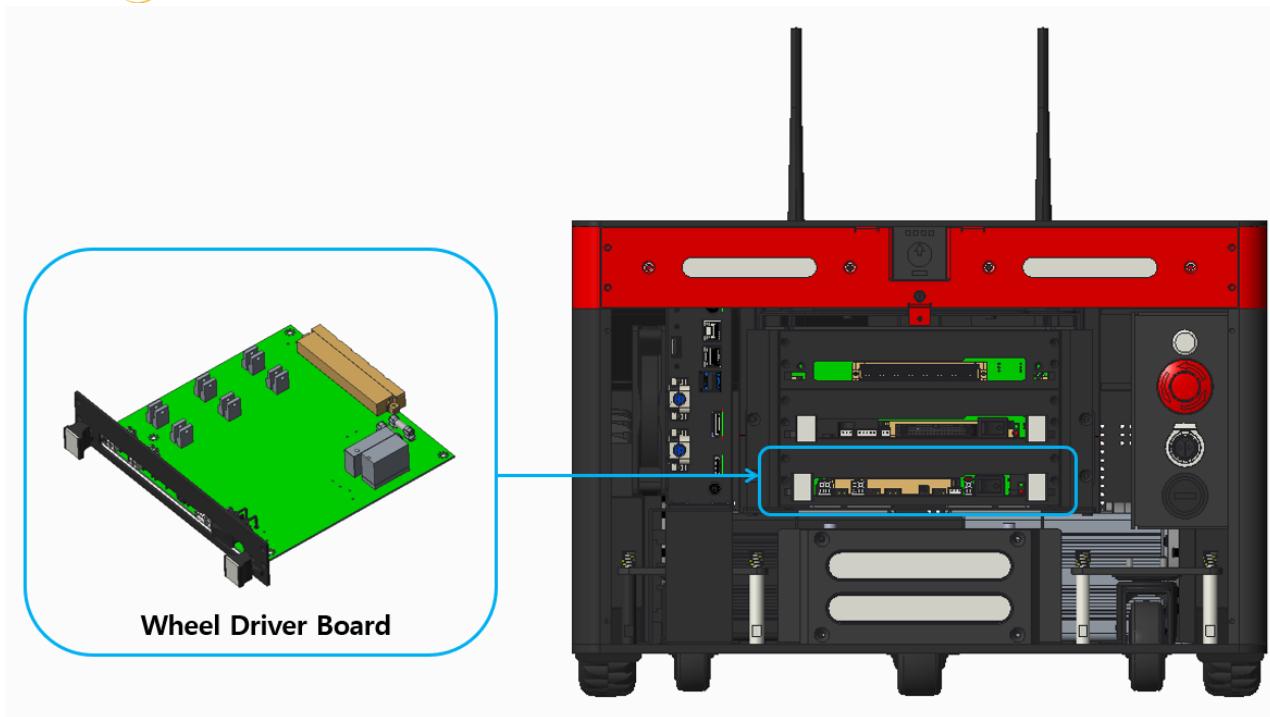
구동 보드는 TETRA-DSV에 장착된 고성능 AC Servo Motor를 제어하는 기능을 수행하는 보드입니다. 구동 보드에는 TETRA-DSV의 주행 시에 TETRA-DSV의 안정성 강화를 위해 범퍼 센서 및 비상 정지 버튼과의 연동 기능이 내장되어 있습니다. 구동 보드에 대한 사양은 표 3-5에 나타나 있으며, AC Servo Motor 사양은 표 3-6에 나타나 있습니다.

ITEMS	SPECIFICATION			
Input Voltage	DC 24V, +10% ~ -15%			
Motor	AC Servo Motor(PMSM)			
Max. Motor No.	2 Axis			
Control Type	PWM Voltage Control			
Feedback	Encoder (Voltage Input Type)			
Input Command	Communication with PC/Controller			
Communication	RS-232C (115200bps)			
Protection	Tracking error, Emergency Stop, Overvoltage, Undervoltage, Overload			
Alarm	Motor Power off if Alarms occur (Relay control)			
WxDxH	196mm × 190mm ×40mm			
Peripheral	Sub-ITEM	No.	Type	Range(V)
	No. Bumper Input	8	Buffer IN	Active LOW LOW : 0, HIGH : 3
	EMS	1	Photo-coupler IN	Active LOW LOW : 0, HIGH : 3

<표3-5> Specifications of Drive Board

ITEMS	UNIT	SPECIFICATION
Flange Size	mm	60
Rated Output	kW	0.1
Poles	-	8
Rated Speed	r/min	3000
Maximum Speed	r/min	3000
Rated Torque	N·m	0.32
	kgf·cm	3.24
Rated Current	A _(rms)	6.2
Phase Resistance	Ω	0.128
Phase Inductance	mH	0.3
Encoder	-	15 wire 2500PPR
Weight	kg	0.78
Driving Voltage	V dC	24

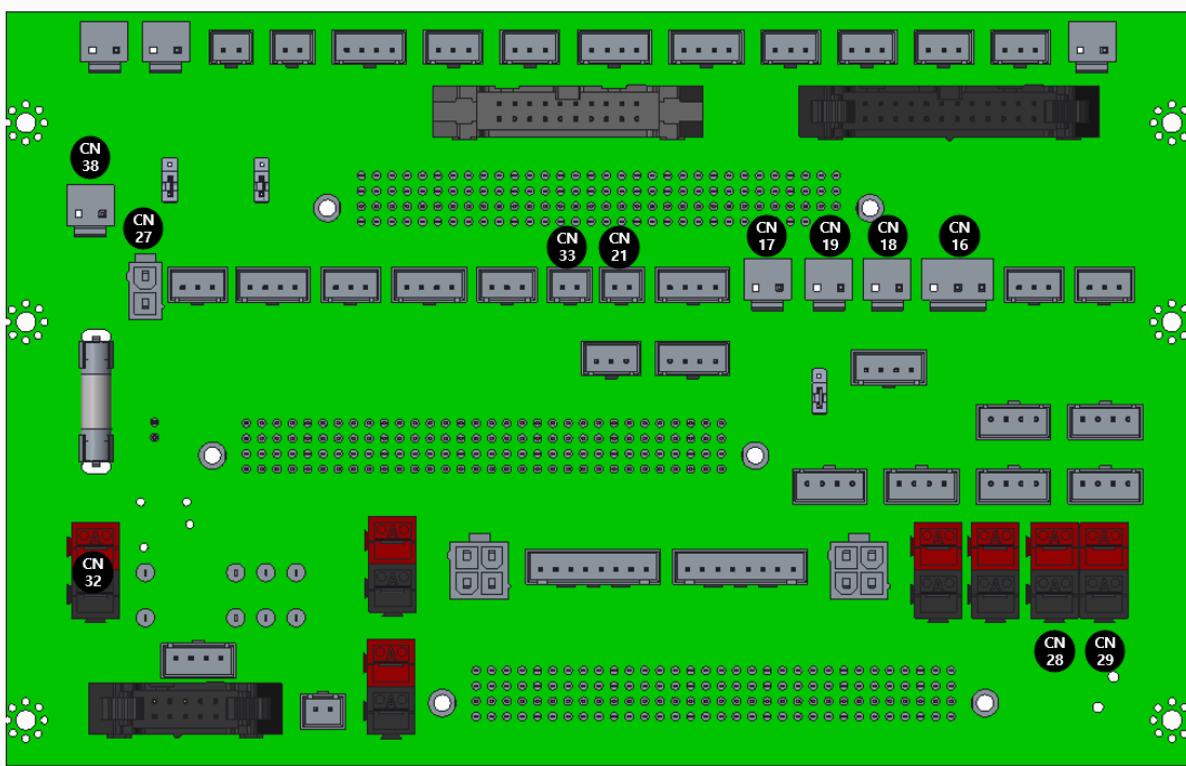
<표3-6> Specifications of AC Servo motor



<그림 3-14> Components Layout of Drive Board

3-13. 인터페이스보드전원 포트

그림 3-15에 나타나 있는 바와 같이 인터페이스보드에는 각종전원공급을 위한 다수의 커넥터가 제공되고 있으며 각 커넥터의 핀 사양은 표 3-7에 나타난 바와 같습니다.



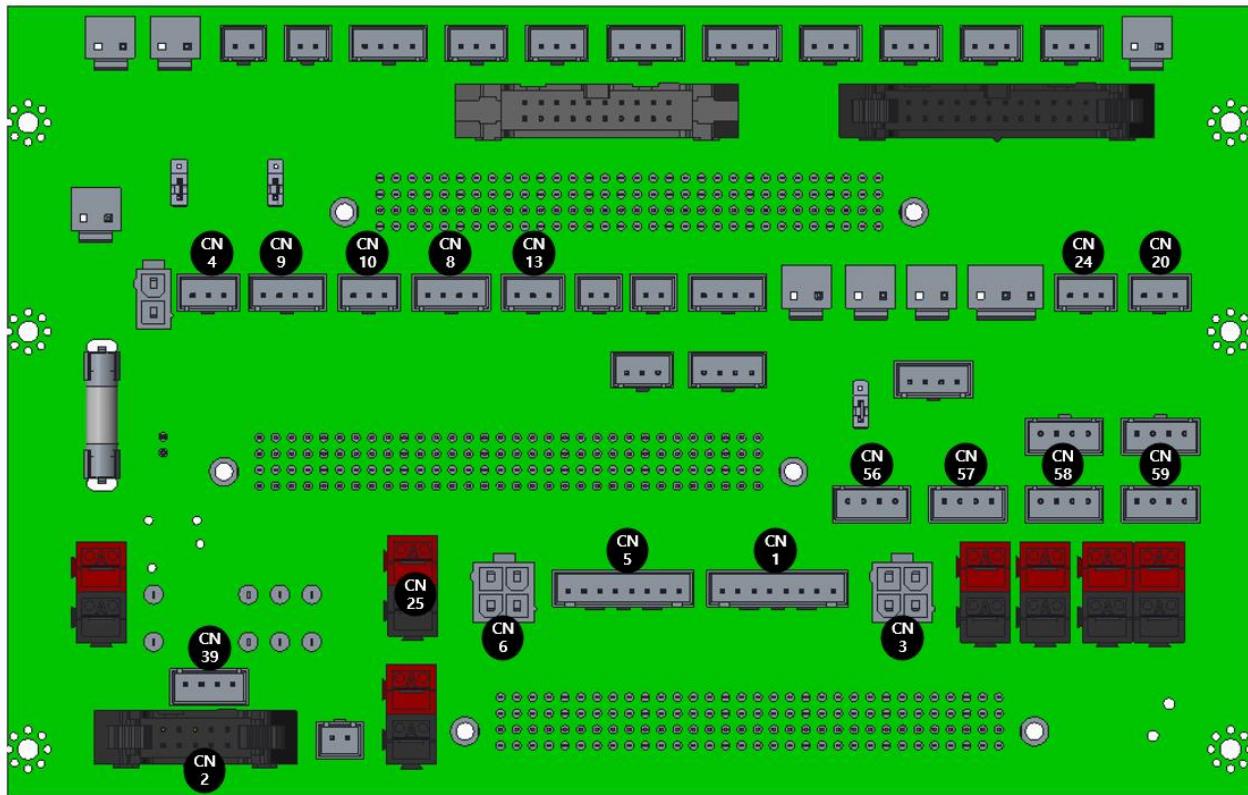
<그림 3-15> BackLinkPower Ports

NO	CONN-CTOR	Pin No.	Pin DESCRIPTION	MAX Current	Power ON/OFF	Control Port	Remark
CN16	YAW396-03 (Yeonho)	1	19V	4A	O	Power0	PC Power
		2	GND				
		3	NC				
CN17	YAW396-02 (Yeonho)	1	12V	4A	O	Power2	LiDAR Power
		2	GND				
CN18	YAW396-02 (Yeonho)	1	10V	4A	O	Power3	
		2	GND				
CN19	YAW396-02 (Yeonho)	1	5V	4A	O	Power4	
		2	GND				
CN27	5569-02A2 (Molex)	1	24V(Battery 25V 이상)	4A	O	Power1	
		2	GND				
CN21	SMAW250-02 (Yeonho)	1	12V/5V	1.5A	O	Power5	J3 1,2-> 5V
		2	GND			Power2	J3 2,3->12V
CN33	SMAW250-02 (Yeonho)	1	12V/5V	1.5A	O	Power4	J2 1,2->5V
		2	GND			Power2	J2 2,3->12V
CN32	PP15 Anderson	1	23V~28V	20A	X		SSR(POWER SW)-FUSE
		2	GND				
CN38	YAW396-02 (Yeonho)	1	23V~28V	7A	X		SSR(POWER SW)-FUSE
		2	GND				
CN28	PP15 Anderson	1	22~28V	20A	X		Battery
		2	GND				
CN29	PP15 Anderson	1	22~28V	20A	X		Battery
		2	GND				

<표 3-7> Pin Map Information of Power Ports

3-14. 인터페이스보드 신호 포트

그림 3-18에 나타나 있는 바와 같이 플랫폼내부의 인터페이스보드는 구동보드 그리고 파워/센서 보드에 접근하기 위한 다양한 포트들이 제공됩니다. 각 포트들의 핀 사양은 표 3-8에 나타난 바와 같습니다.



<그림 3-16> Interface Ports on Back Link

SECTION	NO	CONNECTOR	PIN NO.	PIN DESCRIPTION	SPECIFICATION	REMARK
Drive Board	CN20	SMAW250-03 (Yeonho)	1	RXD	UART(RS-232C) Comm. port 115200bps	Direct Control using User PC
			2	TXD		
			3	GND		
	CN4	SMAW250-03 (Yeonho)	1	EMS	EMS Signal (Active LOW)	
			2	GND		
			3	NC		
	CN39	SMAW250-04 (Yeonho)	1	NC	Servo ON/OFF	
			2	Servo ON/OFF		
			3	NC		
			4	GND		
	CN2	HIF3BA-10PA- 2.54DSA	1	Bumper0	Bumper Signal (Active LOW)	TETRA DSV 전방 범퍼는 0 번포트만 사용
			2	Bumper1, servo on/off		
			3	Bumper2		
			4	Bumper3		
			5	Bumper4		
			6	Bumper5		
			7	Bumper6		
			8	Bumper7		
			9	BUMPER_VCC		
			10	GND		
	CN3	SMW420-04 (Yeonho)	1	Left-U	LEFT MOTOR Phase Signal	
			2	Left-V		
			3	Left-W		
			4	FG		
	CN1	SMW250-08 (Yeonho)	1	5V	LEFT MOTOR Encoder Signal	
			2	GND		

			3	L-A PHASE			
			4	L-B PHASE			
			5	L-C PHASE			
			6	L-U PHASE			
			7	L-V PHASE			
			8	L-W PHASE			
	CN6	SMW420-04 (Yeonho)	1	Right-U	RIGHT MOTOR Phase Signal		
			2	Right-V			
			3	Right-W			
			4	FG			
	CN5	SMW250-08 (Yeonho)	1	5V	RIGHT MOTOR Encoder Signal		
			2	GND			
			3	R-A PHASE			
			4	R-B PHASE			
			5	R-C PHASE			
			6	R-U PHASE			
			7	R-V PHASE			
			8	R-W PHASE			
Power Sensor Board	CN24	SMAW250-03 (Yeonho)	1	RXD	UART(RS-232C) Comm. port 115200bps	Direct Control using User PC Distance Range : 5mm~500mm	
			2	TXD			
			3	GND			
	CN56	SMAW250-04 (Yeonho)	1	5V	SONAR1 : Rear Left Top		
			2	SONAR_OUT1			
			3	SONAR_IN1			
			4	GND			
	CN57	SMAW250-04 (Yeonho)	1	5V	SONAR2: Rear Right Top		
			2	SONAR_OUT2			
			3	SONAR_IN2			
			4	GND			
	CN58	SMAW250-04 (Yeonho)	1	5V	SONAR3: Rear Right Side Bottom		
			2	SONAR_OUT3			
			3	SONAR_IN3			
			4	GND			
	CN59	SMAW250-04 (Yeonho)	1	5V	SONAR4: Rear Left Side Bottom		
			2	SONAR_OUT4			
			3	SONAR_IN4			
			4	GND			
	CN10	SMAW250-03 (Yeonho)	1	12V	Normal : GREEN Low Battery : RED Charging : RED	Brightness Range: 0%~100%	
			2	BATTERY GREEN			
			3	BATTERY RED			
	CN9	SMAW250-04(Yeonho)	1	12V	GREEN:LED Port Bit3 RED:LED Port Bit4 BLUE:LED Port Bit5	Brightness Range: 0%~100%	
			2	RIGHT GREEN			
			3	RIGHT RED			
			4	RIGHT BLUE			
	CN8	SMAW250-04 (Yeonho)	1	12V	GREEN:LED Port Bit0 RED:LED Port Bit1 BLUE:LED Port Bit2	Brightness Range: 0%~100%	
			2	LEFT GREEN			
			3	LEFT RED			
			4	LEFT BLUE			
	CN13	SMAW250-03 (Yeonho)	1	BATTERY	PUSH LOCK -> POWER ON		
			2	SWITCH SIGNAL			
			3	LED			
	CN25	PP15 Anderson	1	CHARGING+	Automatic charging terminal		
			2	CHARGING-			

<표 3-8> Pin Map Information of Rear Case Ports and Modules

Chapter 4. 제어 하드웨어 프로토콜 (Control Hardware Protocol)

4-1. 통신규약(Rules of Communication)

대부분의 DATA는 ASCII Code로 통신합니다. (일부 DATA Packet은 Binary 통신) 한 개의 통신 Packet이란 다음과 같이 STX,DATA,ETX,LRC로 이루어진 구조를 말합니다. 한 개의 통신 Packet안에 여러 개의 DATA가 있을 경우 DATA간 ';' (0x3b)로 구분합니다.(일부 명령 제외)



월세 세액공제

ITEMS	CONTENTS
STX	0x02
ETX	0X03
LRC	STX, LRC 를 제외한 exclusive-OR $LRC = DATA[0] \oplus DATA[1] \oplus \dots \oplus DATA[N] \oplus ETX$

Receiver DATA는 binary(HEX)로 통신합니다. 한 개의 통신 Packet은 다음과 같이 STX,NUM,DATA,ETX,LRC로 이루어 집니다. NUM은 2byte로 이루어 지며 DATA의 byte수를 알려줍니다.
한 개의 DATA는 2byte로 이루어 집니다.



ITEMS	CONTENTS
STX	0x02
NUM	Data byte 수
ETX	0X03
LRC	STX, NUM, LRC 를 제외한 exclusive-OR $LRC = DATA[0_H] \oplus DATA[0_L] \oplus \dots \oplus DATA[N_H] \oplus DATA[N_L]$

4-2. 통신 Packet 운영 방법 (Packet Communication Methods of Operation)

각각의 제어 보드에서 요구한 기능에 대하여 각 Packet은 다음과 같은 FLAG값을 갖습니다.

FLAG – 0x30 : 기능실행 OK

요구한 프로토콜의 실행이 완료되었음을 의미하며 제어 보드로부터의 응답이 이상없음을 나타냅니다.

FLAG – 0x31 : 프로토콜 ERROR

세부프로토콜에서 벗어난 기능을 요구하였을 때 제어 보드로부터 받게 되는 값입니다. 예로서

- 없는 기능을 요구
- 정의한 값을 벗어난 경우
- 데이터 패킷의 길이가 틀린 경우

등이 있습니다.

FLAG – 0x32 : 기능실행 FAIL

요구한 기능을 실행하다가 실패한 경우 받게 되는 값입니다. 0x32를 받은 측은 ACK 신호를 보내 통신을 종료합니다. 만약, 실행실패원인을 알고 싶은 경우에는 지정 된 프로토콜을 이용하여 에러 정보를 받을 수 있습니다.

4-3. 전원/센서 모듈 프로토콜 (Power Sensor Board Protocol)

ITEMS	Type	COMMAND	CONTENTS
Power status	HEX	PD	Power Status
Battery & Charger Status	HEX	PV	Battery & Charger Status Read
Analog Port Data	HEX	PA	Analog Port Data Read
Regulator Power Enable	ASCII	PE	Regulator Power ON/OFF
Wheel Power Enable	ASCII	PM	Wheel Driver BD Power ON/OFF
Out Port Control	ASCII	PO	Out Port ON/OFF Command
IN,OUT Port Status	HEX	PI	IN Port, OUT Port Status Read
Sonar Sensor Loop Enable	ASCII	PS	Sonar Sensor Loop Enable Command
Sonar Sensor Data	HEX	PN	Sonar Sensor Data Read
Parameter Write	ASCII	PW	Power Sensor BD Parameter Write Command
Parameter Flash Write	ASCII	PF	Parameter Flash memory Write Command
Parameter Data Read	ASCII	PR	Parameter Data Read
LED Brightness Control	ASCII	LI	LED Module Brightness Control(8EA)
LED Brightness Toggle Data	ASCII	LD	LED Brightness Toggle Data Write
LED Brightness Toggle Enable	ASCII	LT	LED Brightness Toggle Enable Command
Cumulative Voltage Data Read	HEX	IV	Cumulative Voltage Data Read (2byte, 500EA)
Cumulative Current Data Read	HEX	IC	Cumulative Current Data Read (2byte, 500EA)

<표4-1> Protocol commands summary of Power Sensor Board

4-3-1 PowerSensor BD status Read

플랫폼의 배터리 전압, 소모전류, 각 PORT들의 전원 ON/OFF상태등의 정보를 알려줍니다.

Data 형식은 HEX이며 1개 Data는 2Byte입니다.

Transmitter	STX	P	D	ETX	LRC				
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	D2_H	D2_L
				D3_H	D3_L	D4_H	D4_L	D5_H	D5_L
				D6_H	D6_L	D7_H	D7_L	D8_H	D8_L
				D9_H	D9_L	D10_H	D10_L	D11_H	D11_L
				D12_H	D12_L	ETX	LRC		

DATA	CONTENTS							
D0	Battery Voltage (0.1V)							
D1	System Current (0.1A)							
D2	Charge Current (0.1A)							
D3	Charge Signal (0~1023)							
D4	IN Port Status							
	EX_IN7	EX_IN6	EX_IN5	EX_IN4	EX_IN3	EX_IN2	EX_IN1	EX_IN0
D5	OUT Port Status							
	EX_OUT7	EX_OUT6	EX_OUT5	EX_OUT4	EX_OUT3	EX_OUT2	EX_OUT1	EX_OUT0
D6	Power Status							
	Error7	Error6	Error5	Error4	Error3	Error2	Error1	Error0
	Power7	Power6	Power5	Power4	Power3	Power2	Power1	Power0
D7	Charge Status(1~12)							
D8	Temperature0 (0.1°C)							
D9	Temperature1 (0.1°C)							

D7	Charge Status
0	자동충전 Mode OFF
1	자동충전 Mode ON
2	Robot 충전단자, 충전스테이션 충전단자 접촉 상태
3	Battery 충전 중
5	Robot 충전단자, 충전스테이션 충전단자 접촉 후 Robot 충전 Relay ON 상태
6	Battery Re-Charge Voltage (Para23) 보다 Battery 전압이 높은 상태
7	Robot 충전단자 불안정 접촉상태
11	Docking Terminal Loading 접촉
12	Docking Terminal Unloading 접촉

예)

Transmitter	STX	P	D	ETX	LRC				
Receiver	STX	0x00	0x2A	0x01	0x0F	0x00	0x1F	0x00	0x00
				0x00	0x00	0x00	0x01	0x00	0xBB
				0x00	0x58	0x00	0x2E	0x00	0x13
				0x00	0x27	0x00	0xFF	0x00	0x00
				0x00	0x01	ETX	LRC		

DATA	CONTENTS							
D0	Battery Voltage : 271 -> 27.1V							
D1	System Current : 31 -> 3.1A							
D2	Charge Current : 0 -> 0A							
D3	Charge Signal: 0 -> 충전스테이션에 접촉되지 않음							
D4	IN Port Status -> 19							
	OFF	OFF	OFF	ON	OFF	OFF	ON	ON
D5	OUT Port Status -> 39							
	OFF	OFF	ON	OFF	OFF	ON	ON	ON
D6	Power Status -> 63							
	NC	NC	Normal	Normal	Normal	Normal	Normal	Normal
	-	-	ON	ON	ON	ON	ON	ON
D7	Charge Status: 1 -> 충전대기 상태							
D8	Temperature0 : 252 -> 25.2°C							
D9	Temperature1 : 287 -> 28.7°C							

4-3-2 TETRA Status Read

TETRA-DS V에 필요한 Power 및 센서 정보를 알려줍니다.

Data 형식은 HEX이며 1개 Data는 2Byte입니다.

Transmitter	STX	P	Q	ETX	LRC				
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L		
				D2_H	D2_L	D3_H	D3_L		
				D4_H	D4_L	D5_H	D5_L		
				D6_H	D6_L	D7_H	D7_L		
				D8_H	D8_L	D9_H	D9_L	ETX	LRC

DATA	CONTENTS							
D0	Battery Voltage%							
D1	Battery Voltage							
D2	ROBOT Current							
D3	Charger Terminal Status							
D4	INPUT DATA FLAG							
D5	OUTPUT DATA FLAG							

D6	NC
D7	NC
D8	NC
D9	NC

D3	Charge Status
0	자동충전 Mode OFF
1	자동충전 Mode ON
2	Robot 충전단자, 충전스테이션 충전단자 접촉 상태
3	Battery 충전 중
5	Robot 충전단자, 충전스테이션 충전단자 접촉 후 Robot 충전 Relay ON 상태
6	Battery Re-Charge Voltage (Para23) 보다 Battery 전압이 높은 상태
7	Robot 충전단자 불안정 접촉상태
11	Conveyor Loading 접촉
12	Conveyor Unloading 접촉

4-3-3 Analog Port Data Read

8개의 ADC(Analog to Digital Converter) Port 값을 읽어옵니다.

Data 형식은 HEX이며 1개 Data는 2Byte입니다.

Transmitter	STX	P	A	ETX	LRC				
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	D2_H	D2_L

				D3_H	D3_L	D4_H	D4_L	D5_H	D5_L
				D6_H	D6_L	D7_H	D7_L	ETX	LRC

DATA	VALUE	CONTENTS
D0	0~1023	ADC0 Data
D1	0~1023	ADC1 Data
D2	0~1023	ADC2 Data
D3	0~1023	ADC3 Data
D4	0~1023	ADC4 Data
D5	0~1023	ADC5 Data
D6	0~1023	ADC6 Data
D7	0~1023	ADC7 Data

4-3-4 Power Sensor BD Regulator Power ON/OFF

TETRA-DS5 Power Sensor BD에 내장된 Regulator(8EA) 전원을 ON/OFF 제어할 수 있습니다.

Data형식은 ASCII입니다.

10진수의 Data(ASCII)의 Binary값이 Power Enable BIT로 표현됩니다.

Transmitter	STX	P	E	C2	C1	C0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

ITEMS	CONTENTS
C0	1 의 자리
C1	10 의 자리
C2	100 의 자리

Data	Regulator Power							
	NC	NC	Power5	Power4	Power3	Power2	Power1	Power0
	-	-	WheelRly	5V(4A)	10V(4A)	12V(4A)	24V(4A)	19V(4A)

예1)

Transmitter	STX	P	E	6	3	ETX	LRC
Receiver	STX	0	ETX	LRC			

Data (BIN)	Regulator Power							
	NC	NC	Power5	Power4	Power3	Power2	Power1	Power0
	-	-	ON	ON	ON	ON	ON	ON

예2)

Transmitter	STX	P	E	6	ETX	LRC
Receiver	STX	0	ETX	LRC		

Data (BIN)	Regulator Power							
	NC	NC	Power5	Power4	Power3	Power2	Power1	Power0
	-	-	OFF	OFF	OFF	ON	ON	OFF

예3)

Transmitter	STX	P	E	5	4	ETX	LRC
Receiver	STX	0	ETX	LRC			

Data (BIN)	Regulator Power							
	NC	NC	Power5	Power4	Power3	Power2	Power1	Power0
	-	-	ON	ON	OFF	ON	ON	OFF

4-3-5 Single PowerSensor BD Regulator Power ON/OFF

해당 ID의 Out Port를 ON/OFF 합니다. (Single 제어 모드에서 사용하는 명령.)

PE 명령과 달리 Pe 명령은 소문자 'e'임을 주의하세요.

(ROS의 Service Name은 'Power_single_enable_cmd' 입니다.)

Transmitter	STX	P	e	ID	;	DATA	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

	RANGE	CONTENTS
ID	0~5	POWER ID
ID	0~1	0 : OFF, 1 : ON

4-3-6 Wheel Driver BD Power RelayON/OFF

TETRA-DSV Wheel Driver BD 전원 공급 Relay를 ON/OFF 제어할 수 있습니다.

Data형식은 ASCII입니다.

Transmitter	STX	P	M	DATA	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

DATA	CONTENTS
1	Wheel Driver BD Relay ON
0	Wheel Driver BD Relay OFF

4-3-7 Out PortON/OFF

TETRA-DSV PowerSensor BD에 Out Port를 ON/OFF 제어할 수 있습니다. Data형식은 ASCII이며 10진수의 Data(ASCII)의 HEX값이 Power Enable BIT로 표현됩니다.

(ROS의 Service Name은 'Power_output_cmd' 입니다.)

Transmitter	STX	P	O	C2	C1	C0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

ITEMS	CONTENTS
C0	1 의 자리
C1	10 의 자리
C2	100 의 자리

Data (BIN)	Out Port							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0

4-3-8 OUT Port Single Control ON/OFF

해당 ID의 Out Port를 ON/OFF 합니다. (Single 제어 모드에서 사용하는 명령.)

PO 명령과 달리 Po 명령은 소문자 'o'임을 주의하세요.

(ROS의 Service Name은 'Power_single_output_cmd' 입니다.)

Transmitter	STX	P	o	ID	;	DATA	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

	RANGE	CONTENTS
ID	0~7	Out Port ID
ID	0~1	0 : OFF, 1 : ON

4-3-9 IN Port & OUT Port status Read

IN Port, OUT Port 상태를 알려줍니다.

Data 형식은 HEX이며 1개 Data는 2Byte입니다.

(ROS의 Service Name은 'Power_io_status_cmd' 입니다.)

Transmitter	STX	P	I	ETX	LRC
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L

DATA	CONTENTS							
D0	IN Port Status							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	EX_IN7	EX_IN6	EX_IN5	EX_IN4	EX_IN3	EX_IN2	EX_IN1	EX_IN0
D1	OUT Port Status							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	EX_OUT7	EX_OUT6	EX_OUT5	EX_OUT4	EX_OUT3	EX_OUT2	EX_OUT1	EX_OUT0

4-3-10 Sonar Sensor Loop Enable

TETRA-DSV의 초음파 센서동작 Loop를 ON/OFF 합니다.

Data형식은 ASCII입니다.

(ROS의 Service Name은 'Power_sonar_start_cmd' 입니다.)

Transmitter	STX	P	S	DATA	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

DATA	CONTENTS
1	Sonar Loop Enable
0	Sonar Loop Disable

4-3-11 Sonar Sensor Data Read

Para3의 개수로 Sonar sensor Data를 읽어옵니다. 장애물이 없는 경우 Para4 값으로 표현됩니다.
Data 형식은 HEX이며 1개 Data는 2Byte입니다.
(ROS의 Service Name은 'Power_sonar_read_cmd' 입니다.)

Transmitter	STX	P	N	ETX	LRC										
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	D2_H	D2_L	D3_H	D3_L	D4_H	D4_L	ETX	LRC

*참고 – 당사의 TETRA-DS V에 적용된 초음파 센서는 (주) 하기소닉의 HG-B40C (5v) 제품입니다.

DATA	RANGE	CONTENTS
D0	0~Para4	NC (No Connect)
D1	0~Para4	Rear Left Top (mm)
D2	0~Para4	Rear Right Top (mm)
D3	0~Para4	Rear Side Right Bottom (mm)
D4	0~Para4	Rear Side Left Bottom (mm)

4-3-12 PowerSensor BD Parameter Data Write

PowerSensor BD의 Parameter ID에 Data를 Write 하며 Data 형식은 ASCII입니다.

(ROS의 Service Name은 'Power_parameter_write_cmd' 입니다.)

Transmitter	STX	P	W	ID	;	DATA	ETX	LRC				
Receiver	STX	FLAG	ETX	LRC								

ID	DEFAULT	RANGE	CONTENTS
0	4	0~10	UART1 Baudrate
1	6	0~7	UART2 Baudrate
2	1	0~1	초음파 선택 (0:HG-B40, 1:HC-SR04)
3	16	0~500	초음파 수신거리 OFFSET
4	5	1~5	초음파 센서 수량
5	500	40~3000	초음파 최대 감지거리
6	0	0~2000	NC (No Connect)
7	0	0~1	장애물이 없을 때 표시 모드 설정 (0 or 최대감지거리)
8	15	1~100	충전 중 배터리 잔량 OFFSET
9	2	1~100	충전 중 배터리 잔량 GAIN
10	100	0~500	Display LED 최소 밝기
11	200	0~2000	Display LED 최대 밝기
12	100	0~500	Battery 상태 LED 최소 밝기
13	200	0~2000	Battery 상태 LED 최대 밝기
14	0	0~200	Left Green LED OFFSET
15	35	0~200	Left Red LED OFFSET
16	25	0~200	Left Blue LED OFFSET

17	10	0~200	Right Green LED OFFSET
18	25	0~200	Right Red LED OFFSET
19	25	0~200	Right Blue LED OFFSET
20	15	0~200	Battery Green LED OFFSET
21	15	0~200	Battery Red LED OFFSET
22	255	0~255	Power BD 전원 투입 시 Power Enable
23	0	0~255	Power BD 전원 투입 시 OUT PORT
24	279	240~300	Battery Re-Charge Voltage
25	6	1~20	Battery Re-Charge Off Offset Voltage
26	230	200~250	Battery 최소 전압
27	285	260~300	Battery 최대 전압
28	10	1~10000	Battery 누적 전압, 전류 Sampling Time (100msec)
29	1	0~1	Conveyor mode select (0:disable, 1:enable)
30	497	400~600	Loading Express OK value
31	7	1~10	Terminal status classification range

통신속도 (P0)		통신속도 (P1)	
Value	Baudrate	Value	Baudrate
0	9600bps	0	9600bps
1	19200bps	1	19200bps
2	38400bps	2	38400bps
3	57600bps	3	57600bps
4	115200bps	4	115200bps
5	125000bps	5	125000bps
6	150000bps	6	150000bps
7	187500bps	-	-
8	230400bps	-	-
9	250000bps	-	-
10	500000bps	-	-

Power Enable(P7)								
bit	7	6	5	4	3	2	1	0
Name	NC	NC	Power5	Power4	Power3	Power2	Power1	Power0
Volt	-	-	5V	5V	10V	12V	24V	18.5V
Max I	-	-	1.5A	4A	4A	4A	4A	4A
Default	-	-	ON	ON	ON	ON	ON	ON

Out Port(P8)								
bit	7	6	5	4	3	2	1	0
Name	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
Default	NC	NC	ON	ON	ON	ON	ON	ON

4-3-13 Parameter Data Flash Memory Write

Parameter를 Flash memory에 Write합니다.

Transmitter	STX	P	F	ETX	LRC
Receiver	STX	FLAG	ETX	LRC	

4-3-14 Parameter Data Read

Power Sensor BD의 Parameter를 Read합니다.

(ROS의 Service Name은 'Power_parameter_read_cmd'입니다.)

Transmitter	STX	P	R	ETX	LRC						
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H D3_H D6_H D9_H D12_H D15_H D18_H D21_H D24_H	D0_L D3_L D6_L D9_L D12_L D15_L D18_L D21_L D24_L	D1_H D4_H D7_H D10_H D13_H D16_H D19_H D22_H D25_H	D1_L D4_L D7_L D10_L D13_L D16_L D19_L D22_L D25_L	D2_H D5_H D8_H D11_H D14_H D17_H D20_H D23_H D26_H	D2_L D5_L D8_L D11_L D14_L D17_L D20_L D23_L D27_L	ETX	LRC

4-3-15 LED Brightness Toggle Data Write

Display LED를 DATA1의 가속도로 DATA2밝기로 ON, DATA3의 가속도로 DATA4밝기로 OFF 합니다. DATA0으로 Toggle하지 않는 ID의 LED 밝기를 지정합니다.

Data 형식은 ASCII입니다.

Transmitter	STX	L	D	DATA0	;	DATA1	;	DATA2	;	DATA3	;	DATA4	ETX	LRC
Receiver	STX	FLAG	ETX	LRC										

DATA	DEFAULT	RANGE	CONTENTS
DATA0	50	0~100	Left, Right 중 Toggle 하지 않는 LED 밝기
DATA1	5	0~50	LED ON 가속도
DATA2	100	0~100	LED ON 밝기
DATA3	5	0~50	LED OFF 가속도
DATA4	0	0~100	LED OFF 밝기

4-3-16 Display Toggle LED Select

TETRA-DSV는 플랫폼 좌,우 각 3색 LED (GREEN,RED,BLUE)의 밝기를 제어할 수 있습니다.

선택 LED를 LD명령에서 설정한 값으로 Toggle 제어합니다.

Data형식은 ASCII입니다. 10진수의 Data(ASCII)의 HEX값이 LED BIT로 표현됩니다.

Transmitter	STX	L	T	C1	C0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

ITEMS	CONTENTS
C0	1의 자리
C1	10의 자리

Data (BIN)	LED Port							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	NC	NC	RIGHT BLUE	RIGHT RED	RIGHT GREEN	LEFT BLUE	LEFT RED	LEFT GREEN

예)

Transmitter	STX	L	T	32	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

Transmitter	STX	L	D	100	;	30	;	100	;	5	;	10	ETX	LRC
Receiver	STX	FLAG	ETX	LRC										

DATA	CONTENTS
DATA0	Left LED R,G,B 100% ON(WHITE)
DATA1	LED ON 가속도 30
DATA2	LED ON 밝기 100%
DATA3	LED OFF 가속도 5
DATA4	LED OFF 밝기 10%

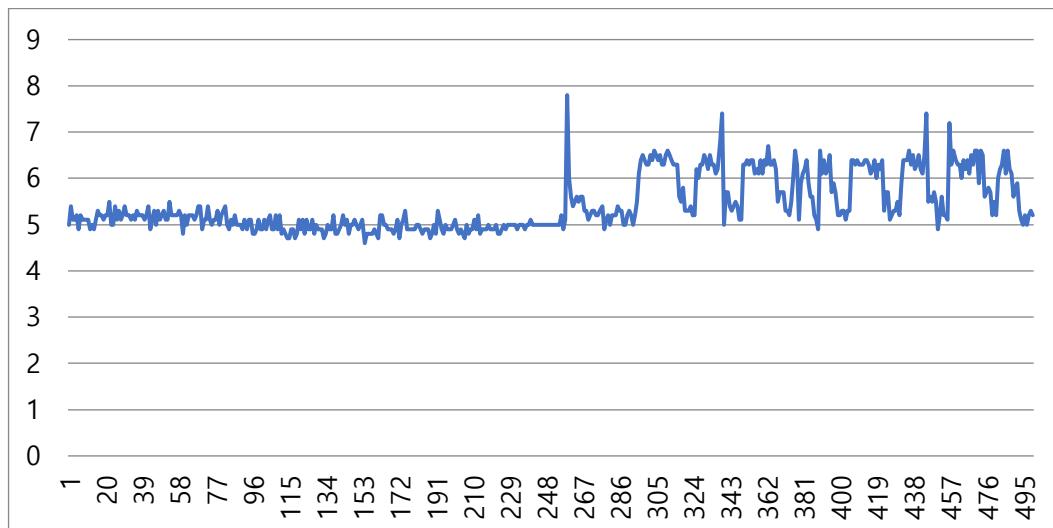
RIGHT BLUE LED 30의 가속도로 100% ON, 5의 가속도로 10%까지 OFF

LEFT LED 100% GREEN,RED,BLUE ON (WHITE)

4-3-17 Cumulative current Read

플랫폼은 설정되어있는 시간마다 현재의 소모전류를 저장합니다. Power ON 시 초기 Sampling time은 1sec이며, Para27에 의해 변경할 수 있습니다. 한 개의 DATA당 2Byte가 할당되어 있으며, 총 500개의 DATA를 저장합니다. DATA의 저장은 FIFO(First In First Out) 방식으로 저장됩니다. 그림 4-1과 같이 플랫폼의 소모전류 변화를 모니터링할 때에 활용하면 유용합니다.

Data 형식은 HEX이며 1개 Data는 2Byte입니다.



<그림4-1> Current consumption graph

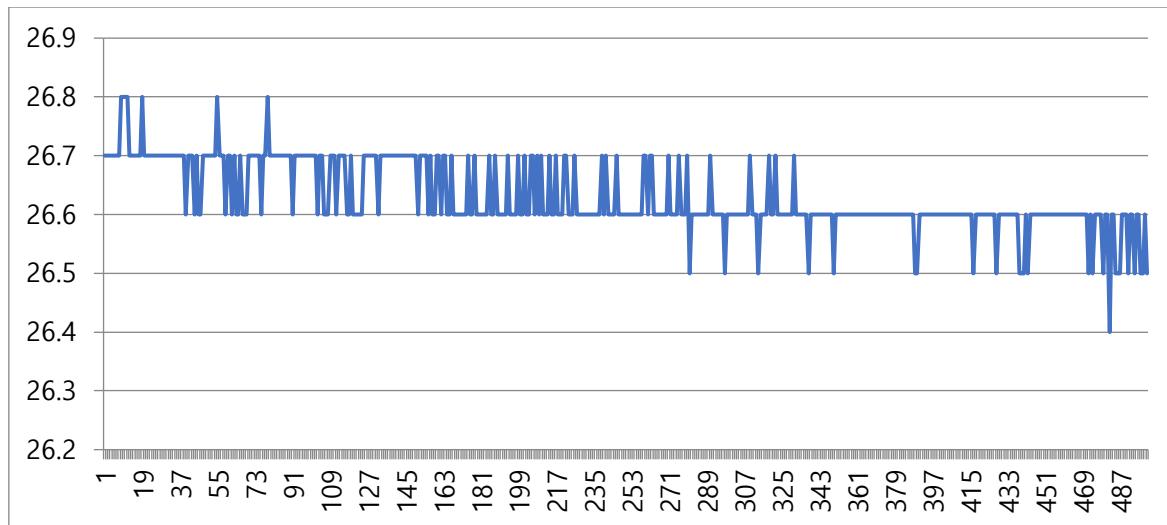
Transmitter	STX	I	C	ETX	LRC				
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	D2_H	D2_L
				D3_H	D3_L	D4_H	D4_L	D5_H	D5_L
				D6_H	D6_L	D7_H	D7_L	D8_H	D8_L
				D9_H	D9_L	D10_H	D10_L	D11_H	D11_L
			
				D488_H	D488_L	D489_H	D489_L	D490_H	D490_L
				D491_H	D491_L	D492_H	D492_L	D493_H	D493_L
				D494_H	D494_L	D495_H	D495_L	D496_H	D496_L
				D497_H	D497_L	D498_H	D498_L	D499_H	D499_L
								ETX	LRC

DATA	CONTENTS
N0	Data 갯수의 하위 바이트(HEX)
N1	Data 갯수의 상위 바이트(HEX)
D0~D499	누적 전류 DATA (HEX), 단위 : mA,

4-3-18 Cumulative voltage Read

플랫폼은 설정되어 있는 시간마다 현재의 전압을 저장합니다. Power ON 시 초기 Sampling time은 500msec이며, 해당 명령에 의해 변경할 수 있습니다. 한 개의 DATA당 1Byte가 할당되어 있으며, 총 1000개의 DATA를 저장합니다. DATA의 저장은 FIFO(First In First Out) 방식으로 저장됩니다. 그림 4-2와 같이 내장된 배터리의 전압변화를 모니터링할 때에 활용하면 유용합니다.

Data 형식은 HEX이며 1개 Data는 2Byte입니다.



<그림4-2> Voltage consumption graph

Transmitter	STX	I	V	ETX	LRC				
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	D2_H	D2_L
				D3_H	D3_L	D4_H	D4_L	D5_H	D5_L
				D6_H	D6_L	D7_H	D7_L	D8_H	D8_L
				D9_H	D9_L	D10_H	D10_L	D11_H	D11_L
			
				D488_H	D488_L	D489_H	D489_L	D490_H	D490_L
				D491_H	D491_L	D492_H	D492_L	D493_H	D493_L
				D494_H	D494_L	D495_H	D495_L	D496_H	D496_L
				D497_H	D497_L	D498_H	D498_L	D499_H	D499_L
								ETX	LRC

DATA	CONTENTS
N0	Data 갯수의 하위 바이트(HEX)
N1	Data 갯수의 상위 바이트(HEX)
D0~D499	누적 전류 DATA (HEX), 단위 : mA,

4-3-19 Cumulative buffer erase

Power Sensor Board의 누적 전압과 전류 Data Buffer를 초기화합니다.

Transmitter	STX	I	R	ETX	LRC
Receiver	STX	FLAG	ETX	LRC	

4-3-20 Cumulative buffer erase

IN Port, OUT Port 상태를 알려줍니다.

Data 형식은 HEX이며 1개는 Data는 2Byte입니다. (Single 제어 모드에서 사용하는 명령.)

PI 명령과 달리 Pi명령은 소문자 'i'임을 주의하세요.

Transmitter	STX	P	i	ETX	LRC
Receiver	STX	FLAG	ETX	LRC	

DATA	UNIT	CONTENTS
DATA0	BIT	IMPORT DATA FLAG
DATA1	BIT	OUTPORT DATA FLAG

4-4. 구동 보드 프로토콜 (Drive Board Protocol)

ITEMS	DATA Type	COMMAND	CONTENTS
Drive Status Read	ASCII	AA	Error Exist, Motor Power On/Off, In Position 모션 Patten Run State, Servo On/Off
Encoder Position Read	ASCII	AC0	Read current encoder data (Encoder)
Coordinates Read	ASCII	AC1	Read coordinates data (X, Y, θ)
Coordinates Change	ASCII	CX	Change coordinates data
Velocity Control(speed)	ASCII	BE	Command velocity for velocity control loop
Velocity Control(position)	ASCII	BH	Command velocity for position control loop
Error Reset	ASCII	CG	Clear error
Servo On/Off	ASCII	DB	Servo ON/OFF (Motor On/Off)
Control Mode change	ASCII	CZ	Select control loop (Velocity/Position)

<표4-2> Protocol commands summary of Drive Board

4-4-1구동 보드 파라미터 일람표 (Drive Board Parameter Chart)

No	초기값	단위	범위	내용
0	0		0~20	모터 Type (0: 24V, 100Watt)
1	120	%	0~500	최대 Overload
2	10000	Pulse	1~20000	Encoder pulse (4 체배)
3	20		1~200	감속비 (15 : 15/1, 20 : 20/1)
4	203	mm	1~1000	바퀴직경
5	438	mm	1~1000	바퀴간 거리
6	50	10msc	8~1000	가감속 시간(RUN Speed 까지 도달하는 시간)
7	3	%	0~100	EMG 감속도(작을수록 감속 기울기가 큼)
8	1500	mm/sec	0~1000	최대속도
9	0		0~1	왼쪽바퀴 엔코더 회전방향 0 : cw, 1:ccw
10	0		0~1	오른쪽바퀴 엔코더 회전방향 0:cw, 1:ccw
11	0		0~1	왼쪽바퀴 모터 회전방향 0:cw, 1:ccw
12	1		0~1	오른쪽바퀴 모터 회전방향 0:cw, 1:ccw
13	3000		-10000~10000	왼쪽바퀴 Temporary Valuable(I BANDWIDTH)
14	3000		-10000~10000	오른쪽바퀴 Temporary Valuable(I BANDWIDTH)
15	50	mm/sec	-500~500	왼쪽바퀴 RUN 속도
16	50	mm/sec	-500~500	오른쪽바퀴 RUN 속도
17	20		0~1000	왼쪽바퀴 위치 비례계인
18	20		0~1000	오른쪽바퀴 위치 비례계인
19	100		0~10000	왼쪽바퀴 위치 적분계인
20	100		0~10000	오른쪽바퀴 위치 적분계인
21	300	mm	0~10000	왼쪽바퀴 follow error range
22	300	mm	0~10000	오른쪽바퀴 follow error range
23	5	mm	0~10000	왼쪽바퀴 inpos range
24	5	mm	0~10000	오른쪽바퀴 inpos range
25	300		0~1000	왼쪽바퀴 속도 비례계인
26	300		0~1000	오른쪽바퀴 속도 비례계인
27	100		0~10000	왼쪽바퀴 속도 적분계인
28	100		0~10000	오른쪽바퀴 속도 적분계인
29	1		1~10	Baudrate 설정 (1 : 115200bps)
30	2000	ms	35~20000	상위제어기와 통신 두절 시 좌우모터 정지시간

<표4-3> 파라미터 일람표

4-4-2 Drive status Read

구동 보드의 상태정보를 알려줍니다. Error Code는 표4-4을 참고하시기 바랍니다.

Transmitter	STX	A	A	ETX	LRC							
Receiver	STX	FLAG	DATA0	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6	DATA7	ETX	LRC

DATA	CONTENTS							
	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
DATA0	0	0	1	Left Servo ON/OFF	Left Error	Left In Position	Left RUN	Left Power ON/OFF
DATA1	Left motor Error Code							
DATA2	0	0	1	Right Servo ON/OFF	Right Error	Right In Position	Right RUN	Left Power ON/OFF
DATA3	Right motor Error Code							
DATA4	0	0	0	0	Bumper7	Bumper6	Bumper5	Bumper4
DATA5	0	0	0	0	Bumper4	Bumper3	Bumper2	Bumper1
DATA6	0	0	0	0	0	0	0	EMG
DATA7	0	0	0	0	0	0	DIR1	DIR0

DIR1	DIR0	Direction of movement
0	0	Rotate Left, Stop
0	1	Advance
1	0	Reverse
1	1	Rotate Right

CODE	CONTENTS	CAUSE	MEASURES
0x30 ('0')	Normal state	-	-
0x31 ('1')	Emergency Stop	1) 비상정지 스위치를 눌렀을 때	비상정지 스위치 해제
0x32 ('2')	Motor hall sensor error	1) Hall 소자 수신부 이상 2) 모터라인 단선	모터점검/모터라인 점검
0x33 ('3')	Encoder error	1) 엔코더 라인 이상 2) 모터라인 단선 및 오배선 3) 구동장치 Gain 설정 오류 4) Power Module 파손 5) 엔코더 수신부 이상	엔코더 배선 점검 모터 라인 계통 이상 Gain Tuning
0x34 ('4')	Detect Over Voltage	1) 전원전압 정격 초과 2) Power Module 파손 3) 가감속 설정 불량	전원 전압 점검 구동장치 B/D 점검 Parameter 설정 변경
0x35 ('5')	Detect Under Voltage	1) 전원전압 정격 이하	전원 전압 점검
0x36 ('6')	Detect Over Load	1) 모터의 정격 토크가 구동장치의 최대 출력 초과 2) 토크 Limit 설정 이상 3) 동작속도/가감속 설정 불량 4) 동작영역에 장애물	모터의 정격토크 검사 토크 Limit 설정 변경 속도/가감속 설정 변경 장애물 제거
0x37 ('7')	Detect Over Speed	1) 속도지령이 정격속도를 초과	구동장치의 Max 속도 설정 변경
0x38 ('8')	Detect Following Error	1) Following Parameter 설정 오류 2) 구동장치 Gain 설정 오류 3) 엔코더/모터 라인 이상	Parameter 설정 변경 엔코더/모터 라인 점검

<표4-4> Error Code

4-4-3 Encoder Position Read

구동 보드의 Left motor Encoder, Right motor Encoder 위치정보를 알려줍니다. Encoder Data는 좌, 우 모터 각각 10bytes로 할당되어 있습니다.

Transmitter	STX	A	C	0	ETX	LRC					
Receiver	STX	FLAG	Left Encoder position (10 Bytes)				Right Encoder position (10 Bytes)				ETX LRC

4-4-4 Coordinates Read

좌, 우 Encoder정보를 바탕으로 플랫폼의 현재 좌표정보를 알려줍니다. 전원 투입 시 초기 좌표는 X = 0(mm), Y = 0(mm), θ = 900(0.1°) 입니다. X, Y, θ 각각 10bytes로 할당되어 있습니다.

Transmitter	STX	A	C	1	ETX	LRC					
Receiver	STX	FLAG	X (10 Bytes)			Y (10 Bytes)			θ (10 Bytes)		ETX LRC

예) Coordinates X = 56.6mm, Y = 198.4, θ = 58.9°

Transmitter	STX	A	C	1	ETX	LRC					
Receiver	STX	0x30	0x20	0x20	0x20	0x20	0x20	0x20	0x35	0x36	0x2e 0x36
	0								5	6	.
		0x20	0x20	0x20	0x20	0x20	0x31	0x39	0x38	0x2e	0x34
							1	9	8	.	4
		0x20	0x20	0x20	0x20	0x20	0x20	0x35	0x38	0x39	0x2e
							5	8	9	.	
											ETX LRC

4-4-5 Coordinates Change

플랫폼의 현재 좌표(X, Y, θ)를 변경합니다. 좌표 변경 명령은 Servo OFF에서만 변경이 가능합니다.

Transmitter	STX	C	X	X (100um)	;	Y (100um)	;	θ (0.1°)	ETX	LRC	
Receiver	STX	FLAG	ETX	LRC							

4-4-6Motion

로봇이 움직일 위치를 구동장치에 제공하여 모션을 수행하는 명령입니다. 현재 위치에서 지정한 데이터만큼을 더하여 로봇을 움직입니다. 모션동작이 완료되기 이전에도 연속해서 다음 모션지령을 보낼 수 있습니다. (그러나 연속된 모션지령을 보낼 때 하나의 프레임이 끝나기 전에는 실행할 수 없습니다.)

Transmitter	STX	B	D	Type	PASS 모션	Data	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

구동장치의 모션에 관계됩니다. 구동장치는 받은 데이터가 타당한지 검사한 후 FLAG에 결과 값을 실어서 응답합니다. 통신 종료 후 구동장치는 모션을 수행하며 모션 중 최소한의 기능은 동작하여야 합니다.

모션동작의 예)

1) 현재 위치에서 1[m] 직진

Transmitter	STX	B	D	0	0	10000	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

2) 현재 위치에서 90도 우회전

Transmitter	STX	B	D	2	0	9000	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

현재 위치에서 1[m] 직진 후 회전 동작을 수행합니다. 회전 동작을 수행할 시점은 파라미터에서 결정합니다. 예를 들어 파라미터가 10이라면 직진 동작에서 9000[mm]에 도달 후 반경이 300[mm]이고 회전 각이 90도인 회전 모션을 수행합니다.

4-4-7 Velocity Control (Speed Mode)

구동 보드 내부 속도 LOOP의 속도지령 명령입니다. 속도 LOOP의 속도지령 명령은 상위제어기와의 통신 두절 시, 오동작 방지를 위해 설정된 시간 내에 명령을 받지 못하면 좌, 우 모터 속도를 0 (mm/sec)으로 보냅니다. 초기 설정 값은 2000msec입니다.

Transmitter	STX	B	E	Left Wheel Speed (mm/sec)	:	Right Wheel Speed (mm/sec)	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

4-4-8 Velocity Control (Position Mode)

구동 보드 내부 위치 LOOP의 속도지령 명령입니다.

Transmitter	STX	B	H	Left Wheel Speed (mm/sec)	:	Right Wheel Speed (mm/sec)	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

4-4-9 Error Reset

구동 보드에 발생한 모든 Error를 초기화 합니다.

Transmitter	STX	C	G	ETX	LRC
Receiver	STX	FLAG	ETX	LRC	

4-4-10 Servo ON/OFF

구동 보드의 좌, 우 Servo ON/OFF 명령입니다.

Transmitter	STX	D	B	Left Servo (1 : ON, 0 : OFF)	Right Servo (1 : ON, 0 : OFF)	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

4-4-11 Control Mode Change

구동 보드의 위치제어 모드와 속도제어 모드 변환 명령입니다.

속도제어모드

Transmitter	STX	C	Z	1	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

위치제어모드

Transmitter	STX	C	Z	0	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

4-4-12 Parameter Read

해당 ID의 파라미터 값을 Read 합니다.

Transmitter	STX	X	L	Parameter ID	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

4-4-13 Parameter Write

해당 ID의 파라미터 값을 Write 합니다.

Transmitter	STX	X	B	Parameter ID	Data	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

4-4-14 Parameter Save

현재 버퍼의 저장되어 있는 모든 파라미터 값을 Flash Memory에 Save 합니다.

Transmitter	STX	X	D	ETX	LRC
Receiver	STX	FLAG	ETX	LRC	

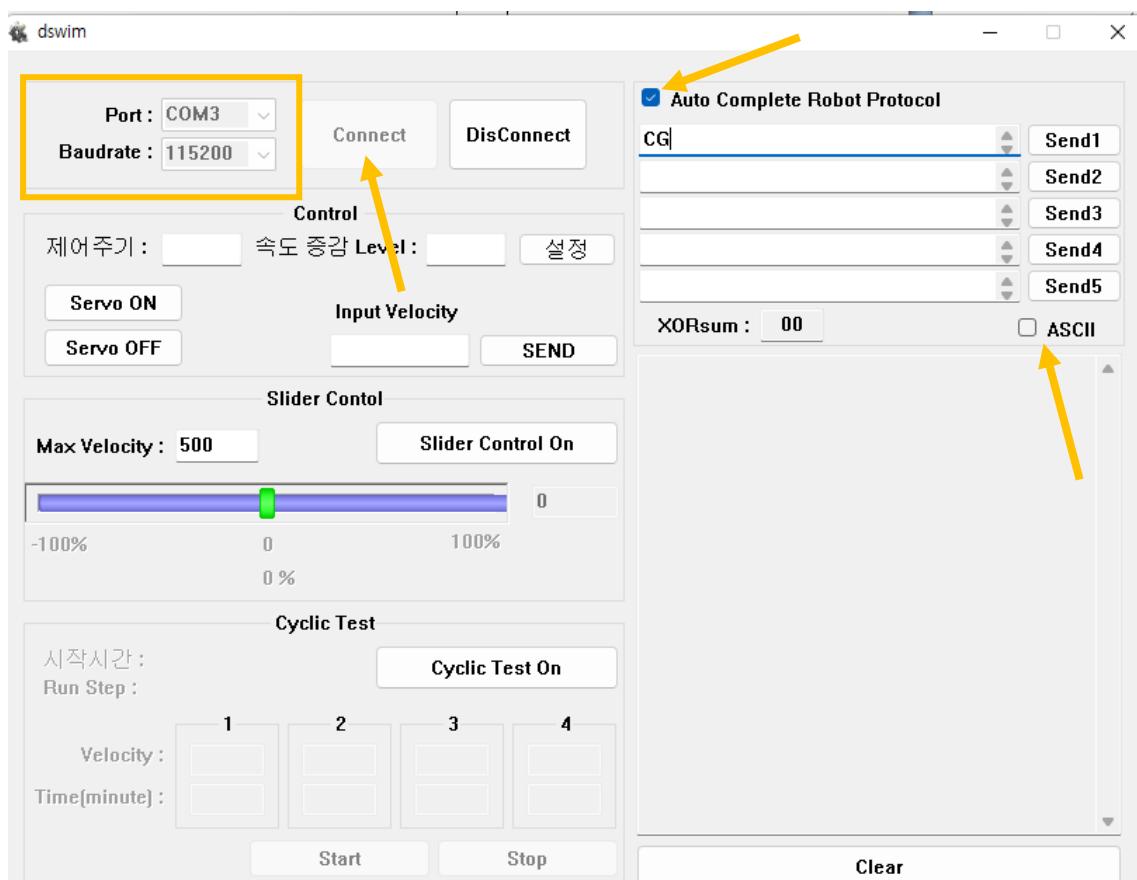
*SERVO ON 시에는 수행되지 않습니다.

4-5.프로토콜 실행 예제 (Example of Protocol execution)

Ex1) 현재 위치에서 1[m] 직진 후 90도 우회전 명령 내리기

0. dswim 실행, PORT, Baudrate 확인 및 Wheel Driver Board 와 Motor Connecting

- 1) 위의 [그림 3-16](#)을 참고하여 모터의 Encoder 케이블과 상 신호 케이블을 Back Link Board의 CN1(Left Encoder), CN5(Right Encoder), CN3(Left Phase), CN6(Right Phase)에 연결합니다.
- 2) Back Link Board의 CN20에 RS-232 케이블을 연결한 후 dswim을 실행합니다.
(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)



* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. 구동보드에 발생한 모든 Error 초기화. (선택사항)

Transmitter	STX	C	G	ETX	LRC
Receiver	STX	FLAG	ETX	LRC	

EMG, 범퍼센서 등 구동보드에 발생되었던 모든 Error를 초기화 시켜줍니다. (필수적인 명령은 아니지만 범퍼나 EMG 등에 의한 ERROR를 해제함으로써 명령 실행이 가능한 상태로 만들어 줍니다.)

2. 위치 제어 모드 변경

Transmitter	STX	C	Z	0	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

속도 제어 모드에서 위치 제어 모드를 내리게 되면 명령을 수행하지 못하기 때문에 위치 제어 모드로 변경해주어야 합니다.

3. Servo ON

Transmitter	STX	D	B	1	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

연결된 모터에 SERVO를 ON 시켜줍니다. SERVO ON 상태에서만 모터가 구동됩니다.

4. 현재 위치에서 1[m] 직진 명령

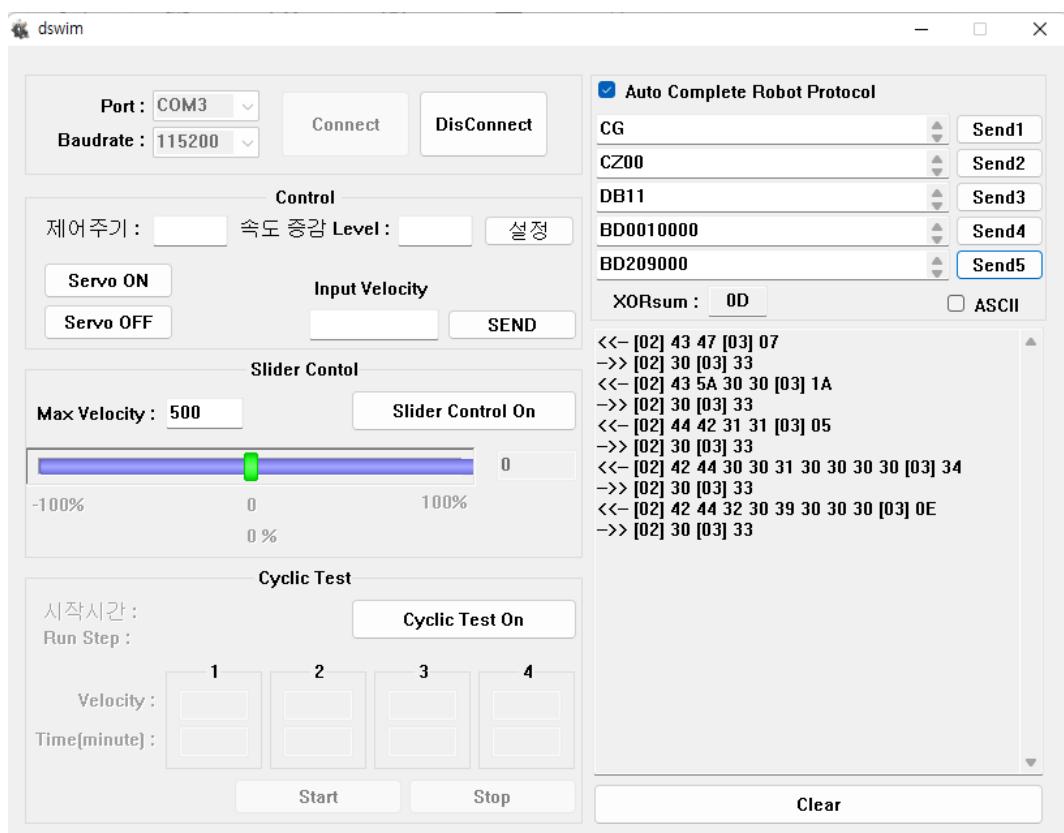
Transmitter	STX	B	D	0	0	10000	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

현재 위치에서부터 1[m]간 직진합니다.

5. 현재 위치에서 90도 우회전

Transmitter	STX	B	D	2	0	9000	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

현재 위치에서부터 90도 우회전 합니다.

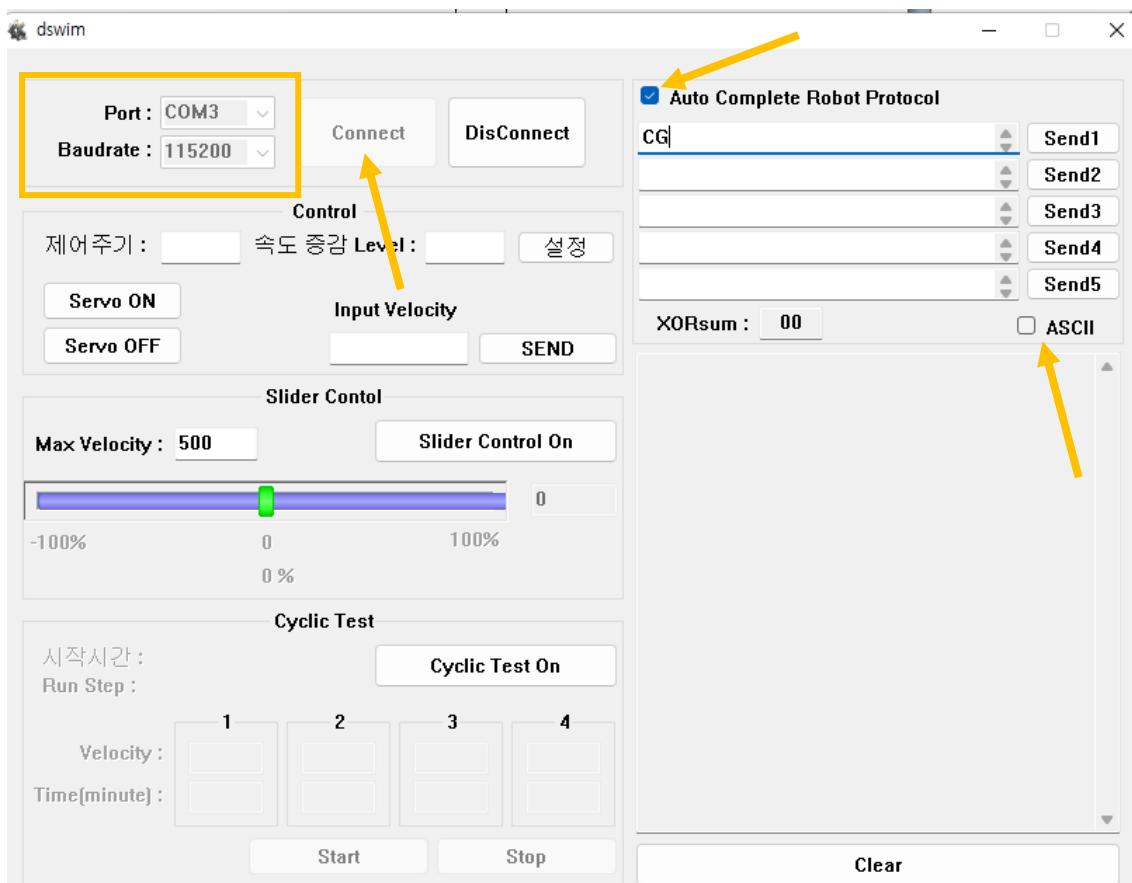


*Ex1의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

Ex2) 위치 제어 모드에서 모터 제어하기

0. dswim 실행, PORT, Baudrate 확인 및 Wheel Driver Board 와 Motor Connecting

- 1) 위의 [그림 3-16](#)을 참고하여 모터의 Encoder 케이블과 상 신호 케이블을 Back Link Board의 CN1(Left Encoder), CN5(Right Encoder), CN3(Left Phase), CN6(Right Phase)에 연결합니다.
- 2) Back Link Board의 CN20에 RS-232 케이블을 연결한 후 dswim을 실행합니다.
(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)



* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. 구동보드에 발생한 모든 Error 초기화. (선택사항)

Transmitter	STX	C	G	ETX	LRC
Receiver	STX	FLAG	ETX	ETX	LRC

EMG, 범퍼센서 등 구동보드에 발생되었던 모든 Error를 초기화 시켜줍니다. (필수적인 명령은 아니지만 범퍼나 EMG 등에 의한 ERROR를 해제함으로써 명령 실행이 가능한 상태로 만들어 줍니다.)

2. 위치 제어 모드 변경

Transmitter	STX	C	Z	0	0	ETX	LRC
Receiver	STX	FLAG	ETX	ETX	LRC		

속도 제어 모드에서 위치 제어 명령을 내리게 되면 명령을 수행하지 못하기 때문에 위치 제어 모드로 변경해주어야 합니다.

3. Servo ON

Transmitter	STX	D	B	1	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

연결된 모터에 SERVO를 ON 시켜줍니다. SERVO ON 상태에서만 모터가 구동됩니다.

4. 위치 제어 모드에서 200(mm/sec)의 속도로 모터 제어하기

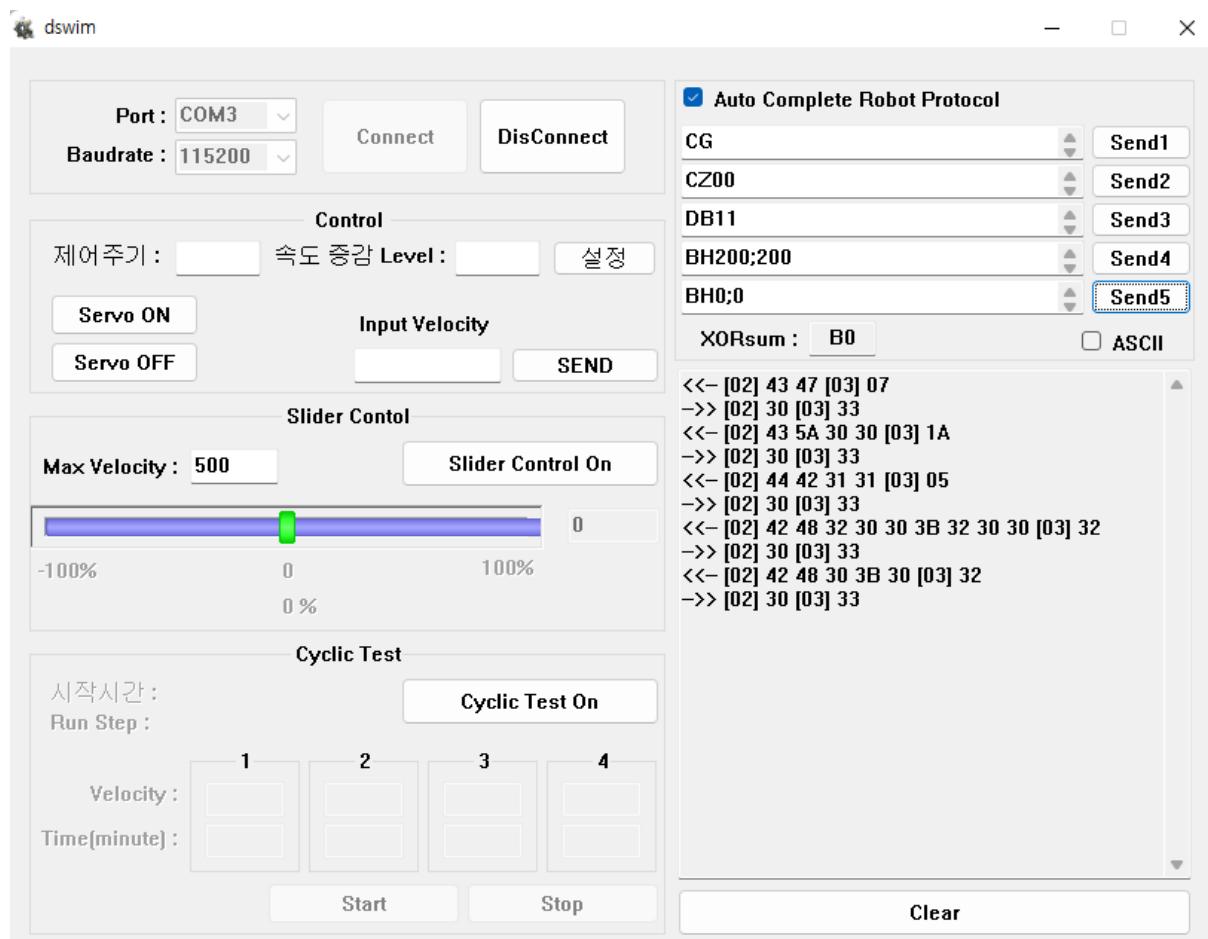
Transmitter	STX	B	H	200	;	200	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

200(mm/sec)의 속도로 직진합니다.

5. 위치 제어 모드에서 모터 정지시키기

Transmitter	STX	B	H	0	;	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

모터가 정지합니다.

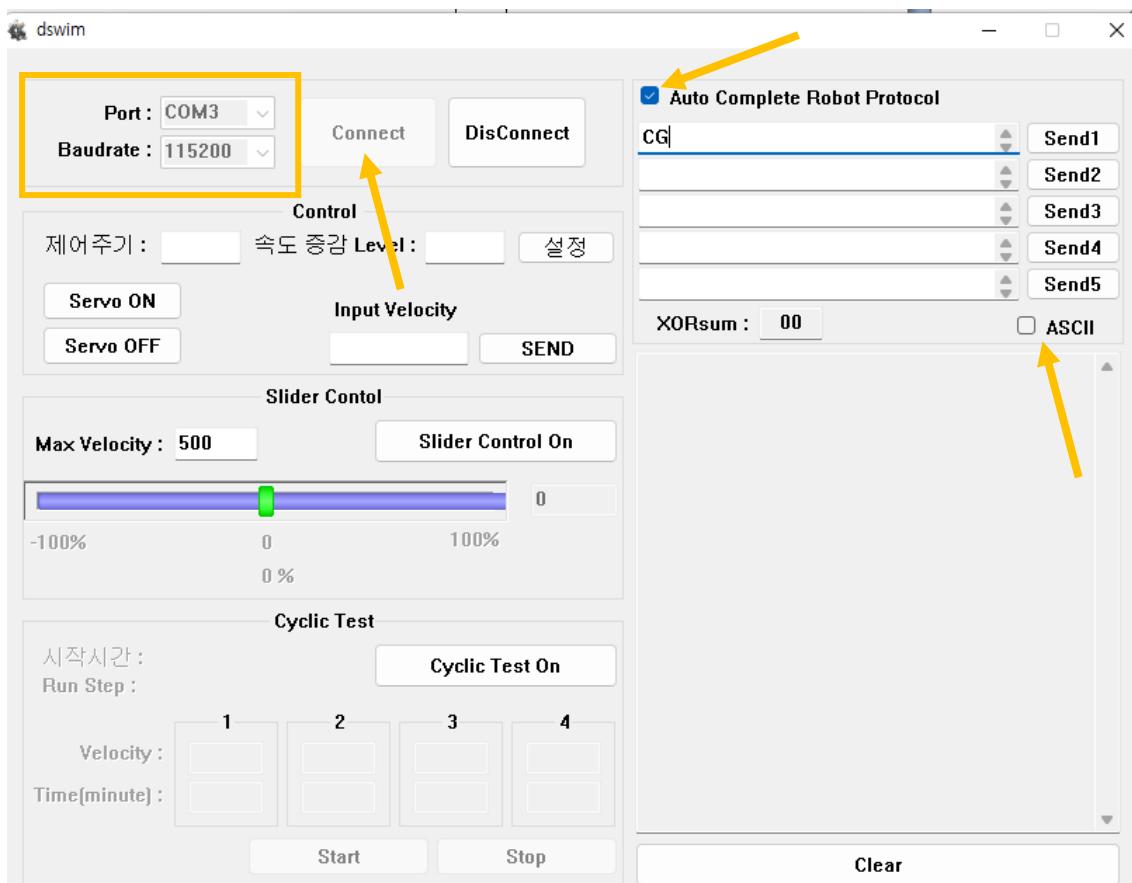


*Ex2의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

Ex3) 속도 제어 모드에서 모터 제어하기

0. dswim 실행, PORT, Baudrate 확인 및 Wheel Driver Board 와 Motor Connecting

- 1) 위의 [그림 3-16](#)을 참고하여 모터의 Encoder 케이블과 상 신호 케이블을 Back Link Board의 CN1(Left Encoder), CN5(Right Encoder), CN3(Left Phase), CN6(Right Phase)에 연결합니다.
- 2) Back Link Board의 CN20에 RS-232 케이블을 연결한 후 dswim을 실행합니다.
(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)



* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. 구동보드에 발생한 모든 Error 초기화. (선택사항)

Transmitter	STX	C	G	ETX	LRC
Receiver	STX	FLAG	ETX	ETX	LRC

EMG, 범퍼센서 등 구동보드에 발생되었던 모든 Error를 초기화 시켜줍니다. (필수적인 명령은 아니지만 범퍼나 EMG 등에 의한 ERROR를 해제함으로써 명령 실행이 가능한 상태로 만들어 줍니다.)

2. 속도 제어 모드 변경

Transmitter	STX	C	Z	1	1	ETX	LRC
Receiver	STX	FLAG	ETX	ETX	LRC		

위치 제어 모드에서 속도 제어 명령을 내리게 되면 명령을 수행하지 못하기 때문에 속도 제어 모드로 변경해주어야 합니다.

3. Servo ON

Transmitter	STX	D	B	1	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

연결된 모터에 SERVO를 ON 시켜줍니다. SERVO ON 상태에서만 모터가 구동됩니다.

4. 속도 제어 모드에서 320(mm/sec)로 모터 제어하기

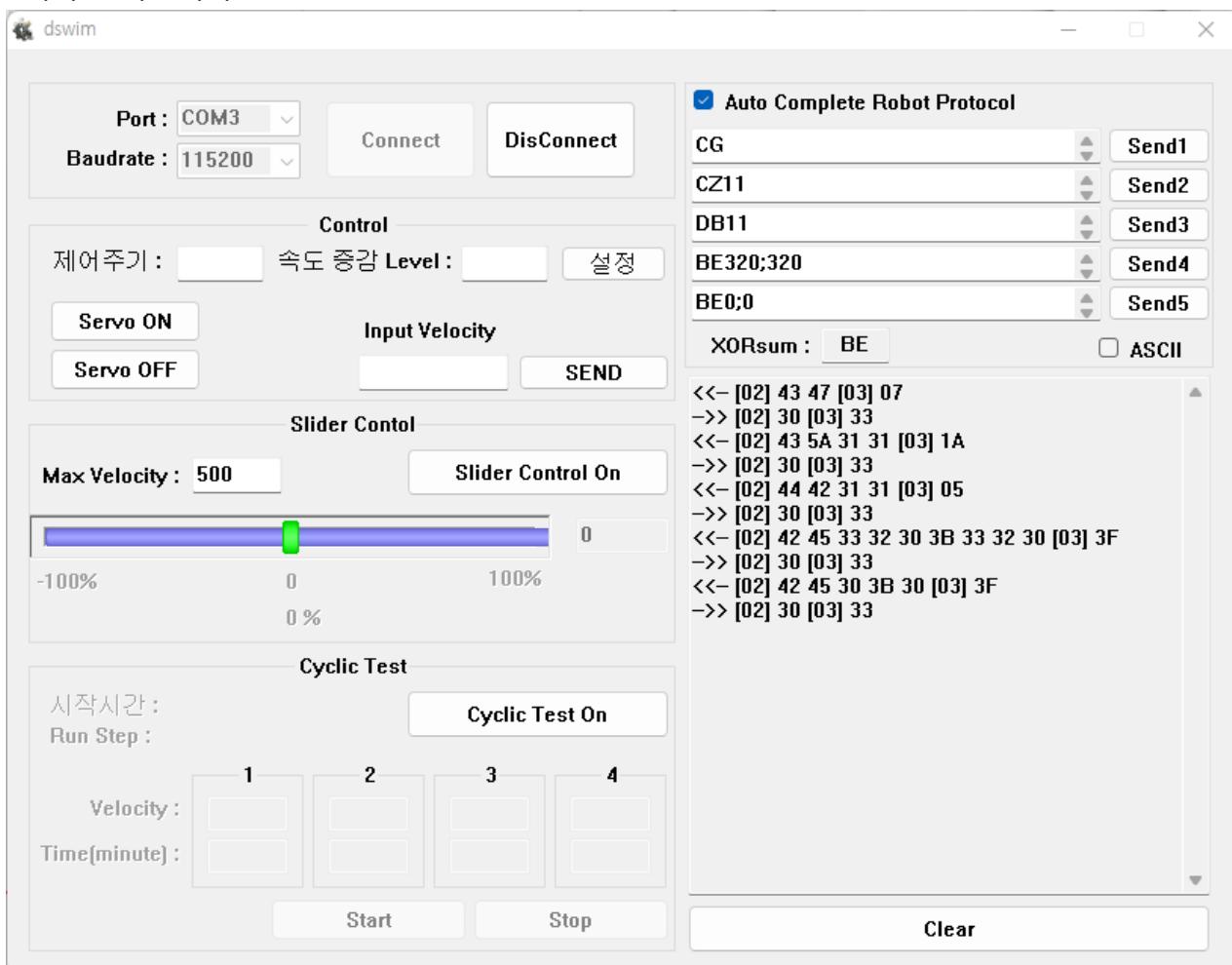
Transmitter	STX	B	E	320	;	320	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

320(mm/sec)의 속도로 직진합니다.

5. 속도 제어 모드에서 모터 정지시키기

Transmitter	STX	B	E	0	;	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

모터가 정지합니다.



*Ex3의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

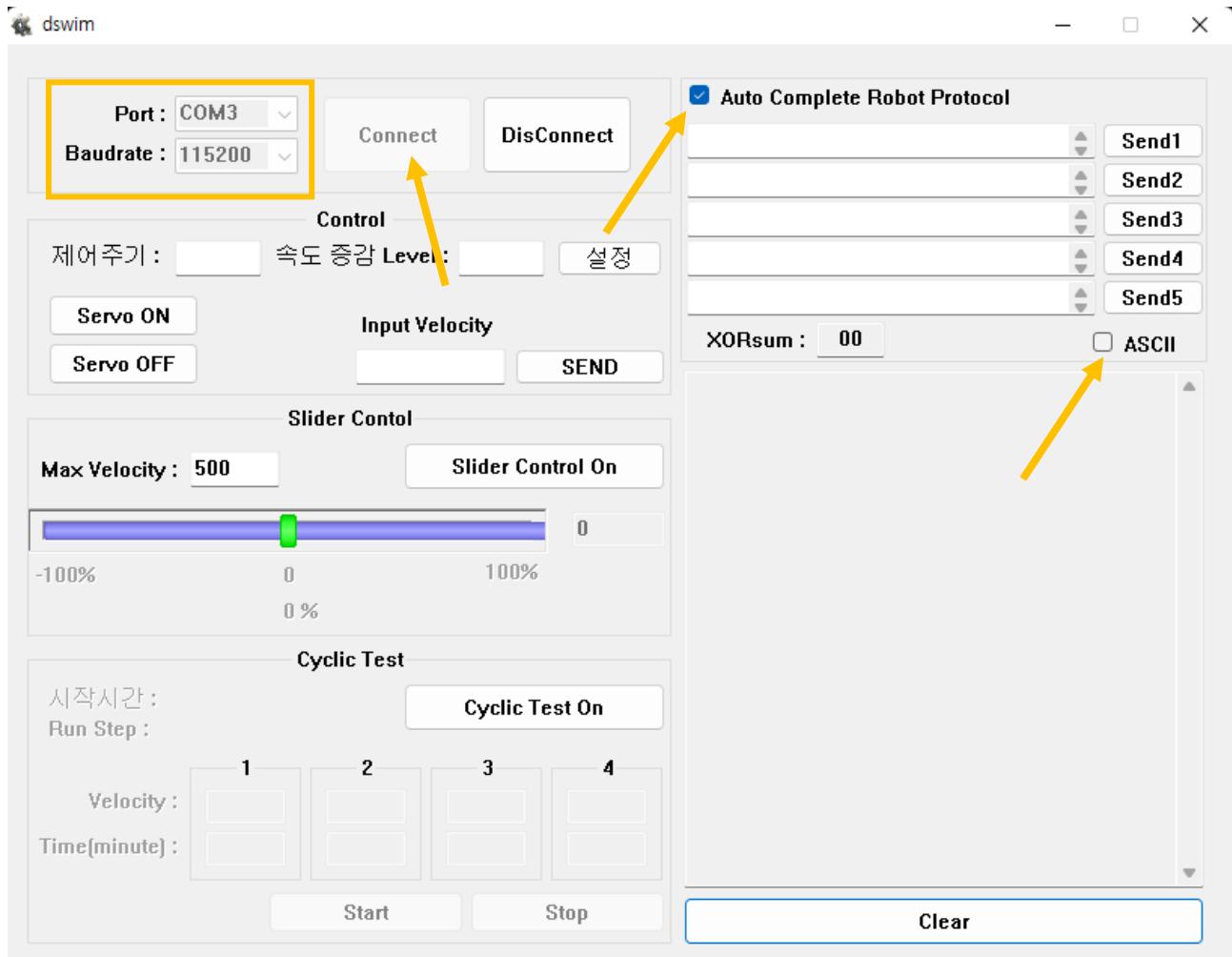
Ex4) 원하는 가속도, 밝기, 색상으로 LED 제어하기

0. dswim 실행, PORT, Baudrate 확인 및 Power Senser Board와 LED Connecting

1) 위의 [그림 3-16](#)을 참고하여 LED와 Back Link Board의 CN9(Left), CN8(Right)을 연결합니다.

2) Back Link Board의 CN24에 RS-232 케이블을 연결한 후 dswim을 실행합니다

(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)



* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. LED 색상 정의 (흰색) 및 LED ON

Data	LED Port (Binary)							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	NC	NC	RIGHT BLUE	RIGHT RED	RIGHT GREEN	LEFT BLUE	LEFT RED	LEFT GREEN

*Data 형식은 ASCII입니다. 위의 표를 참고하여 binary (2진수) 값을 ASCII (10진수) 값으로 변환해서 입력해주면 HEX(16진수)값으로 자동 변환되어 LED bit로 표현됩니다.

Transmitter	STX	L	T	6	3	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

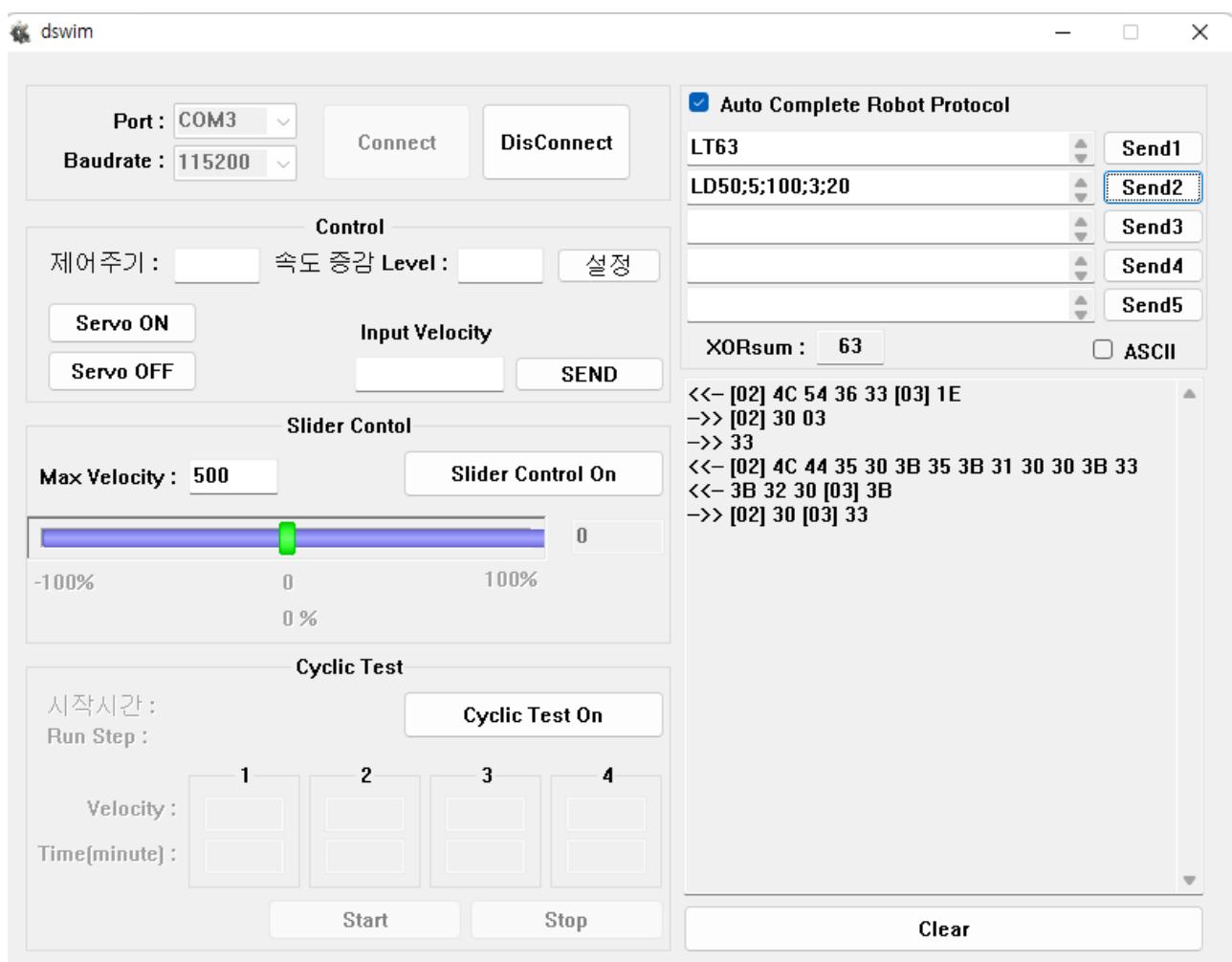
ASCII(10진수) 값 63은 Binary (2진수) 값으로 0011 1111 입니다. 따라서 LEFT, RIGHT의 RED, GREEN, BLUE를 전부 ON합니다. (빛의 삼원색이 혼합된 색인 흰색이 표출되나 LED 각각의 오차로 인해 완벽한 흰색의 빛이 나오지는 않을 수도 있습니다.)

2. 흰색의 LED가 5의 속도로 100% 밝기까지 켜졌다가 3의 속도로 20% 밝기까지 꺼지는 반복 명령 내리기

*자세한 내용은 [페이지 20](#) 을 참고하세요

Transmitter	STX	L	D	50	;	5	;	100	;	3	;	20	ETX	LRC
Receiver	STX	FLAG	ETX	LRC										

흰색의 LED가 50으로 Left, Right중 Toggle하지 않는 LED 밝기를 지정 후 5의 속도로 100%까지 켜졌다 3의 속도로 20%까지 꺼지는 상태가 반복됩니다.



*Ex4의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

Ex5) Sonar 제어하기

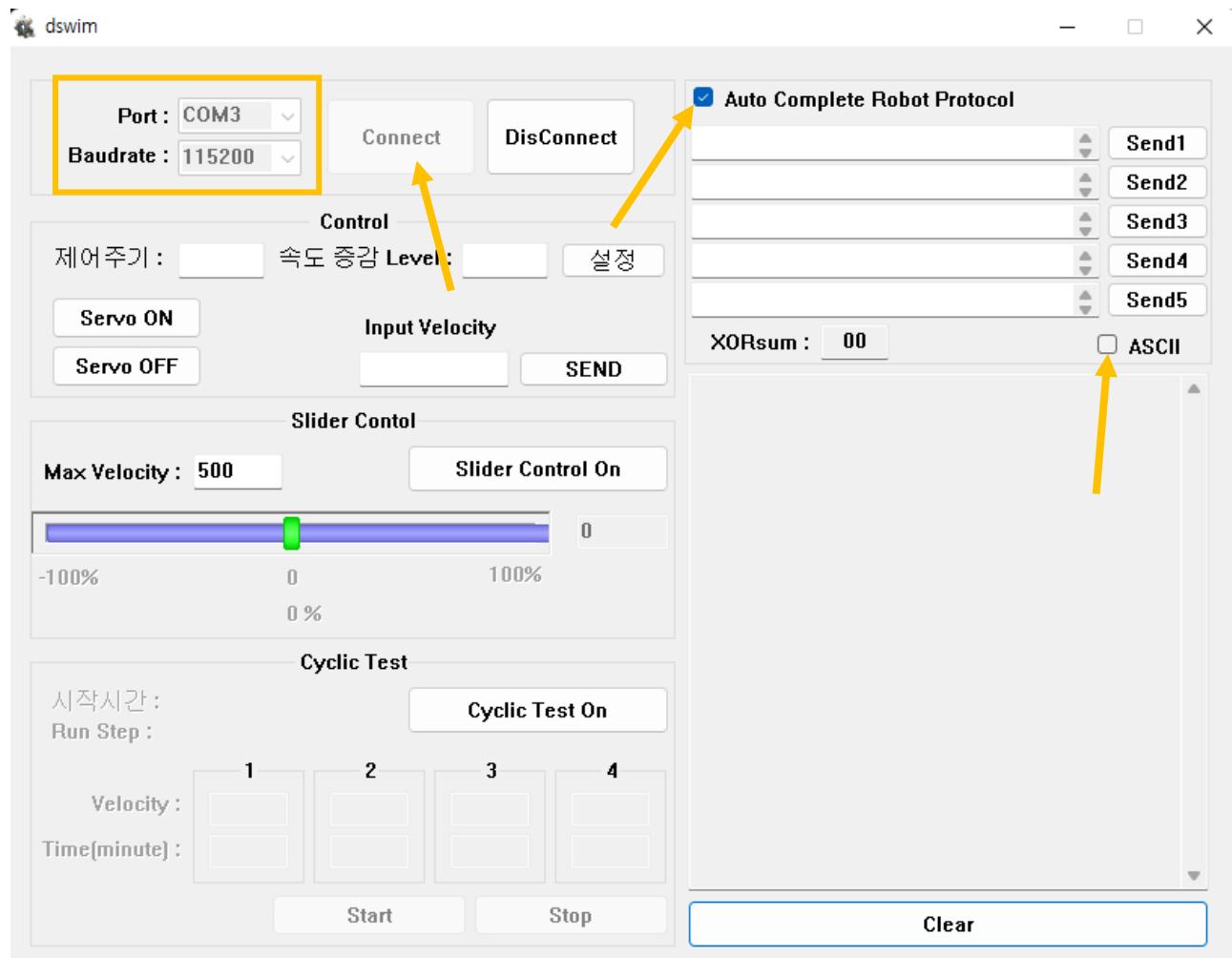
0. dswim 실행, PORT, Baudrate 확인 및 Power Senser Board와 Sonar Senser Connecting

1) 위의 [그림 3-16](#)을 참고하여 Sonar 케이블과 Back Link Board의 CN34를 연결합니다.

2) Back Link Board의 CN24에 RS-232 케이블을 연결한 후 dswim을 실행합니다

(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)

*참고 – 당사의 TETRA-DS V에 적용된 초음파 센서는 (주) 하기소닉의 HG-B40C (5v) 제품입니다.



* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. 초음파 센서 LOOP 동작 ON 시키기

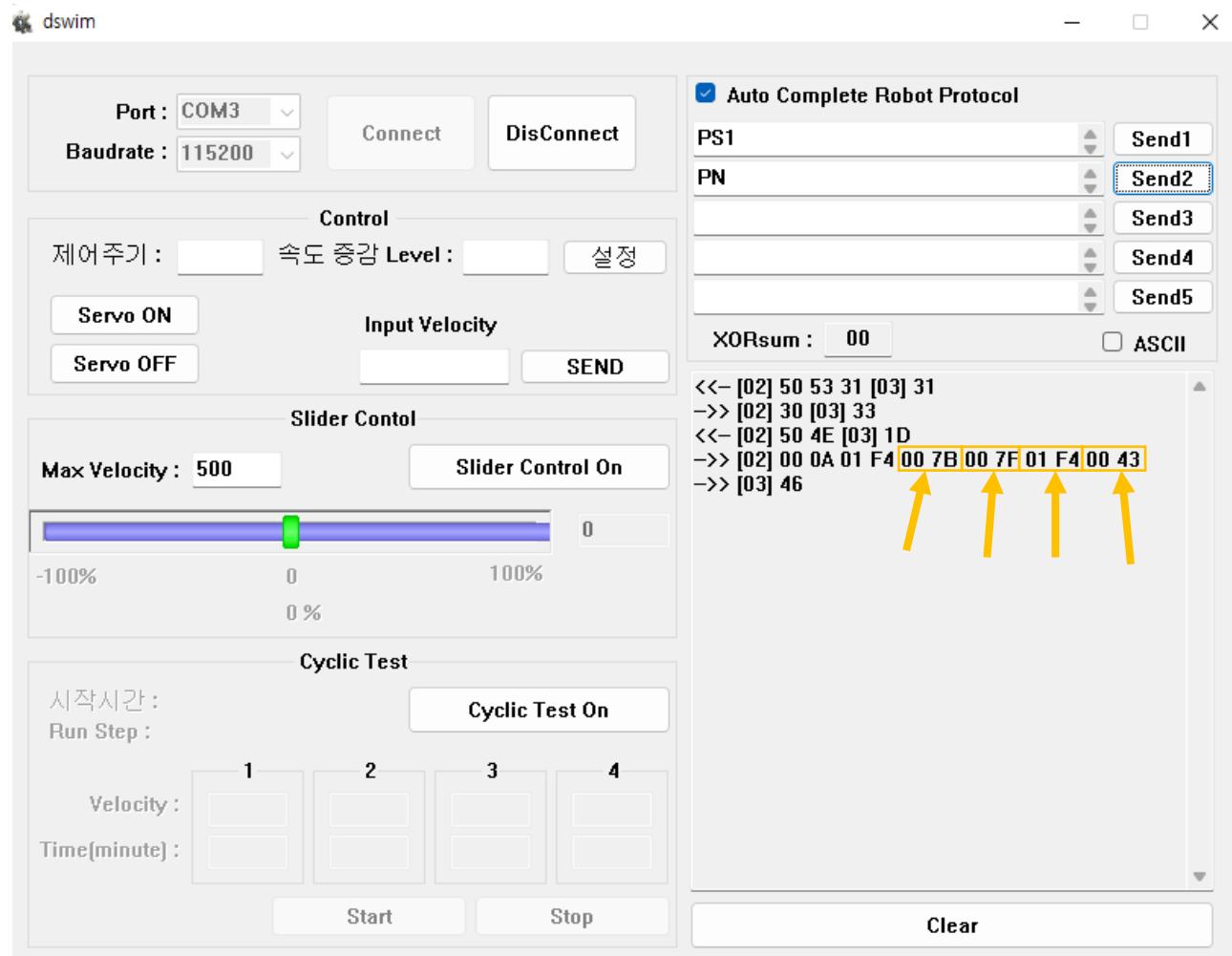
Transmitter	STX	P	S	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

PS1 명령을 사용하게 되면 Power Senser Board에 내장된 초음파 센서 LOOP를 동작 시킵니다. 동작을 멈추려면 PS0 명령을 내리면 됩니다.

2. Sonar Senser Data 읽어오기 (2-3-9 참고)

Transmitter	STX	P	N	ETX	LRC										
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	D2_H	D2_L	D3_H	D3_L	D4_H	D4_L	ETX	LRC

이런 식으로 PN 명령을 내리면 Sonar Senser Data를 읽어옵니다. 정상적인 명령 수행 시 아래의 그림처럼 왼쪽 화살표부터 Sonar1~4의 값이 출력되는 것을 확인할 수 있습니다.



*Ex5의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

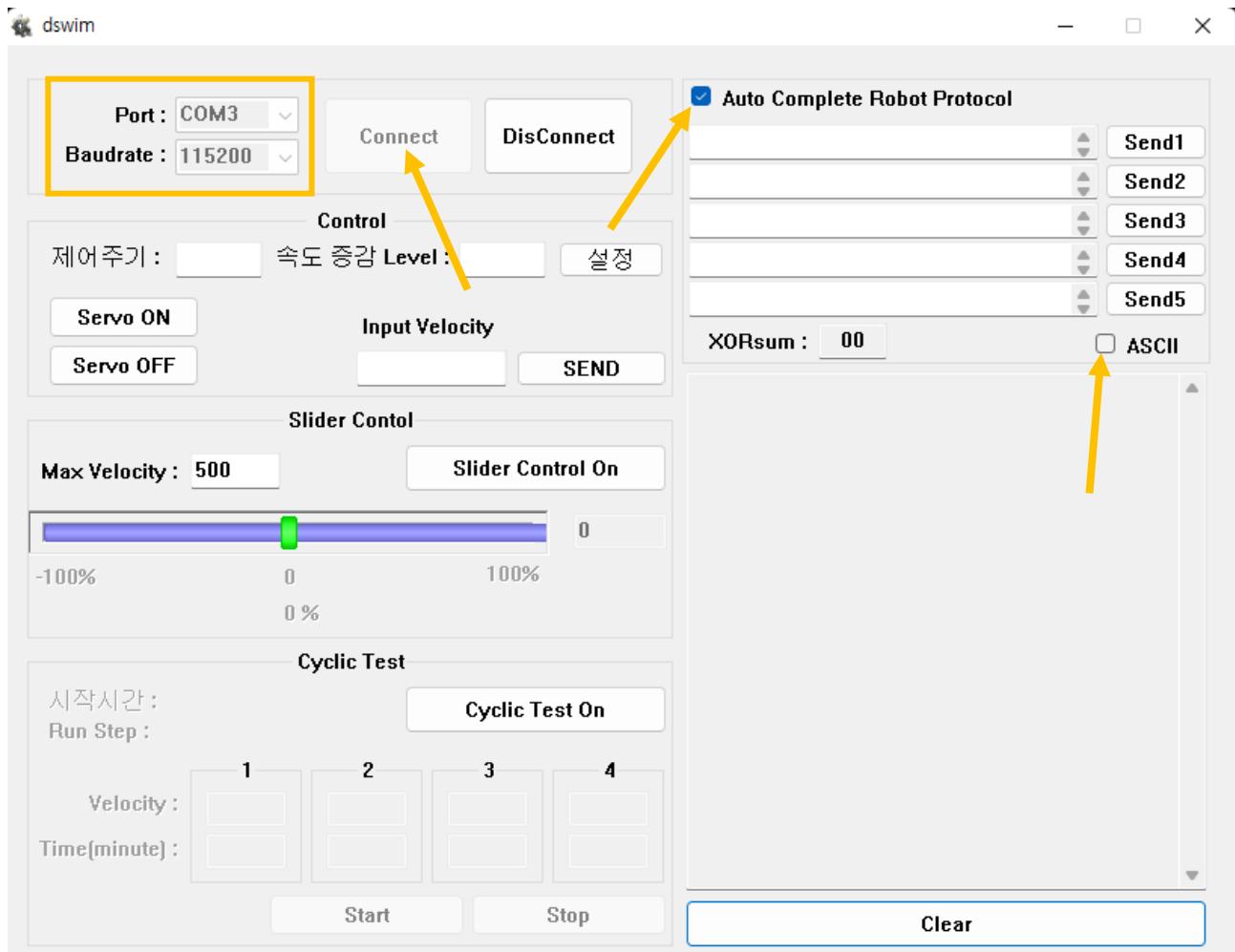
Ex6) Bumper 센서 동작 상태 확인하기

0. dswim 실행, PORT, Baudrate 확인 및 Power Senser Board와 Bumper Senser Connecting

1) 위의 [그림 3-16](#)을 참고하여 BUMPER 케이블과 Back Link Board의 CN39를 연결합니다.

2) Back Link Board의 CN20에 RS-232 케이블을 연결한 후 dswim을 실행합니다

(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)



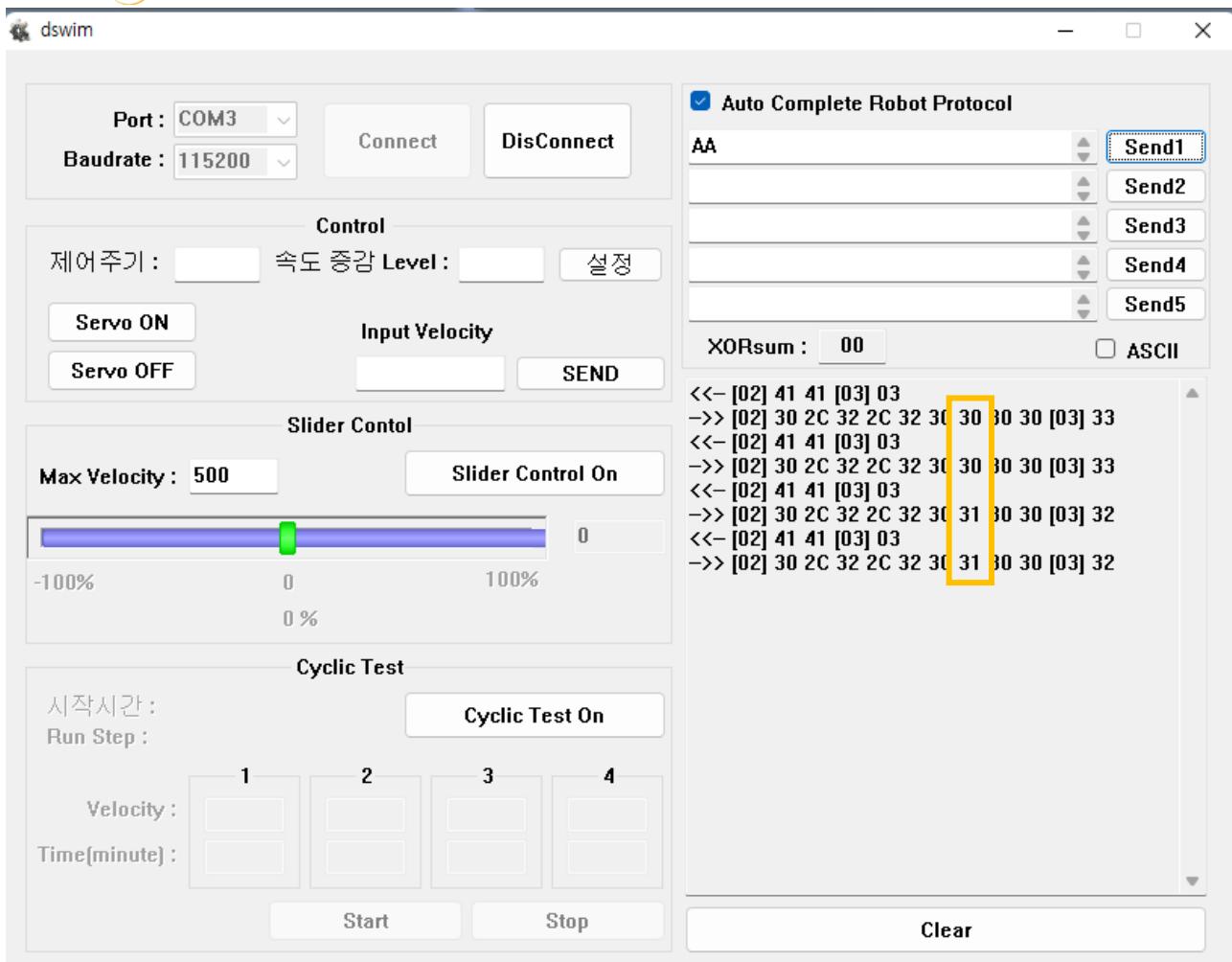
* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. 범퍼 센서 동작 ON 시키기

범퍼 센서는 대상이 접촉되었을 경우, True 값을 발생시키고 접촉 상태가 해제되었을 경우에는 False 값을 발생시킵니다.

따라서, AA 프로토콜 명령을 통해 범퍼 센서의 동작 상태를 확인할 수 있습니다. ([2-4-1 참고](#))

Transmitter	STX	A	A	ETX	LRC								
Receiver	STX	FLAG	DATA0	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6	DATA7	ETX	LRC	



*Ex6의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

위의 그림과 같이 범퍼센서가 작동 시 AA 명령을 내리면 31이 출력되는 것을 확인할 수 있습니다.
참고로 범퍼센서가 작동되면 C-BOX에 연결된 모터는 자동으로 비상 정지 상태가 됩니다.



*위 사진은 예시를 위한 당사의 TETRA-DS V 사진으로 위와 같이 BUMPER 센서를 활용할 수 있습니다.

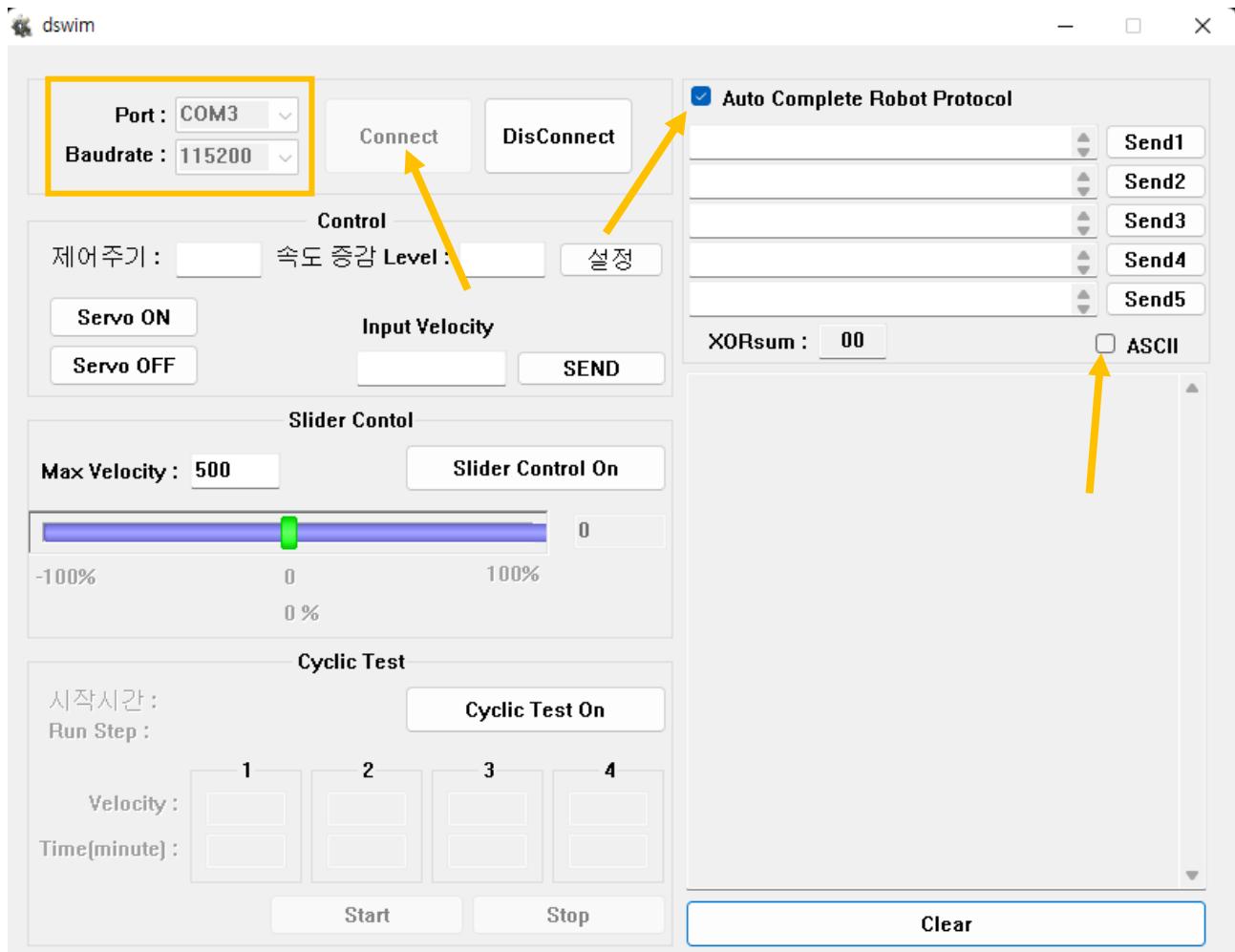
Ex7) EMS 스위치 동작 상태 확인하기

0. dswim 실행, PORT, Baudrate 확인 및 Power Senser Board와 EMS SWITCH Connecting

1) 위의 [그림 3-18](#)을 참고하여 EMS 케이블과 Back Link Board의 CN4를 연결합니다.

2) Back Link Board의 CN20에 RS-232 케이블을 연결한 후 dswim을 실행합니다

(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)



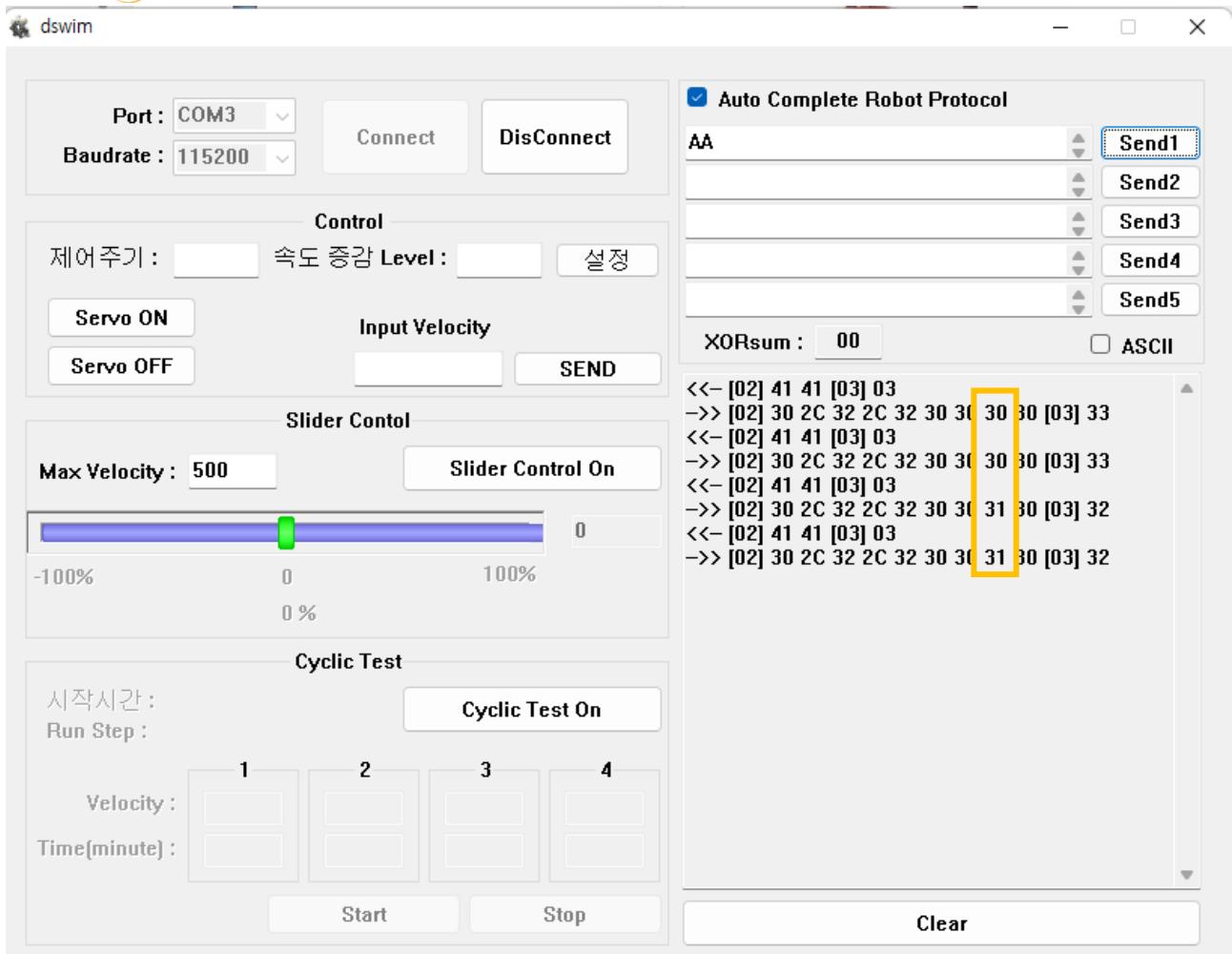
* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. EMS 스위치 동작 ON 시키기

EMS 스위치는 스위치가 눌러졌을 때 True 값을 발생시키고 접촉 상태가 해제되었을 경우에는 False 값을 발생시킵니다.

따라서, AA 프로토콜 명령을 통해 범퍼 센서의 동작 상태를 확인할 수 있습니다. ([2-4-1 참고](#))

Transmitter	STX	A	A	ETX	LRC								
Receiver	STX	FLAG	DATA0	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6	DATA7	ETX	LRC	



*Ex7의 모든 명령이 정상적으로 실행되었을 때의 dswim 화면입니다.

위의 그림과 같이 EMS 스위치 ON 시 AA 명령을 내리면 31이 출력되는 것을 확인할 수 있습니다.
참고로 EMS 스위치가 ON 되면 C-BOX에 연결된 모터는 자동으로 비상 정지 상태가 됩니다.

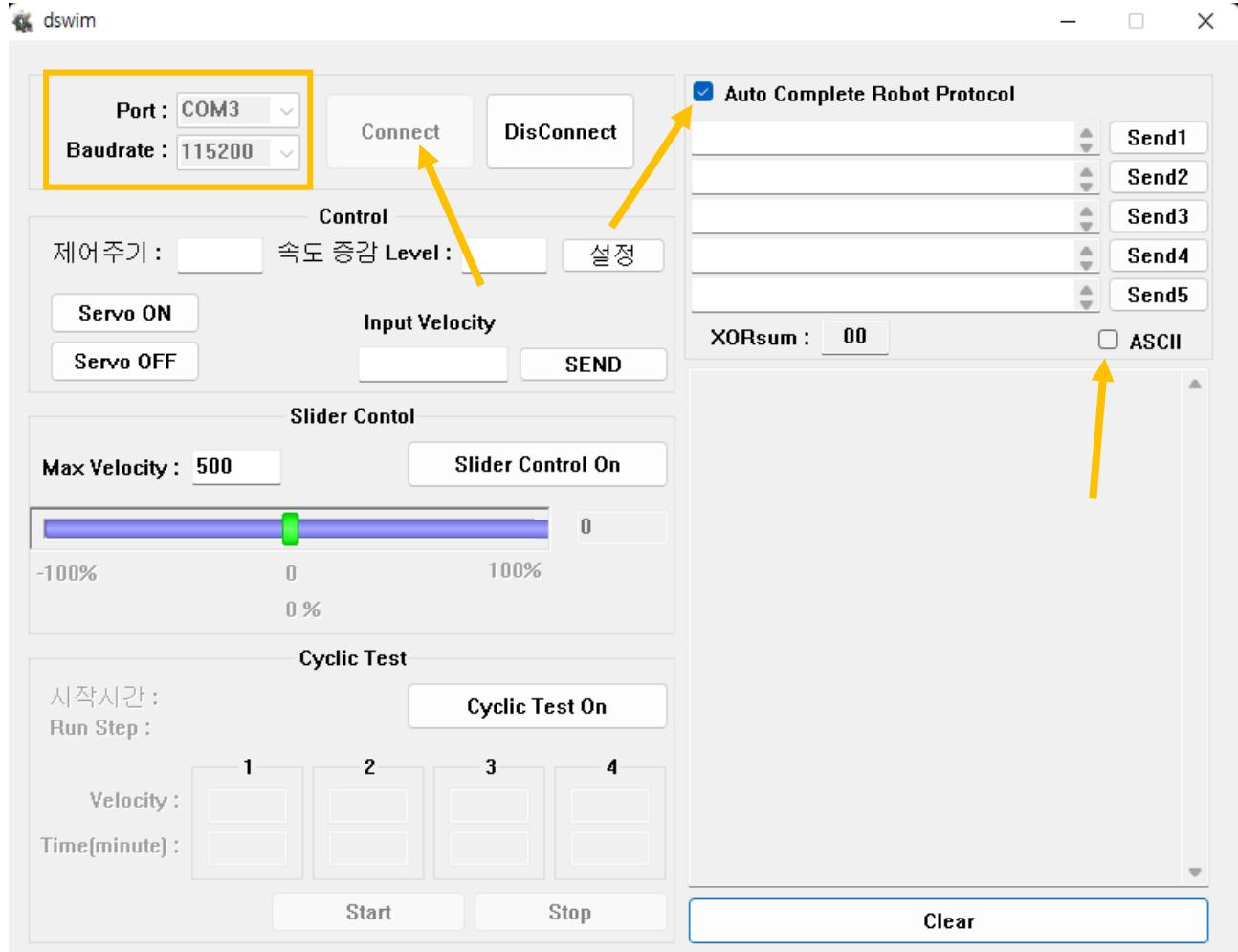


*위 사진은 예시를 위한 당사의 TETRA-DS V 사진으로 위와 같이 EMS 스위치를 활용할 수 있습니다.

Ex8) I.O PORT를 사용하여 RELAY 제어하기

0. dswim 실행, PORT, Baudrate 확인 및 Power Senser Board와 I.O PORT Connecting

- 1) 위의 [그림 3-13](#)을 참고하여 I.O 케이블과 Power Senser Board 전면의 CN15를 연결합니다.
- 2) Back Link Board의 CN24에 RS-232 케이블을 연결한 후 dswim을 실행합니다
(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)

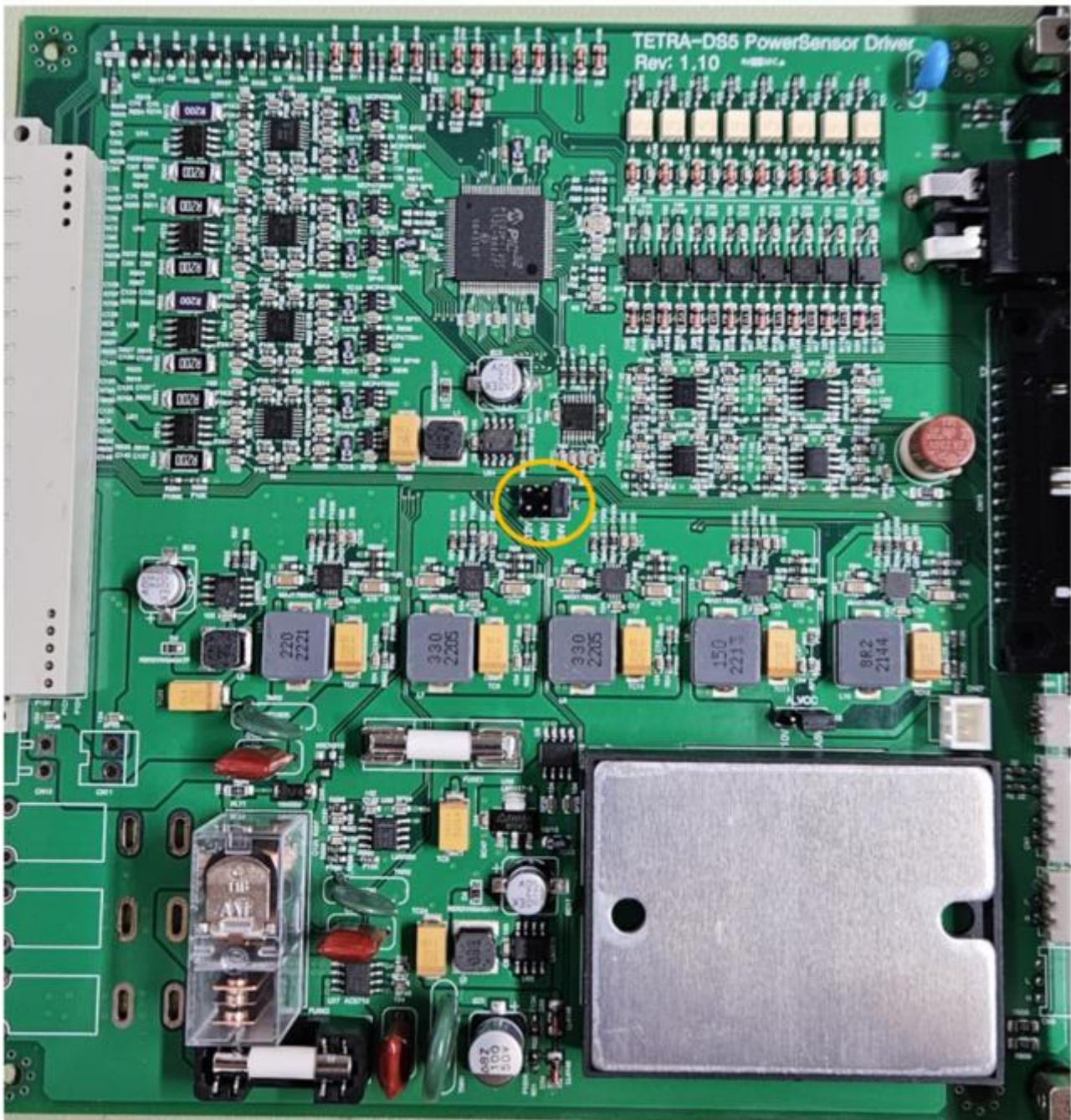


* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안나올 수도 있습니다.)

1. JUMPER 연결하기

Power Senser Board에서 J11을 찾아 검은색 커버를 사용하는 I.O의 전압의 위치에 꽂습니다.

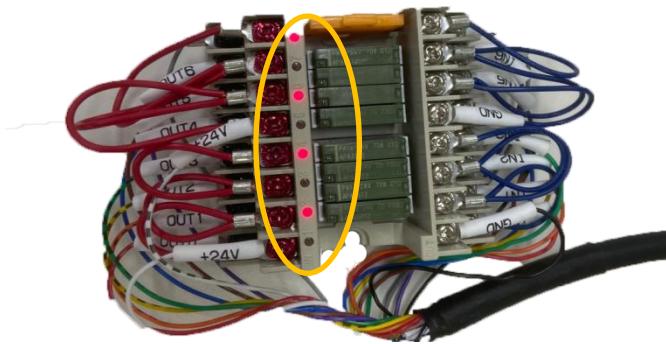
Ex8) 예제에 사용되는 I.O 포트는 24V (24V동작 RELAY)로 연결하였습니다.



2. PO 명령으로 2,4,6,8 RELAY만 ON하기

Transmitter	STX	P	O	1	7	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

2,4,6,8 RELAY만 ON 해야 하므로 Hex 값으로 1010 1010 이고 10진수로 170입니다. PO170을 수행합니다.



정상적으로 명령 수행 시 앞의 사진과 같이 LED가 ON됩니다.

3. OUT PORT OFF하기

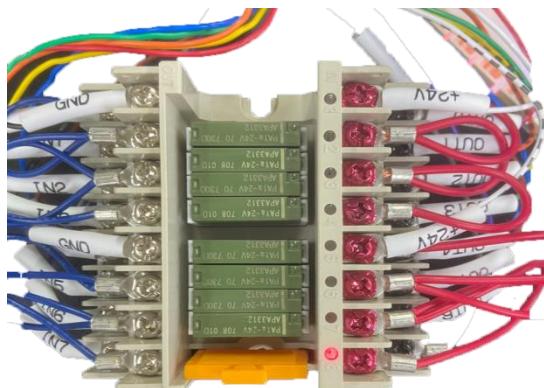
Transmitter	STX	P	O	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

포트의 상태까지 확인을 마쳤으면 OUT PORT를 OFF합니다. OUT PORT를 ON할 때와 같은 방식이며 PO0 명령을 내리면 RELAY가 OFF됩니다.

4. Po 명령으로 RELAY ON하기

이번엔 Po명령을 통해 RELAY 하나만 ON해보겠습니다.

Transmitter	STX	P	o	7	:	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				



정상적으로 명령 수행 시 앞의 사진과 같이 LED가 ON됩니다.

5. Po 명령으로 OUT PORT OFF하기

Transmitter	STX	P	o	7	:	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

포트의 상태까지 확인을 마쳤으면 OUT PORT를 OFF합니다. OUT PORT를 ON할 때와 같은 방식이며 Po7;0 명령을 통해 RELAY를 OFF 합니다.

위와 같이 PO 명령을 통해 여러 가지의 RELAY를 한 번에 같이 제어할 수 있고, PO 명령을 통해 RELAY를 하나씩 제어할 수도 있습니다.

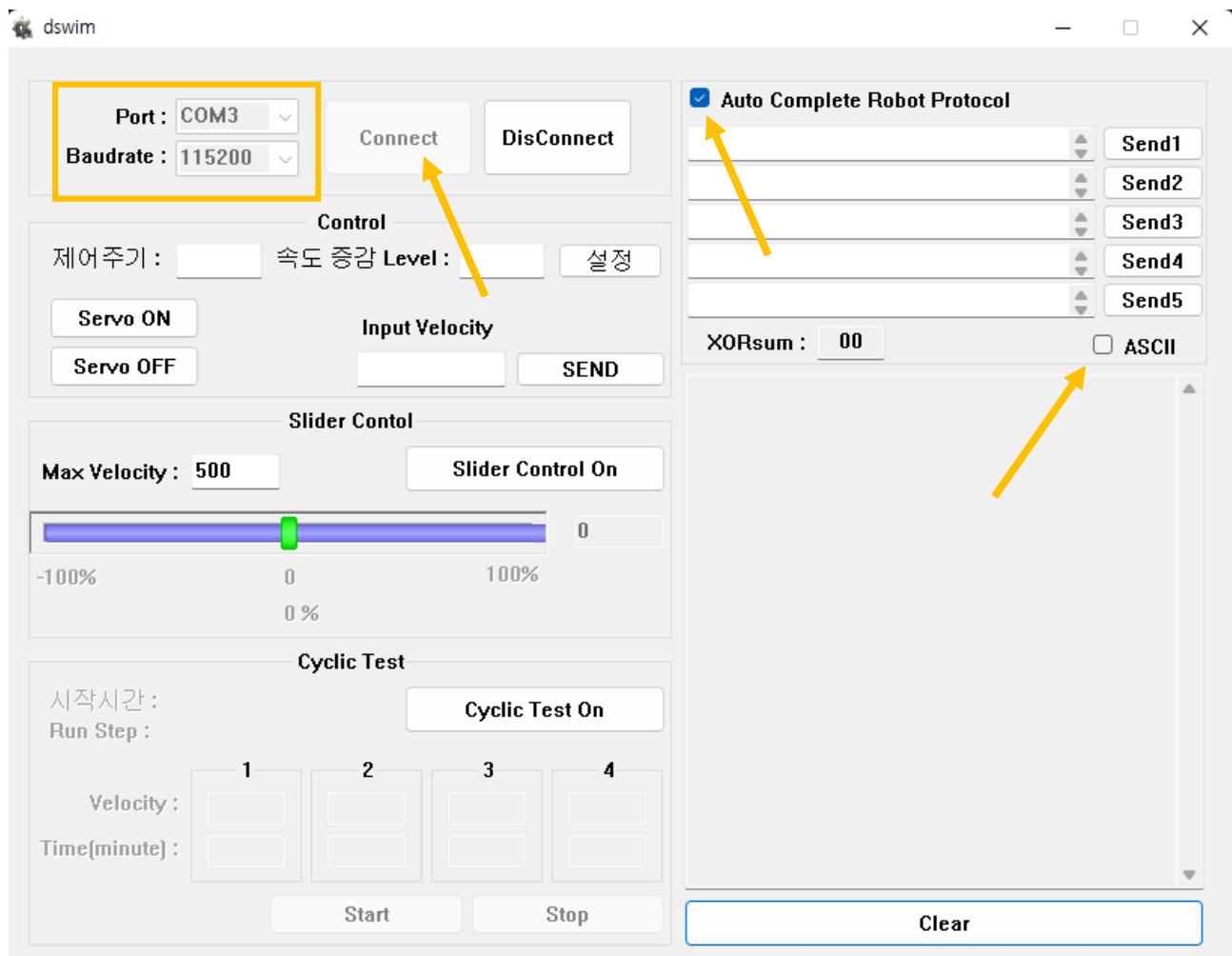
Ex9) Power Senser Board에 공급되는 전원 제어하기

0. dswim 실행, PORT, Baudrate 확인 및 Power Senser Board와 I.O PORT Connecting

1) Back Link Board의 CN24에 RS-232 케이블을 연결합니다.

2) dswim을 실행합니다

(PC의 PORT 번호를 모를 때에는 제어판 -> 장치관리자 -> 포트(COM & LPT)에서 확인)

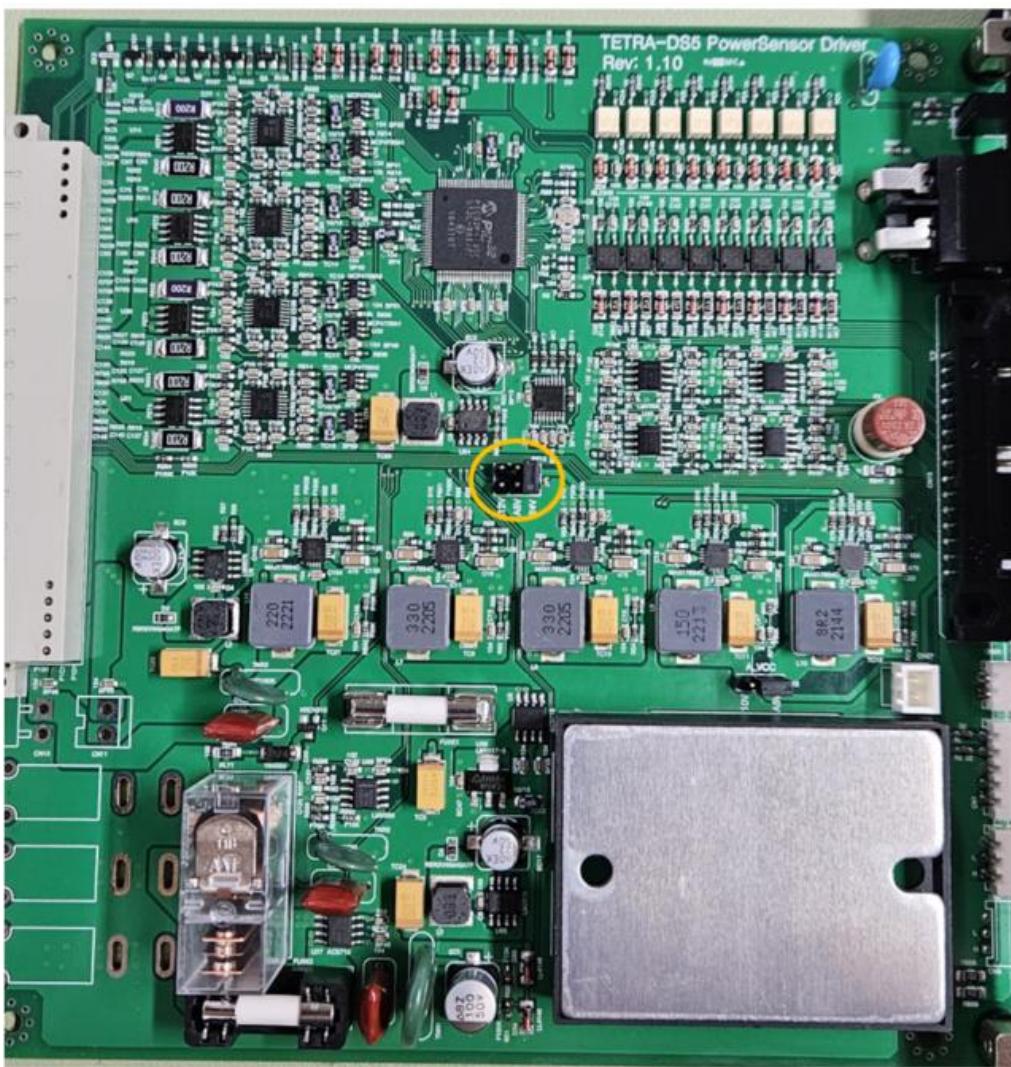


* Auto Complete Robot Protocol 체크, ASCII 체크 해제 후 사용하시면 편리합니다. (다만, ASCII 코드를 사용하는 통신 방식에서는 ASCII 체크박스에 체크하지 않으면 원하는 결과 값이 안 나올 수도 있습니다.)

1. JUMPER 연결하기

Power Senser Board에서 J11을 찾아 검은색 커버를 사용하는 I.O의 전압의 위치에 끌습니다.

Ex9) 예제에 사용되는 I.O 포트는 24V (24V동작 RELAY)로 연결하였습니다.



2. PE 명령으로 TETRA-DSV Power Sensor Board 모듈에 공급되는 전원 모두 ON하기

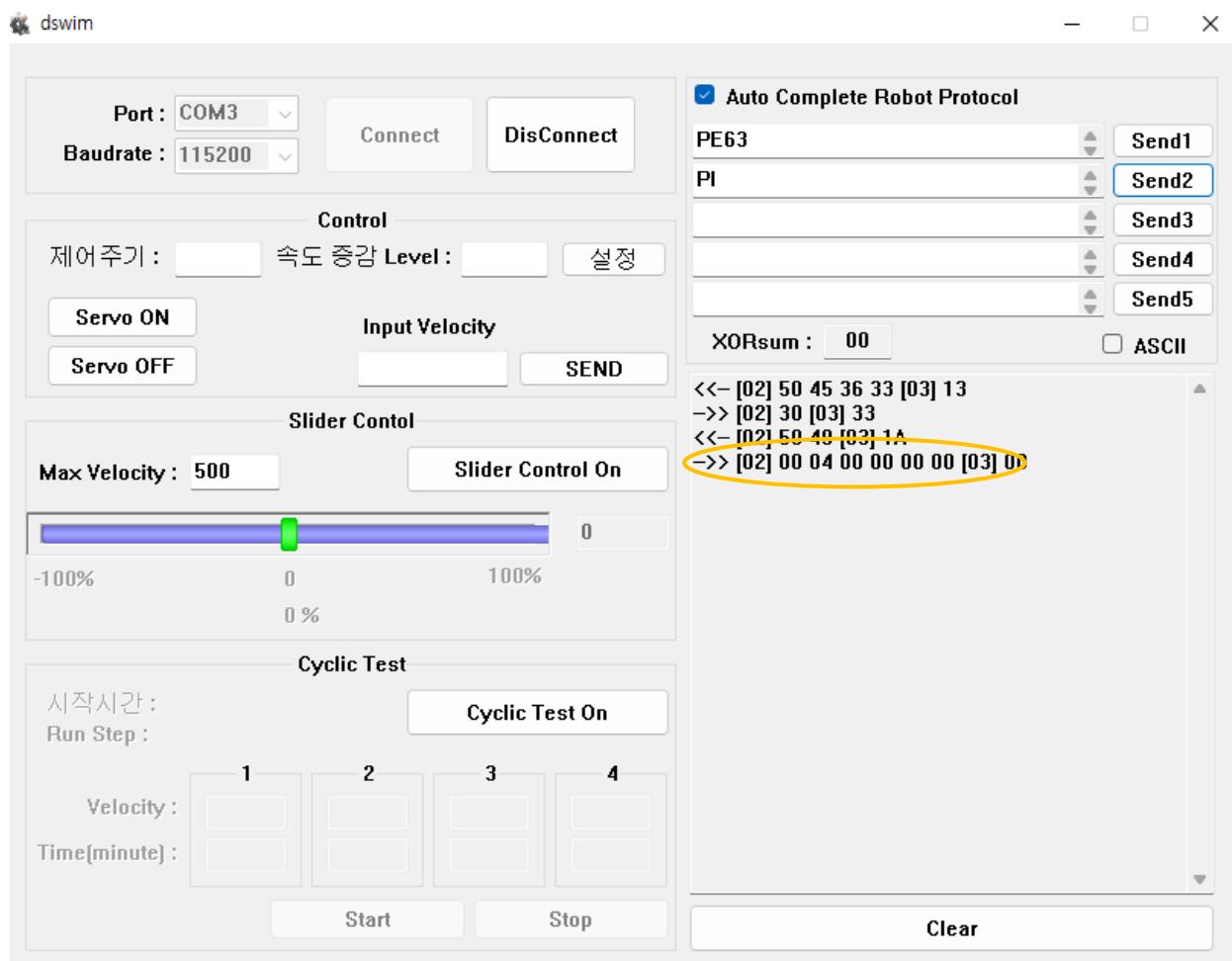
Transmitter	STX	P	E	6	3	ETX	LRC
Receiver	STX	FLAG	ETX	LRC			

2-3-5를 참고하여 TETRA-DSV Power Senser Board 모듈에 공급되는 모든 전원을 ON 합니다.

3. PI 명령으로 TETRA-DSV Power Sensor Board에 공급되는 전원 모두 ON 되었는지 확인하기

Transmitter	STX	P	I	ETX	LRC
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L

D1_H	D1_L	ETX	LRC
------	------	-----	-----



정상적으로 명령 수행 시 앞의 사진과 같이 출력됩니다.

4. TETRA-DSV Power Sensor Board 모듈에 공급되는 전원 모두 OFF하기

Transmitter	STX	P	I	0	ETX	LRC
Receiver	STX	FLAG	ETX	LRC		

명령이 내려졌는지 확인을 마쳤으면 Power Sensor Board 모듈에 공급되는 모든 전원을 OFF합니다. 전원을 ON할 때와 같은 방식이며 PI0 명령을 내리면 모든 전원이 OFF됩니다.

* 참고로 전원이 OFF 된 상태에서는 C-BOX의 Fan이 돌지 않게 됩니다. 이는 전원이 차단되었다는 뜻을 의미합니다.

5. Pe 명령으로 Power Senser Board 모듈에 공급되는 전원 ON하기

이번엔 Pe 명령을 통해 Power Sensor Board 모듈에 공급되는 전원을 하나만 ON 해보겠습니다.

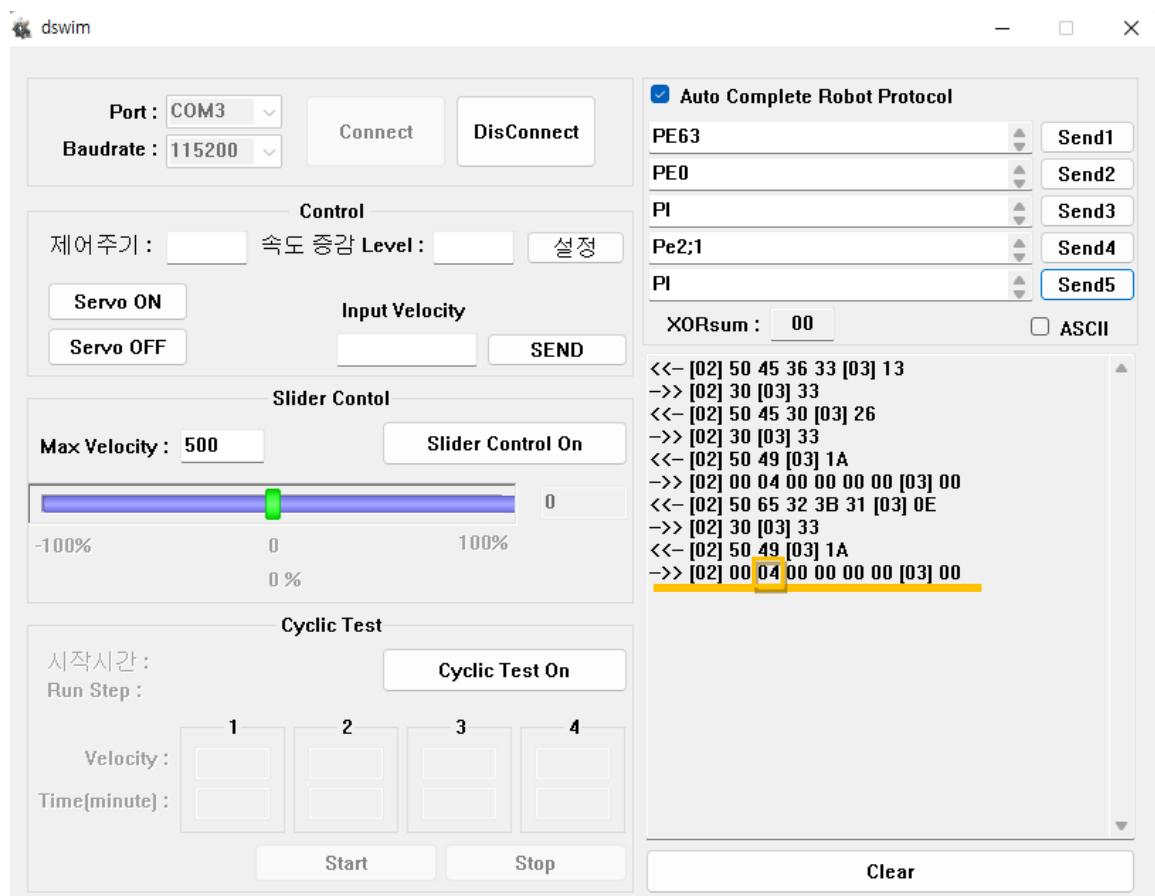
Transmitter	STX	P	e	2	:	1	ETX	LRC
Receiver	STX	FLAG	ETX	LRC				

Pe2;1 명령이 정상적으로 내려지면 다시 Fan이 돌기 시작하는 것을 볼 수 있습니다.

Pe2;1 명령은 메인 전원인 24V POWER1을 ON 하는 명령이기 때문입니다.

6. PI 명령으로 TETRA-DSV Power Sensor Board에 공급되는 24V 전원이 ON 되었는지 확인하기

Transmitter	STX	P	I	ETX	LRC				
Receiver	STX	Data byte 수 H	Data byte 수 L	D0_H	D0_L	D1_H	D1_L	ETX	LRC



정상적으로 명령 수행 시 앞에 사진과 같이 출력됩니다.

위의 밑줄을 보면 POWER1만 ON된 것을 확인할 수 있습니다. 위와 같이 PE 명령을 통해 TETRA-DSV Power Sensor Board 모듈에 공급되는 전원을 한 번에 모두 제어할 수 있고, Pe 명령을 통해 TETRA-DSV Power Sensor Board 모듈에 공급되는 전원을 하나씩 제어할 수도 있습니다.

Chapter 5. 전송계층 프로토콜 TCP (Transmission Control Protocol)

5-1. TCP/IP 모듈 프로토콜요청 값 차트 (TCP/IP Module Protocol Request Chart)

수신 내용	Packet 시작	데이터개수	모드	명령어	파라미터	Packet 끝
Odometry 요청	DS	3	1	ODOM	1 (odom수신) 2 (선,각 속도수신)	XX
Robot Pose 요청	DS	2	1	AMCL	-	XX
Status 정보 요청	DS	2	1	STATUS	-	XX
저장된 Map List 요청	DS	2	1	MAPLIST	-	XX
저장된 위치 List 요청	DS	2	1	LOCLIST	-	XX
위치/상태정보 요청	DS	2	1	DATA	-	XX
충전스테이션 도킹	DS	4	2	DOC	MarkerID, Mode	XX
지도작성 모드	DS	2	2	SLAM	-	XX
지도저장	DS	3	2	MSV	Map_name	XX
지도작성/자율주행 모드 종료	DS	2	2	KILL	-	XX
자율주행모드	DS	3	2	NAV	Map_name	XX
좌표이동	DS	8	2	GO2	position.x position.y orientation.x orientation.y orientation.z orientation.w	XX
저장위치 이동	DS	3	2	GO1	location_name	XX
이동 취소	DS	2	2	GOCXL	-	XX
위치저장	DS	3	2	LSV	location_name	XX
최대속도 설정	DS	3	2	SPEED	Max Speed(m/sec)	XX
GPIO_Output 설정	DS	10	2	OUT	out_put0 out_put1 out_put2 out_put3 out_put4 out_put5 out_put6 out_put7	XX
GPIO_Sigle_Output 설정	DS	4	2	SOUT	out_put_id value	XX
GPIO 상태 확인	DS	2	1	GPIO	-	XX

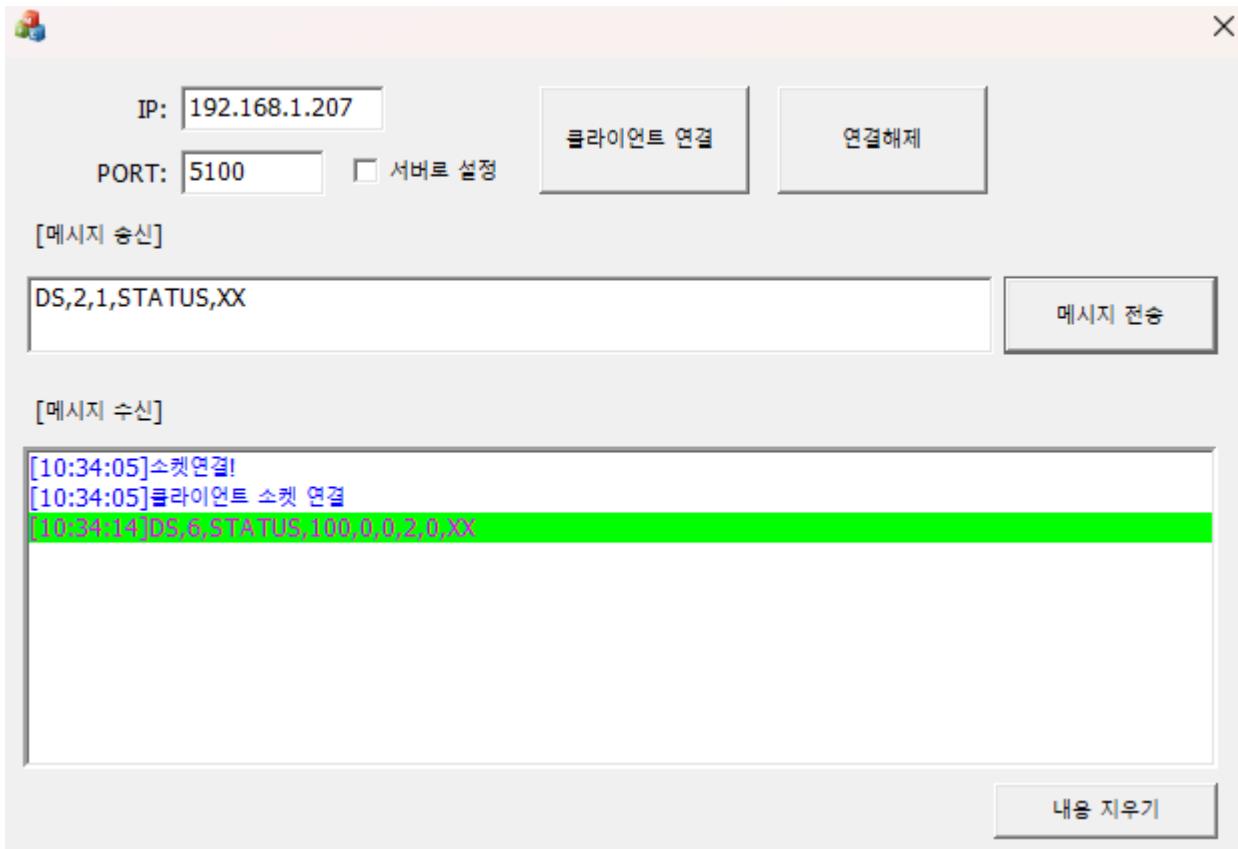
5-2. TCP/IP 모듈 프로토콜 응답 값 차트(TCP/IP Module Protocol Response Chart)

발신 내용	Packet 시작	데이터개수	명령어	Response Data	Packet 끝
Odometry 수신 (PARAM 이 0 인 경우)	DS	8	ODOM	position.x, position.y position.z orientation.x orientation.y orientation.z orientation.w	XX
Odometry 수신 (PARAM 이 1 인 경우)	DS	3	ODOM	linear.x angular.z	XX
Robot Pose 수신	DS	7	AMCL	position.x position.y orientation.x orientation.y orientation.z orientation.w	XX
Robot 상태정보 수신	DS	6	STATUS	Battery EMG Bumper Charging_status running_status	XX
저장된 Map List 요청 수신	DS	n + 1	MAPLIST	map_name[n] 저장된 map의 개수대로 배열개수가 정해짐	XX
저장된 위치 List 요청 수신	DS	n + 1	LOCLIST	location_name[n] 저장된 location의 개수 대로 배열개수가 정해짐	XX

위치/상태정보 수신	DS	12	DATA	position.x position.y orientation.x orientation.y orientation.z orientation.w Battery EMG Bumper Charging_status running_status	XX
자율주행모드 수신	DS	1	NAV	-	XX
좌표이동 수신	DS	1	GO2	-	XX
저장위치 이동 수신	DS	7	GO1	goal_positionX goal_positionY goal_quarterX goal_quarterY goal_quarterZ goal_quarterW	XX
이동 취소 수신	DS	1	GOCXL	-	XX
위치저장 수신	DS	2	LSV	Location name	XX
최대속도 설정	DS	2	SPEED	speed data	XX
GPIO_Output 설정	DS	1	OUT	-	XX
GPIO 상태 확인	DS	17	GPIO	in0 in1 in2 in3 in4 in5 in6 in7 out0 out1 out2 out3 out4 out5 out6 out7	XX

5-3. TCP/IP 모듈 프로토콜 실행 예제(TCP/IP Module Protocol Example)

Ex1) Robot 상태정보 수신 예시

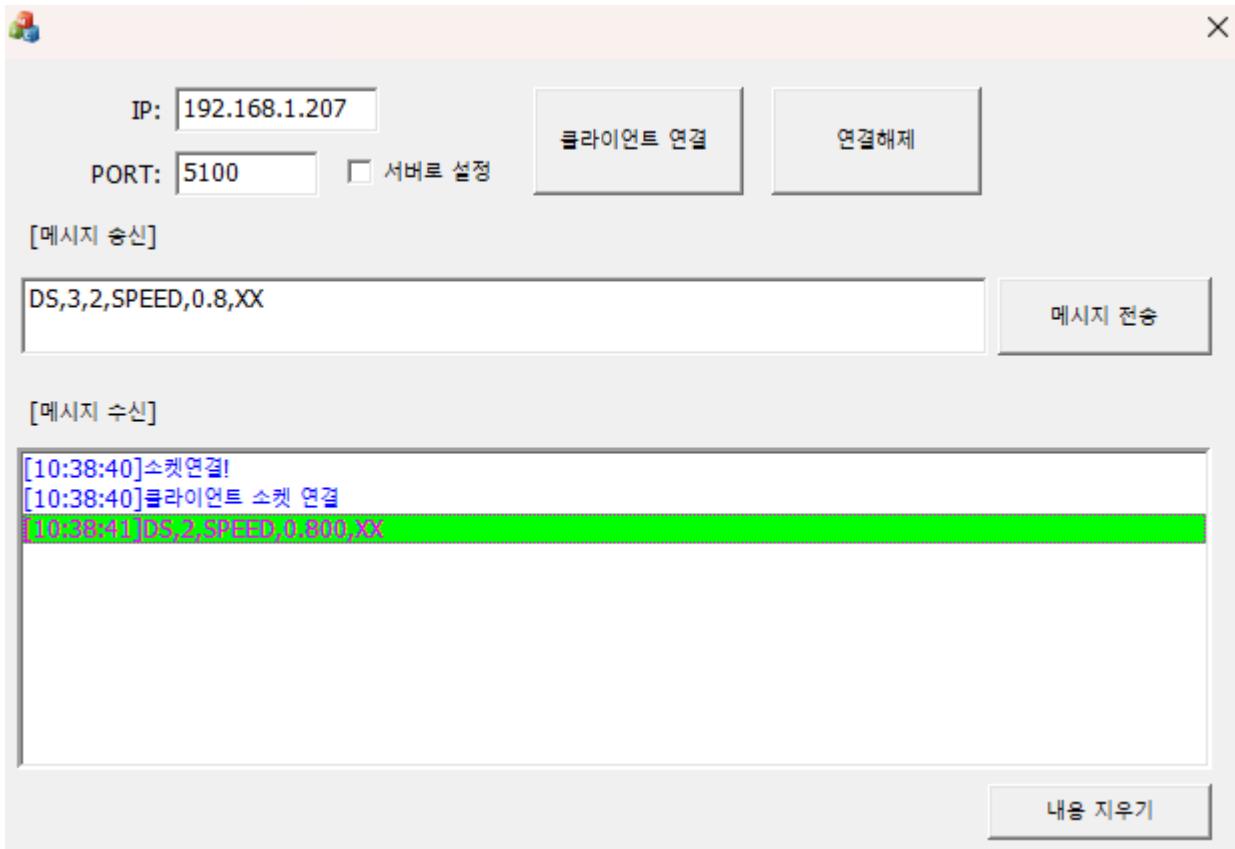


*명령어창에 [DS,2,1,STATUS,XX] 입력.

PLC -> TETRA	DS,2,1,STATUS,XX	Message size: 16
PLC->TETRA	DS,6,STATUS,100,0,0,2,0,XX	Message size: 25

- 100 : 배터리잔량이 100% 임을 표현
- 0 : EMG 버튼이 눌리지 않은 상태. (눌리면 1)
- 0 : 범퍼가 눌리지 않은 상태. (눌리면 1)
- 2 : 로봇의 단자의 상태가 충전스테이션에도 킹되어 있음
(주행 중에는 1, 컨베이어도 킹 단자에 붙으면 11 또는 12)
- 0 : 주행 동작 상태
(주행이 완료된 경우에만 3)

Ex2) Robot 최대속도설정



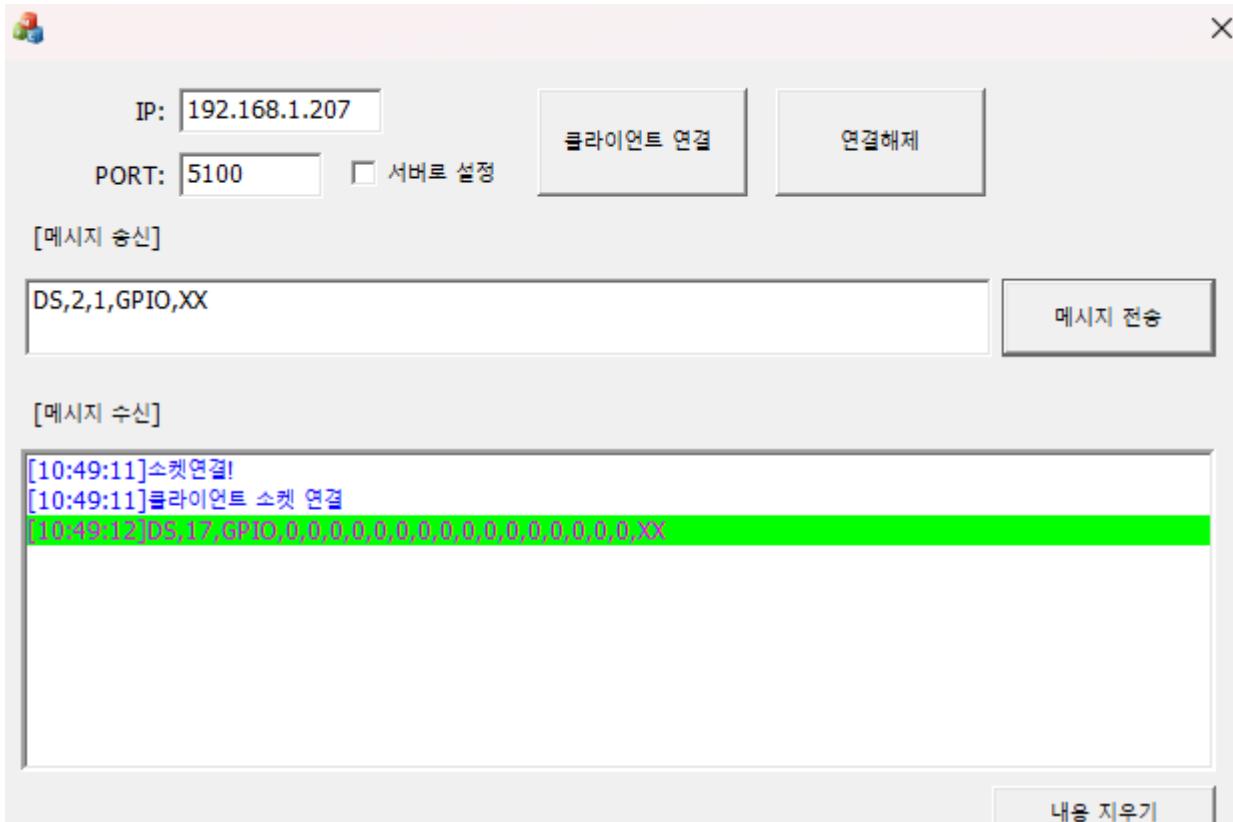
*명령어창에 [DS,3,2,SPEED,0.8,XX] 입력.

PLC -> TETRA	DS,3,2,SPEED,0.8,XX	Message size: 16
PLC->TETRA	DS,2,SPEED,0.800,XX	Message size: 25

0.800 : 0.8m/s의 속도를 로봇의 최대 속도로 설정

(항상 소수점 3자리까지 출력)

Ex3) GPIO상태확인



*명령어창에 [DS,2,1,GPIO,XX] 입력.

PLC -> TETRA	DS,2,1,GPIO,XX	Message size: 16
PLC->TETRA	DS,17,GPIO,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,XX	Message size: 25

입력0~7번까지모두OFF상태&출력0~7번까지모두OFF상태
(ON 상태일경우각각1로 출력)

```

uint8_t PENDING      = 0;      // The goal has yet to be processed by the action server
uint8_t ACTIVE       = 1;      // The goal is currently being processed by the action server
uint8_t PREEMPTED   = 2;      // The goal received a cancel request after it started executing
                             // and has since completed its execution (Terminal State)
uint8_t SUCCEEDED    = 3;      // The goal was achieved successfully by the action server (Terminal State)
uint8_t ABORTED     = 4;      // The goal was aborted during execution by the action server due
                             // to some failure (Terminal State)
uint8_t REJECTED     = 5;      // The goal was rejected by the action server without being processed,
                             // because the goal was unattainable or invalid (Terminal State)
uint8_t PREEMPTING   = 6;      // The goal received a cancel request after it started executing
                             // and has not yet completed execution
uint8_t RECALLING   = 7;      // The goal received a cancel request before it started executing,
                             // but the action server has not yet confirmed that the goal is canceled
uint8_t RECALLED    = 8;      // The goal received a cancel request before it started executing
                             // and was successfully cancelled (Terminal State)
uint8_t LOST         = 9;      // An action client can determine that a goal is LOST. This should not be
                             // sent over the wire by an action server
...

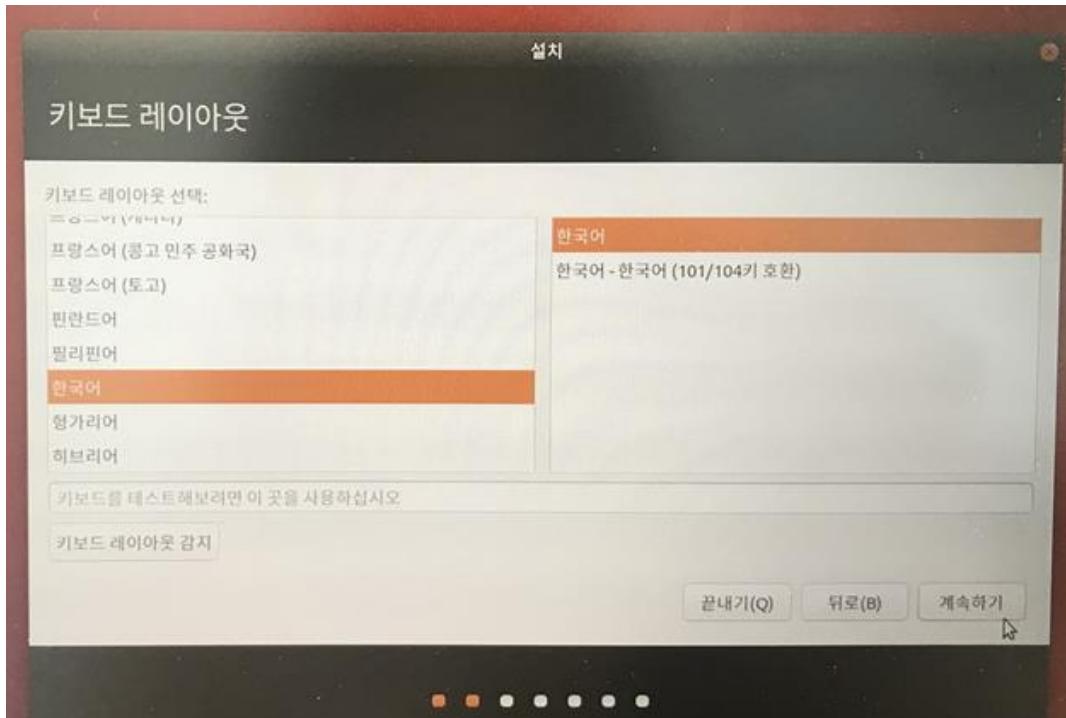
```

Chapter 6. Software Set up

6-0. PC Setup

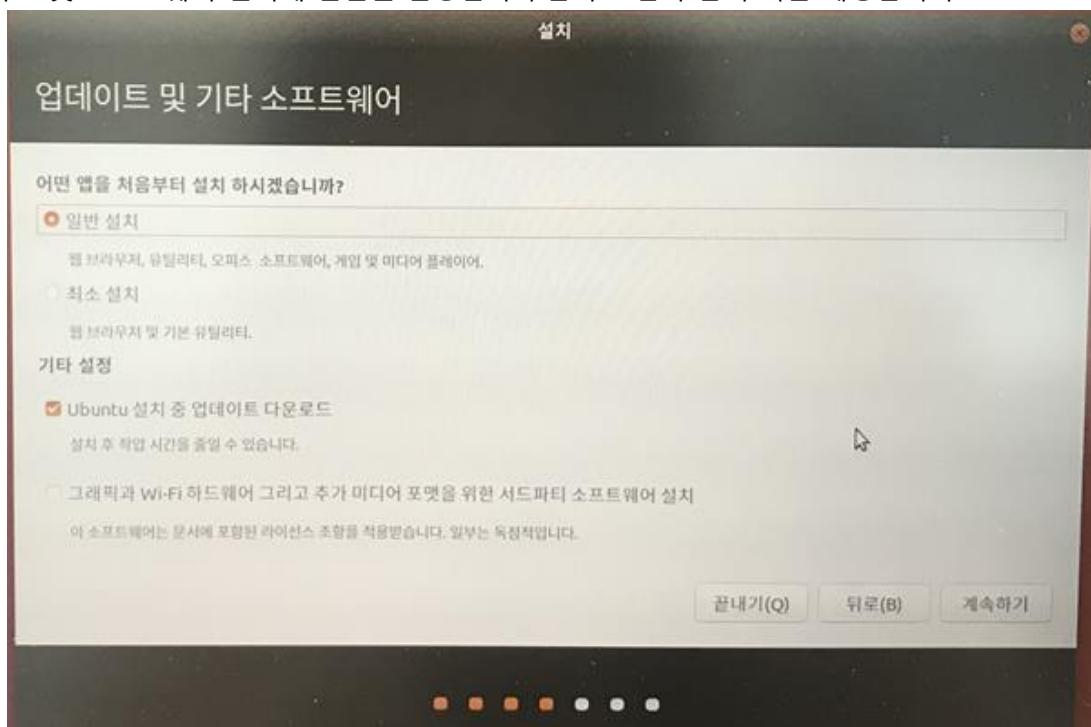
1)PC 설치

- USB 부팅 후 첫 화면이 뜨면 키보드 레이아웃 세팅을 요구합니다. 아래 그림과 같이 세팅합니다.



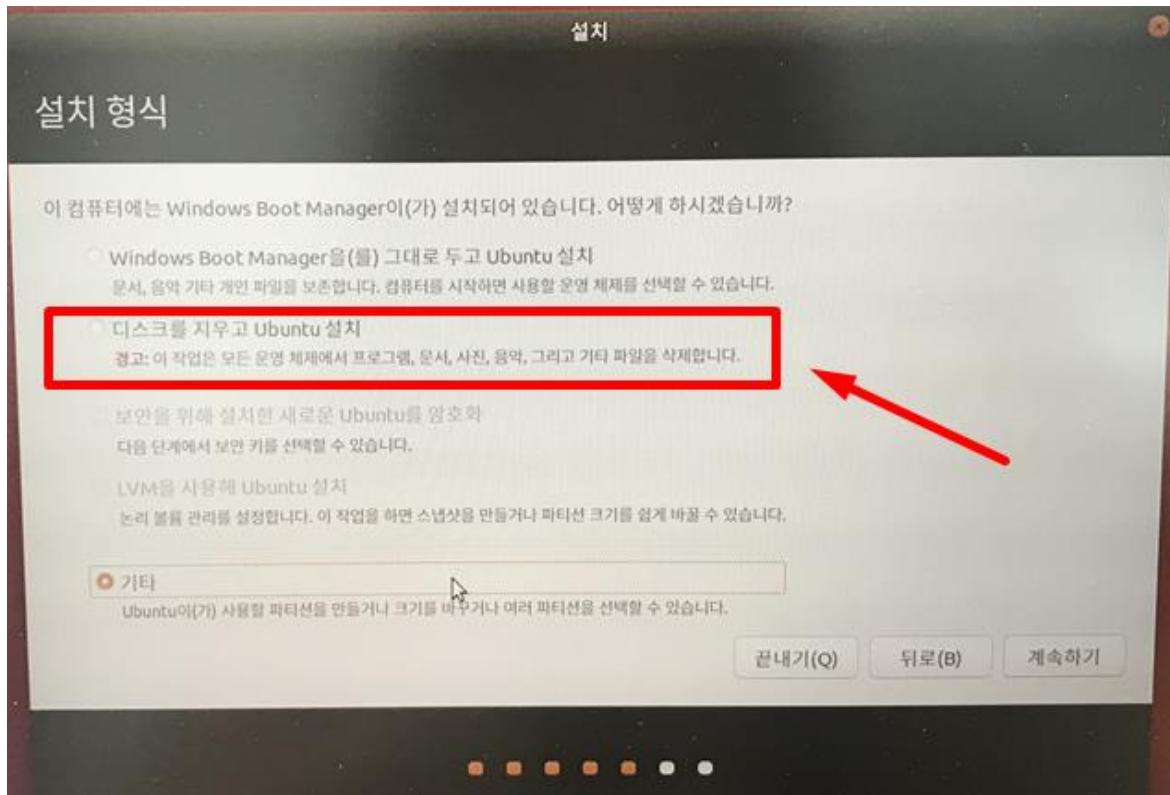
2)업데이트 및 기타 소프트웨어 설정

- 업데이트 및 소프트웨어 설치에 관련된 설정입니다. 밑의 그림과 같이 기본 세팅합니다.



3) 설치 형식 설정 및 Ubuntu 설치

- Ubuntu 18.04를 설치하기 전 마지막 단계입니다. 밑의 그림의 사각형을 보며 주의하여 디스크를 지우고 Ubuntu를 설치합니다. (여기에서의 디스크는 H/D를 말하므로 USB의 디스크는 지원하지 않습니다.)
***PC 의사양과 환경에 따라 설치 시간은 차이가 있을 수 있으며 시간이 꽤 소요되니 참고하세요.**
Ubuntu가 설치되는 과정 중에는 절대로 PC를 끄거나 강제 종료하지 마십시오.



6-1. ROS Melodic 설치

1) source.list 설정

- package.ros.org로부터 소프트웨어를 받아들일 수 있도록 PC를 세팅 합니다.

```
task@task:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

2) keys 설정

```
task@task:~$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

* keyserver 와의 연결 문제가 발생한다면 위의 명령어에서 " 안의 내용을 hkp:pgp.mit.edu:80 혹은 hkp://keyserver.ubuntu.com:80 으로 바꿔보시길 바랍니다.

3) melodic install

- Debian 패키지 목록을 최신화 합니다.

```
task@task:~$ sudo apt update
```

```
$ sudo apt update
```

- Desktop-full 설치를 합니다.

```
task@task:~$ sudo apt install ros-melodic-desktop
```

```
$sudo apt install ros-melodic-desktop-full
```

4) 환경변수 설정

```
task@task:~$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
task@task:~$ source ~/.bashrc
```

```
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

```
$ source ~/.bashrc
```

5) 의존성 패키지 빌드 설치

```
task@task:~$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential
```

```
$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential
```

-의존성 패키지 빌드 설치 후 터미널 창에 roscore 를 실행하여 정상적으로 설치되었는지 확인

```
task@task:~$ roscore
... logging to /home/task/.ros/log/e07c8f10-df90-11eb-84fe-80ee73f1c046/roslaunch-tas
k-27304.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://task:45415/
ros_comm version 1.14.11

SUMMARY
=====

PARAMETERS
  * /rosdistro: melodic
  * /rosversion: 1.14.11

NODES

auto-starting new master
process[master]: started with pid [27315]
ROS_MASTER_URI=http://task:11311

setting /run_id to e07c8f10-df90-11eb-84fe-80ee73f1c046
process[rosout-1]: started with pid [27326]
started core service [/rosout]
```

6) rosdep 초기화

```
task@task:~$ sudo apt install python-rosdep
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-rosdep is already the newest version (0.21.0-1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
task@task:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

      rosdep update

task@task:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/task/.ros/rosdep/sources.cache
```

```
$sudo apt install python-rosdep
$sudorosdep init
$ rosdep update
```

6-2.RealSense SDK 설치 (SDK 2.42.0)

1) source code 다운

<https://github.com/IntelRealSense/librealsense/releases/tag/v2.42.0>사이트에 접속하여

' Source code(tar.gz) '를 다운받고 압축을 해제합니다.

압축 해제 후 압축을 푼 폴더로 이동 후 빌드를 하기 위한 폴더를 생성합니다.

```
task@task:~/librealsense-2.42.0/
task@task:~/librealsense-2.42.0$ mkdir build
task@task:~/librealsense-2.42.0$ cd build
task@task:~/librealsense-2.42.0/build$
```

```
$cd librealsense-2.42.0
$mkdir build
$cd build
```

2) Download lib

```
task@task:~/librealsense-2.42.0/build$ sudo apt-get update
Hit:1 http://packages.ros.org/ros/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://kr.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://kr.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://kr.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
task@task:~/librealsense-2.42.0/build$ sudo apt-get install python3 python3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
python3 set to manually installed.
python3-dev is already the newest version (3.6.7-1~18.04).
python3-dev set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
task@task:~/librealsense-2.42.0/build$ sudo apt install libgtk-3-dev libxcursor-dev libxinerama-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libxcursor-dev is already the newest version (1:1.1.15-1).
libxcursor-dev set to manually installed.
libxinerama-dev is already the newest version (2:1.1.3-1).
libxinerama-dev set to manually installed.
The following additional packages will be installed:
  libatk-bridge2.0-dev libatspi2.0-dev libdbus-1-dev libepoxy-dev libxkbcommon-dev
    libxtst-dev wayland-protocols x11proto-record-dev
Suggested packages:
  libgtk-3-doc
The following NEW packages will be installed:
  libatk-bridge2.0-dev libatspi2.0-dev libdbus-1-dev libepoxy-dev libgtk-3-dev
    libxkbcommon-dev libxtst-dev wayland-protocols x11proto-record-dev
0 upgraded, 9 newly installed, 0 to remove and 2 not upgraded.
Need to get 1,460 kB of archives.
After this operation, 15.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
$ sudo apt-get update  
$ sudo apt-get install python3 python3-dev  
$ sudo apt install libgtk-3-dev libxcursor-dev libxinerama-dev
```

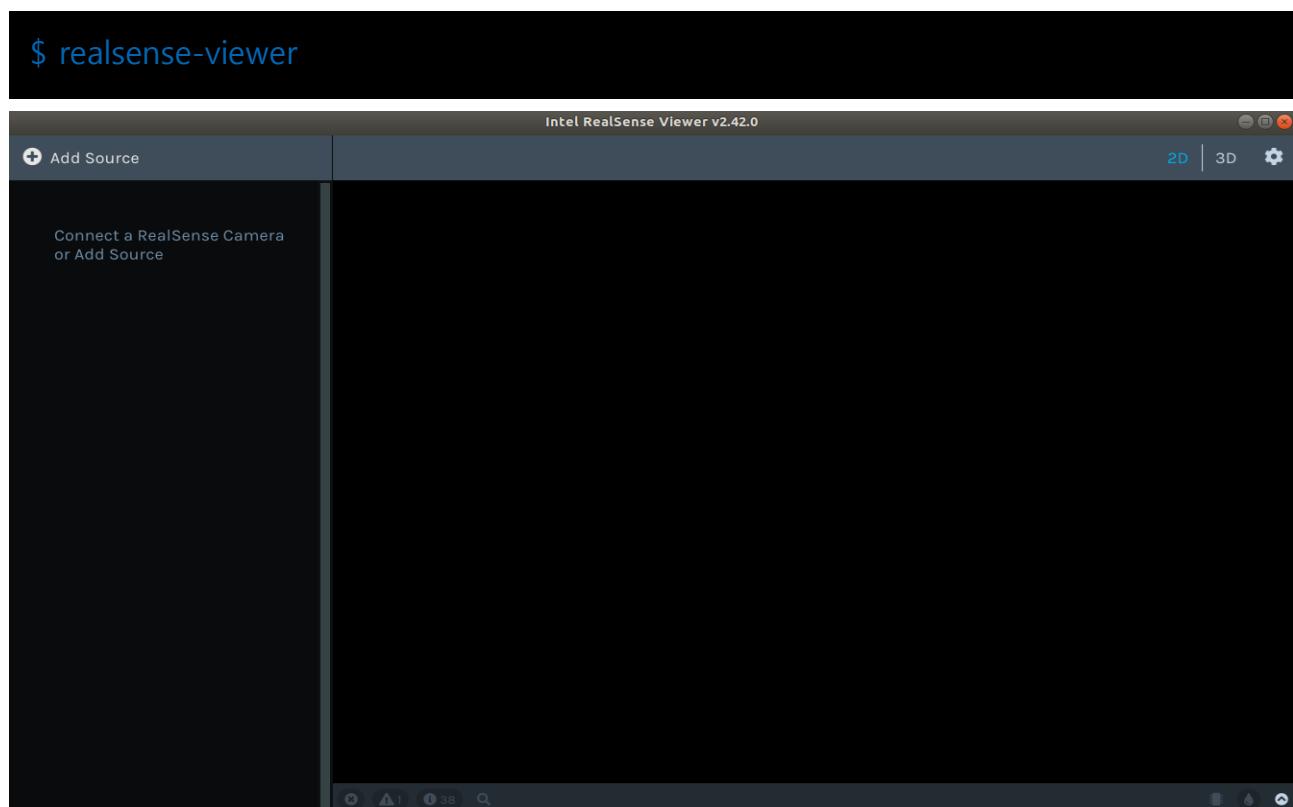
3) Make and Build

```
$ cmake ..  
$make -j4  
$sudo make install  
$ export PYTHONPATH=$PYTHONPATH:/usr/local/lib
```

4) launch

-빌드 완료 후 realsense-viewer 를 실행하여 정상적으로 설치 되었는지 확인합니다.

```
task@task:~/librealsense-2.42.0/build$ realsense-viewer
```



5) rule 설정

```
task@task:~$ cd librealsense-2.42.0/  
task@task:~/librealsense-2.42.0$ cd config/  
task@task:~/librealsense-2.42.0/config$ sudo cp 99-realsense-libusb.rules /etc/udev/rules.d/
```

```
$ cd librealsense-2.42.0  
$ cd config  
$sudo cp 99-realsense-libusb.rules /etc/udev/rules.d  
$sudo service udev restart
```

6-3.Realsense ROS 설치 (v 2.2.22)

1) realsense2 camera 설치

```
task@task:~$ export ROS_VER=melodic  
task@task:~$ sudo apt-get install ros-$ROS_VER-realsense2-camera  
[sudo] password for task:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  ros-melodic-ddynamic-reconfigure ros-melodic-librealsense2  
The following NEW packages will be installed:  
  ros-melodic-ddynamic-reconfigure ros-melodic-librealsense2  
  ros-melodic-realsense2-camera  
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.  
Need to get 13.1 MB of archives.  
After this operation, 36.5 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

```
$export ROS_VER=melodic  
$sudo apt-get install ros-$ROS_VER-realsense2-camera
```

2) realsense ros download

catkin_ws 폴더가 없으면 mkdir catkin_ws 명령어로 폴더를 생성합니다.

```
$mkdir catkin_ws  
$cd catkin_ws  
$mkdir src  
$git clone https://github.com/IntelRealSense/realsense-ros.git  
$cd realsense-ros/  
$git checkout `git tag | sort -V | grep -P "^\d+\.\d+\.\d+" | tail -1`  
$cd ..
```

3) Make

```
task@task:~$ cd catkin_ws/
task@task:~/catkin_ws$ catkin_make
```

```
$cd catkin_ws
$ catkin_make
```

4) package 적용

```
$echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
$source ~/.bashrc
```

5) realsense_ros 확인

realsense ros 패키지가 설치되었는지 확인하기 위해 터미널창을 연 후 다음과 같이 입력합니다.

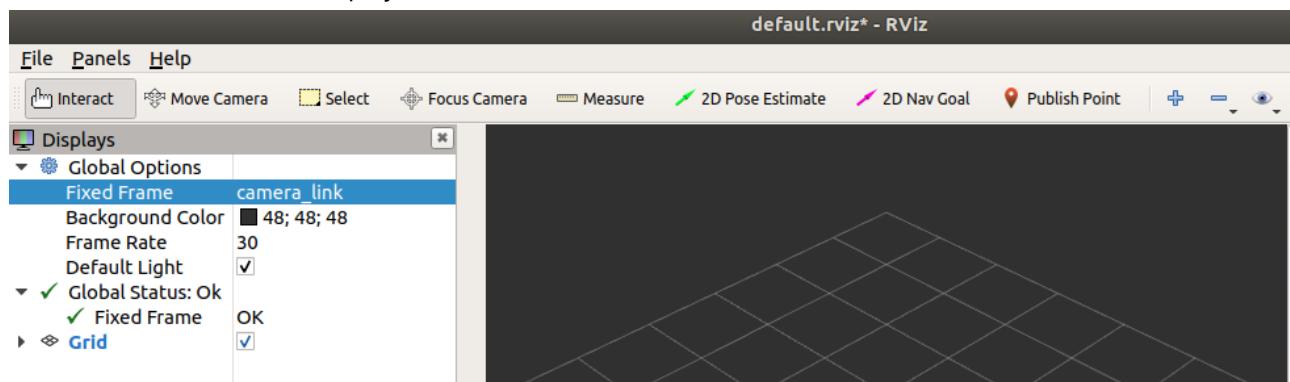
```
task@task:~$ roslaunch realsense2_camera rs_camera.launch
```

```
$ roslaunch realsense2_camera rs_camera.launch
```

터미널창을 추가로 실행한 후 rviz 를 입력합니다.

```
task@task:~$ rviz
[ INFO] [1625728372.216886193]: rviz version 1.13.17
[ INFO] [1625728372.216936272]: compiled against Qt version 5.9.5
[ INFO] [1625728372.216943572]: compiled against OGRE version 1.9.0 (Ghadamon)
[ INFO] [1625728372.220708926]: Forcing OpenGL version 0.
[ INFO] [1625728372.336498295]: Stereo is NOT SUPPORTED
[ INFO] [1625728372.336537627]: OpenGL device: Mesa DRI Intel(R) UHD Graphics 63
0 (CFL GT2)
[ INFO] [1625728372.336563607]: OpenGL version: 3.0 (GLSL 1.3).
```

rviz 프로그램이 실행되면 Displays 의 Fixed Frame 을 camera_link 로 선택합니다.



Add 를 클릭한 후 By topic 의 Image 를 선택한 후 OK 를 클릭합니다.

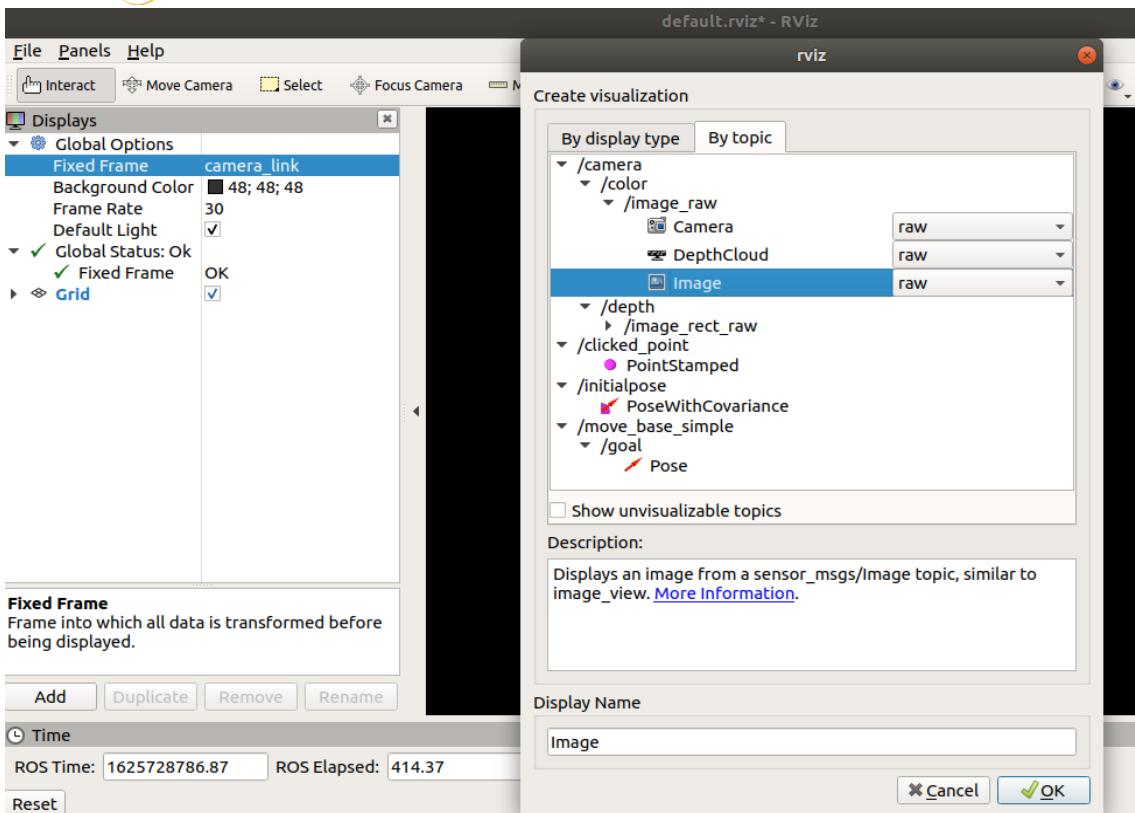
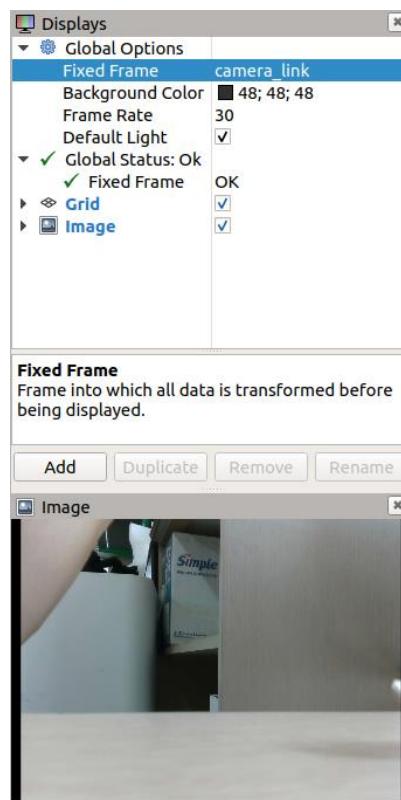


Image 화면이 정상으로 나오는지 확인합니다.



6-4.Cartographer 설치

1) ninja-build 패키지 설치

-패키지 목록 최신화 후 패키지 설치를 진행합니다.

```
task@task:~$ sudo apt-get install -y python-wstool python-rosdep ninja-build stow
$ sudo apt-get update
$ sudo apt-get install -y python-wstool python-rosdep ninja-build stow
```

2) cartographer 작업공간 생성

```
task@task:~$ mkdir cartographer_ws
task@task:~$ cd cartographer_ws/
task@task:~/cartographer_ws$ wstool init src
Writing /home/task/cartographer_ws/src/.rosinstall

update complete.
task@task:~/cartographer_ws$ wstool merge -t src https://raw.githubusercontent.com/cartographer-project/cartographer_ros/master/cartographer_ros.rosinstall
    Performing actions:

        Add new elements:
        cartographer,  cartographer_ros

Config changed, maybe you need run wstool update to update SCM entries.
Overwriting /home/task/cartographer_ws/src/.rosinstall

update complete.
task@task:~/cartographer_ws$ wstool update -t src
[cartographer] Fetching https://github.com/cartographer-project/cartographer.git (version master) to /home/task/cartographer_ws/src/cartographer
Cloning into '/home/task/cartographer_ws/src/cartographer'...
remote: Enumerating objects: 14658, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 14658 (delta 1), reused 8 (delta 0), pack-reused 14646
Receiving objects: 100% (14658/14658), 5.78 MiB | 6.68 MiB/s, done.
Resolving deltas: 100% (11691/11691), done.
[cartographer] Done.
[cartographer_ros] Fetching https://github.com/cartographer-project/cartographer_ros.git (version master) to /home/task/cartographer_ws/src/cartographer_ros
Cloning into '/home/task/cartographer_ws/src/cartographer_ros'...
remote: Enumerating objects: 4490, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 4490 (delta 2), reused 6 (delta 2), pack-reused 4480
Receiving objects: 100% (4490/4490), 4.28 MiB | 6.01 MiB/s, done.
Resolving deltas: 100% (3169/3169), done.
[cartographer_ros] Done.

$ mkdir cartographer_ws
$ cd cartographer_ws
$ wstool init src
$ wstool merge -t src https://raw.githubusercontent.com/cartographer-project/cartographer_ros/master/cartographer_ros.rosinstall
$ wstool update -t src
```

3) cartographer ros 종속성 설치

```
task@task:~/cartographer_ws$ sudo rosdep init
ERROR: default sources list file already exists:
        /etc/ros/rosdep/sources.list.d/20-default.list
Please delete if you wish to re-initialize
task@task:~/cartographer_ws$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/task/.ros/rosdep/sources.cache
task@task:~/cartographer_ws$ rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y

$ sudo rosdep init
$ rosdep update
$ rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y
```

4) abseil-cpp 라이브러리 설치

```
task@task:~/cartographer_ws$ src/cartographer/scripts/install_abseil.sh

$src/cartographer/scripts/install_abseil.sh
```

5) Build & install

```
task@task:~/cartographer_ws$ catkin_make_isolated --install --use-ninja

$catkin_make_isolated --install --use-ninja
```

6-5. ROS Package 설치

```
$ sudo apt-get install ros-melodic-move-base  
$ sudo apt-get install ros-melodic-teb-local-planner  
$ sudo apt-get install ros-melodic-serial  
$ sudo apt-get install ros-melodic-navigation  
$ sudo apt-get install ros-melodic-spatio-temporal-voxel-layer  
$ sudo apt-get install ros-melodic-ar-track-alvar  
$ sudo apt-get install ros-melodic-rgbd-launch
```

6-6.USBRule 설정

6-6-1 usb cam(AR marker) rule 설정

ll /dev 를 이용한 장치 확인

```
tetra@tetra:~$ ll /dev
```

```
$ll /dev
```

list에서 video 번호 확인(video 번호는 PC마다 틀릴 수 있음)

crw-rw----	1	root	tty	7,	1	7월 28 08:42	vcs1
crw-rw----	1	root	tty	7,	2	7월 28 08:42	vcs2
crw-rw----	1	root	tty	7,	3	7월 28 08:42	vcs3
crw-rw----	1	root	tty	7,	4	7월 28 08:42	vcs4
crw-rw----	1	root	tty	7,	5	7월 28 08:42	vcs5
crw-rw----	1	root	tty	7,	6	7월 28 08:42	vcs6
crw-rw----	1	root	tty	7,	128	7월 28 08:42	vcsa
crw-rw----	1	root	tty	7,	129	7월 28 08:42	vcsa1
crw-rw----	1	root	tty	7,	130	7월 28 08:42	vcsa2
crw-rw----	1	root	tty	7,	131	7월 28 08:42	vcsa3
crw-rw----	1	root	tty	7,	132	7월 28 08:42	vcsa4
crw-rw----	1	root	tty	7,	133	7월 28 08:42	vcsa5
crw-rw----	1	root	tty	7,	134	7월 28 08:42	vcsa6
crw-rw----	1	root	tty	7,	64	7월 28 08:42	vcsu
crw-rw----	1	root	tty	7,	65	7월 28 08:42	vcsu1
crw-rw----	1	root	tty	7,	66	7월 28 08:42	vcsu2
crw-rw----	1	root	tty	7,	67	7월 28 08:42	vcsu3
crw-rw----	1	root	tty	7,	68	7월 28 08:42	vcsu4
crw-rw----	1	root	tty	7,	69	7월 28 08:42	vcsu5
crw-rw----	1	root	tty	7,	70	7월 28 08:42	vcsu6
drwxr-xr-x	2	root	root		60	7월 28 08:42	vfio/
crw-----	1	root	root	10,	63	7월 28 08:42	vga_arbiter
crw-----	1	root	root	10,	137	7월 28 08:42	vhci
crw-----	1	root	root	10,	238	7월 28 08:42	vhost-net
crw-----	1	root	root	10,	241	7월 28 08:42	vhost-vsck
crw-rw----	1	root	video	81,	12	7월 28 08:42	video12
crw-rw----	1	root	video	81,	13	7월 28 08:42	video13
crw-rw-rw-	1	root	root	1,	5	7월 28 08:42	zero
crw-----	1	root	root	10,	249	7월 28 08:42	zfs

video 번호 확인 후 연결된 장치의 속성 확인을 위하여 아래와 같이 cmd 창에 입력

-list에서 확인 해야 할 부분은 ATTR{index}, ATTR{idProduct}, ATTR{idVendor} 부분을 확인하여 rules 파일에 등록해주어야 함

```
tetra@tetra:~$ udevadm info --attribute-walk /dev/video12
```

```
$ udevadm info --attribute-walk /dev/video12
```

```

looking at device '/devices/pci0000:00/0000:00:14.0/usb1/1-6/1-6:1.0/video4lin
ux/video12':
KERNEL=="video12"
SUBSYSTEM=="video4linux"
DRIVER==""
ATTR{dev_debug}=="0"
ATTR{index}=="0"
ATTR{name}=="USB 2.0 HD Camera : USB 2.0 HD"

looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-6/1-6:1.0':
KERNELS=="1-6:1.0"
SUBSYSTEMS=="usb"
DRIVERS=="uvcvideo"
ATTRS{authorized}=="1"
ATTRS{bAlternateSetting}==" 0"
ATTRS{bInterfaceClass}=="0e"
ATTRS{bInterfaceNumber}=="00"
ATTRS{bInterfaceProtocol}=="00"
ATTRS{bInterfaceSubClass}=="01"
ATTRS{bNumEndpoints}=="01"
ATTRS{iad_bFirstInterface}=="00"
ATTRS{iad_bFunctionClass}=="0e"
ATTRS{iad_bFunctionProtocol}=="00"
ATTRS{iad_bFunctionSubClass}=="03"
ATTRS{iad_bInterfaceCount}=="02"
ATTRS{interface}=="USB 2.0 HD Camera "
ATTRS{supports_autosuspend}=="1"

looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-6':
KERNELS=="1-6"
SUBSYSTEMS=="usb"
DRIVERS=="usb"
ATTRS{authorized}=="1"
ATTRS{avoid_reset_quirk}=="0"
ATTRS{bConfigurationValue}=="1"
ATTRS{bDeviceClass}=="ef"
ATTRS{bDeviceProtocol}=="01"
ATTRS{bDeviceSubClass}=="02"
ATTRS{bMaxPacketSize0}=="64"
ATTRS{bMaxPower}=="200mA"
ATTRS{bNumConfigurations}=="1"
ATTRS{bNumInterfaces}==" 2"
ATTRS{bcdDevice}=="1001"
ATTRS{bmAttributes}=="80"
ATTRS{busnum}=="1"
ATTRS{configuration}==""
ATTRS{devnum}=="4"
ATTRS{devpath}=="6"
ATTRS{idProduct}=="3822"
ATTRS{idVendor}=="058f"
ATTRS{ltm_capable}=="no"
ATTRS{maxchild}=="0"

```

udev rules 파일 경로로 이동 한 후 webcam.rules 파일을 생성

```
tetra@tetra:~$ cd /etc/udev/rules.d/
tetra@tetra:/etc/udev/rules.d$ sudo gedit webcam.rules
```

```
$cd /etc/udev/rules.d
$sudo gedit webcam.rules
```

아래와 같이 문구를 추가

```
Open ▾  webcam.rules  
/etc/udev/rules.d  
KERNEL=="video*", GROUP="video", ATTRS{idVendor}=="058f", ATTRS{idProduct}=="3822", ATTR{index}=="0", SYMLINK+="cam_rear"  
  
KERNEL=="video*", GROUP="video", ATTRS{idVendor}=="058f",  
ATTRS{idProduct}=="3822", ATTR{index}=="0", SYMLINK+="cam_rear"
```

-ATTR{index}, ATTR{idProduct}, ATTR{idVendor} 부분을 usb 정보에 맞게 입력 한 후 저장

6-6-2 IMU 설정

usb rule 설정과 마찬가지로 II /dev 를 이용하여 장치 확인

```
crw-rw---- 1 root dialout 4, 72 4월 26 08:21 ttyS8  
crw-rw---- 1 root dialout 4, 73 4월 26 08:21 ttyS9  
crw-rw-rw- 1 root dialout 188, 0 4월 26 17:15 ttyUSB0  
crw----- 1 root root 10, 60 4월 26 08:21 udmabuf  
crw----- 1 root root 10, 239 4월 26 08:21 uhid  
crw----- 1 root root 10, 223 4월 26 08:21 uinput
```

장치 속성 확인을 위하여 아래와 같이 cmd 창에 입력

```
tetra@tetra:~$ udevadm info --attribute-walk /dev/ttyUSB0
```

```
$ udevadm info --attribute-walk /dev/ttyUSB0
```

```
looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-8':  
KERNELS=="1-8"  
SUBSYSTEMS=="usb"  
DRIVERS=="usb"  
ATTRS{authorized}=="1"  
ATTRS{avoid_reset_quirk}=="0"  
ATTRS{bConfigurationValue}=="1"  
ATTRS{bDeviceClass}=="00"  
ATTRS{bDeviceProtocol}=="00"  
ATTRS{bDeviceSubClass}=="00"  
ATTRS{bMaxPacketSize0}=="64"  
ATTRS{bMaxPower}=="100mA"  
ATTRS{bNumConfigurations}=="1"  
ATTRS{bNumInterfaces}==" 1"  
ATTRS{bcdDevice}=="0100"  
ATTRS{bmAttributes}=="80"  
ATTRS{busnum}=="1"  
ATTRS{configuration}=="  
ATTRS{devnum}=="4"  
ATTRS{devpath}=="8"  
ATTRS{idProduct}=="ea60"  
ATTRS{idVendor}=="10c4"  
ATTRS{ltm_capable}=="no"  
ATTRS{manufacturer}=="Silicon Labs"  
ATTRS{maxchild}=="0"  
ATTRS{product}=="CP2102 USB to UART Bridge Controller"  
ATTRS{quirks}=="0x0"  
ATTRS{removable}=="removable"  
ATTRS{rx_lanes}=="1"  
ATTRS{serial}=="0001"  
ATTRS{speed}=="12"  
ATTRS{tx_lanes}=="1"  
ATTRS{urbnum}=="13151339"  
ATTRS{version}==" 1.10"
```

udev rules 파일 경로로 이동 한 후 IMU.rules 파일을 생성

```
tetra@tetra:~$ cd /etc/udev/rules.d/
tetra@tetra:/etc/udev/rules.d$ sudo gedit IMU.rules
```

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", ATTRS{serial}=="0001", MODE=="0666", GROUP=="dialout", SYMLINK+="IMU"
```

KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", ATTRS{serial}=="0001",
MODE=="0666", GROUP=="dialout", SYMLINK+="IMU"

-ATTR{serial}, ATTR{idProduct}, ATTR{idVendor} 부분을 usb 정보에 맞게 입력 한 후 저장

- ttyUSB 권한설정

```
tetra@tetra:~$ sudo usermod -a -G dialout $USER
```

```
$ sudo usermod -a -G dialout $USER
```

- 저장 후 터미널 창에 `sudo service udev restart` 를 한 후 PC를 재부팅 하면 설정이 완료 됨

※bashrc

마지막으로 ROS 환경 설정 파일을 자동으로 읽어오도록해주기 위하여 bashrc 파일 수정이 필요함

```
tetra@tetra:~$ gedit ~/.bashrc
```

```
$gedit ~/.bashrc
```

bashrc 파일이 열리면 맨마지막에 아래와같이 IP 와 path 를 확인하여 MASTER_URI, HOSTNAME IP 를 PC에 맞게 수정해야 함

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export ROS_NAMESPACE=TE211800
export ROS_MASTER_URI=http://192.168.20.90:11311
export ROS_HOSTNAME=192.168.20.90

source /opt/ros/melodic/setup.bash
source ~/cartographer_ws/install_isolated/setup.sh
source ~/catkin_ws-devel/setup.bash
```

저장 후 수정한 bashrc 파일을 적용

```
tetra@tetra:~$ source ~/.bashrc
```

```
$source ~/.bashrc
```

Chapter 7. SLAM

7-1. Node 실행

TETRA-DS 와관련된기본 Node 들을실행하기위하여 configuration launch 파일실행

```
tetra@tetra:~$ rosrun tetraDS_2dnav tetra_configuration.launch
```

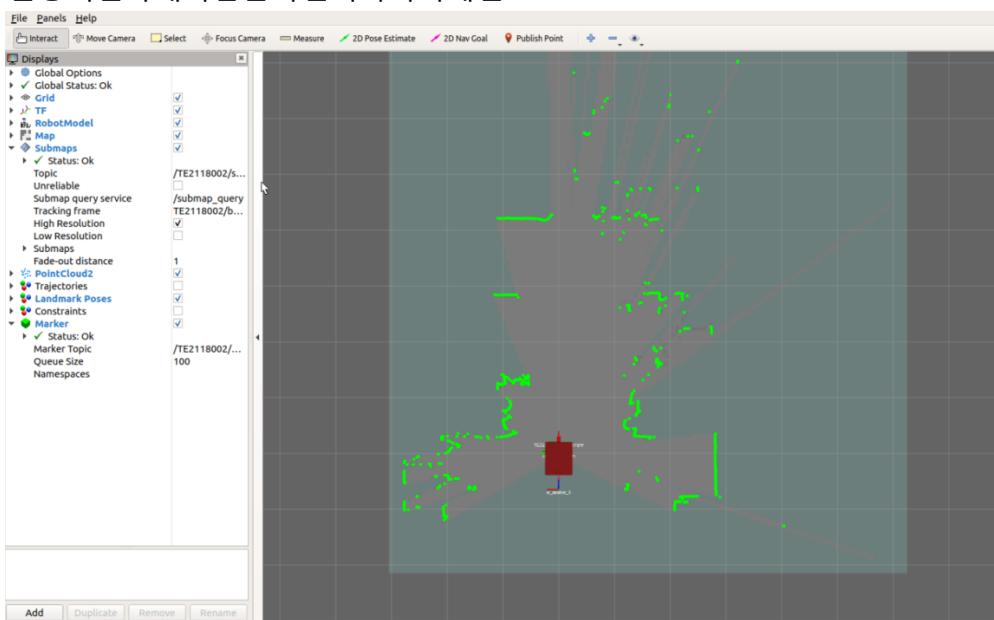
```
$ rosrun tetraDS_2dnav tetra_configuration.launch
```

cartographer 를이용하여환경지도를작성하기위한 mapping launch 파일실행

```
tetra@tetra:~$ rosrun tetraDS_2dnav cartographer_mapping.launch
```

```
$rosrun tetraDS_2dnav cartographer_mapping.launch
```

정상적으로실행되면아래와같은화면이나타나게됨



*지도작성시 로봇의 속도를 빠르게 설정하게 되면 환경에 따라 미끄러짐이 발생되어 위치오차 또는 맵이 틀어질 수 있기 때문에 mapping 시에는속도를천천히하여 mapping 을해야함
(권장속도 : 0.3m/s)

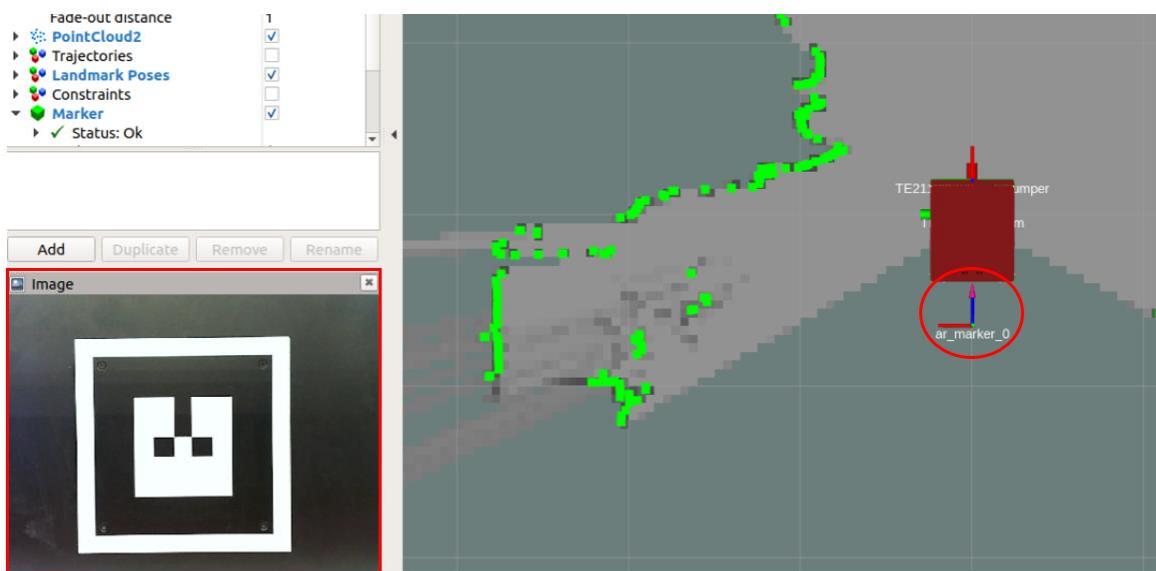
7-2. JoyStick 조작

조이스틱을 조작하여 주변환경에 대하여 mapping 시작할 수 있음

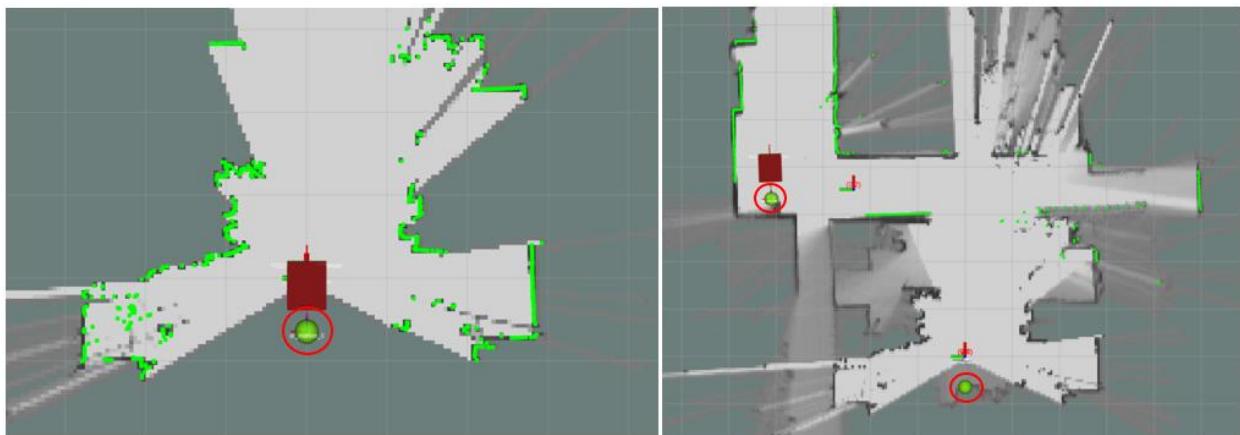


7-3.LandMark 등록

Navigation(자율주행)에서 충전스테이션, 컨베이어도킹시 AR Tag 를 통해 TETRA 의 위치보정을 할 수 있으며, 위치보정을 하기 위해 Mapping 시 Landmark 를 등록해 주어야 함. Mapping 중 AR Tag 를 인식하게 되면 아래와 같이 TF(ar_marker_ID) 가 생성되고, 조이스틱 RB 버튼을 누르면 Landmark 를 저장 할 수 있음



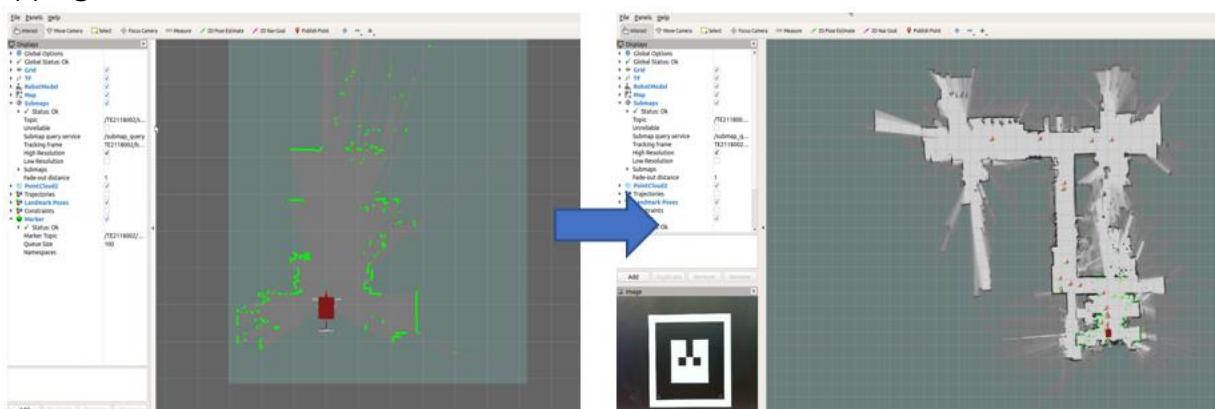
조이스틱 RB 버튼을 눌러 Landmark 를 저장하게 되면 RVIZ 화면에서 초록색 구모양으로 나타나게 됨



*등록한 Landmark list 는 /home/tetra/LANDMARK에 저장 됨

7-4.지도 저장

Mapping 완료후 환경지도를 저장하기 위해 service 호출

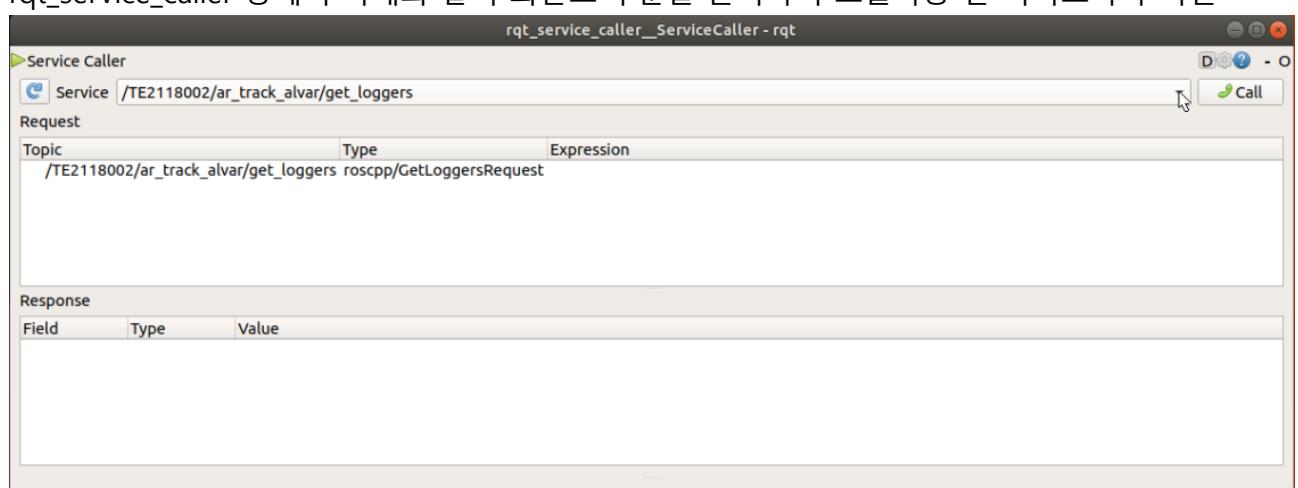


rqt service caller 실행

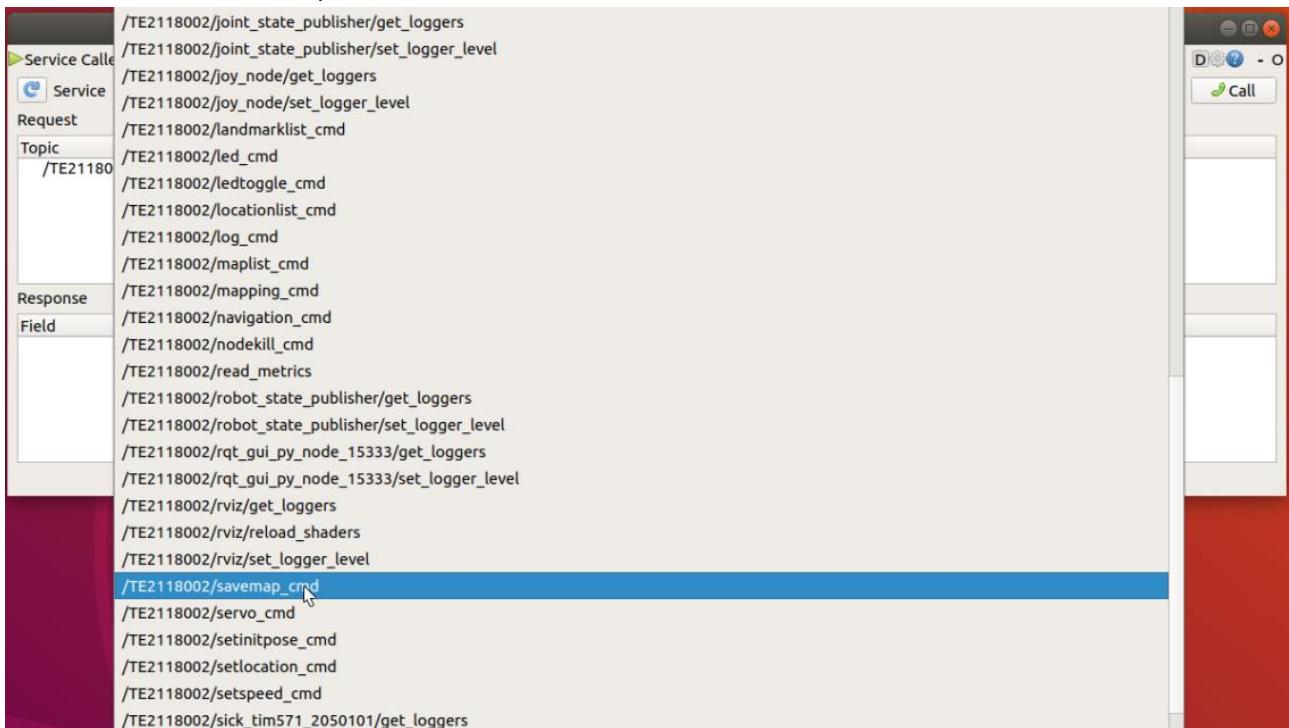
```
tetra@tetra:~$ rosrun rqt_service_caller rqt_service_caller
```

```
$ rosrun rqt_service_caller rqt_service_caller
```

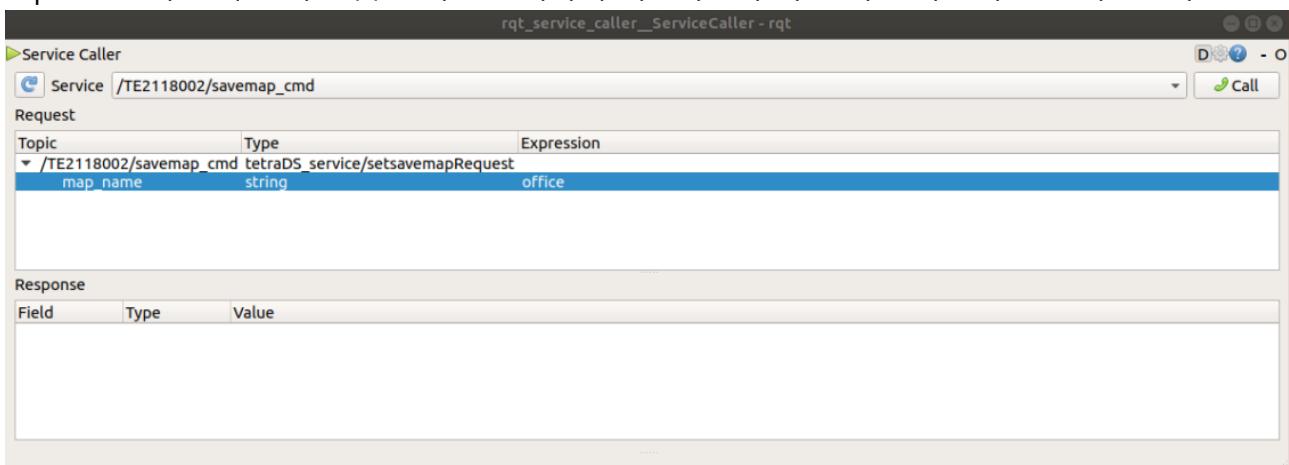
rqt_service_caller 창에서 아래와 같이 화살표 부분을 클릭하여 호출가능 한 서비스목록 확인



서비스목록에서 savemap_cmd 를선택



Expression 부분의 표시된 곳을 더블 클릭하여 저장하고자 하는 이름 작성 후 Call 버튼클릭



*지도가 저장되는 경로는 /home/tetra/catkin_ws/tetraDS_2dnav/maps에 저장됨

Chapter 8. Navigation

8-1. Node 실행

TETRA-DS 와관련된기본 Node 들을실행하기위하여 configuration launch 파일실행

```
tetra@tetra:~$ rosrun tetraDS_2dnav tetra_configuration.launch
```

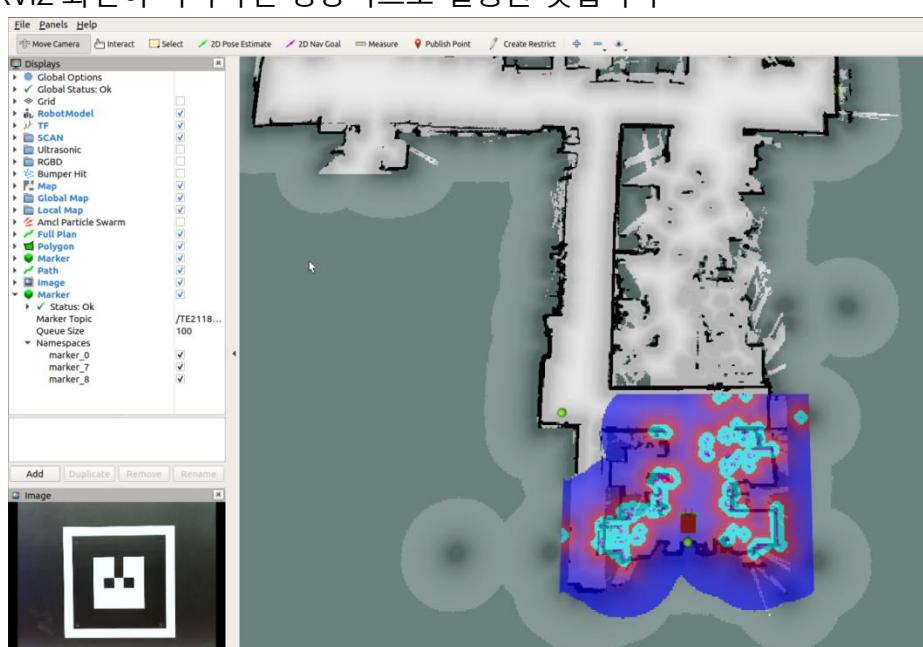
```
$ rosrun tetraDS_2dnav tetra_configuration.launch
```

TETRA-DS 를자율주행하기위한 launch 파일실행

```
tetra@tetra:~$ rosrun tetraDS_2dnav move_base_tetra.launch
```

```
$ rosrun tetraDS_2dnav move_base_tetra.launch
```

아래와 같은 RVIZ 화면이 나타나면 정상적으로 실행된 것입니다



주의사항

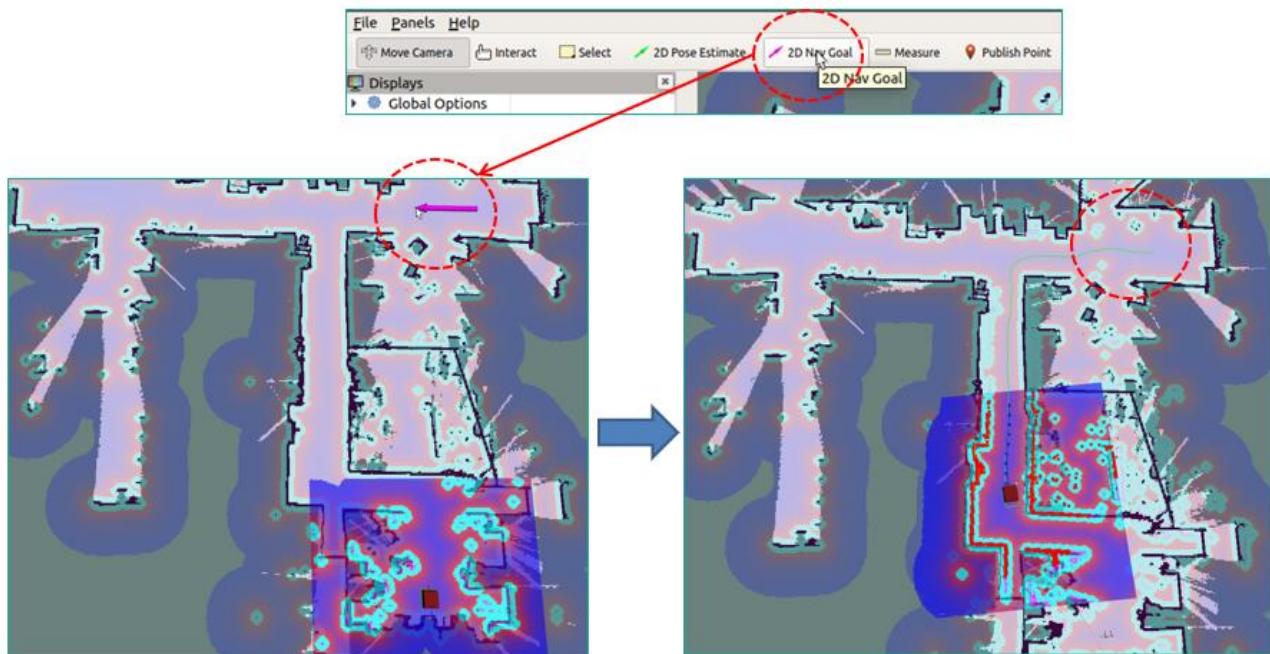
*TETRA-DS5는충전스테이션에도킹되어있는상태여야함

*실행순서는 **tetra_configuration launch** 파일이실행되고후에 **cartographer mapping launch** 또는 **move_base_tetra launch** 가실행되어야함

8-2. 위치 이동

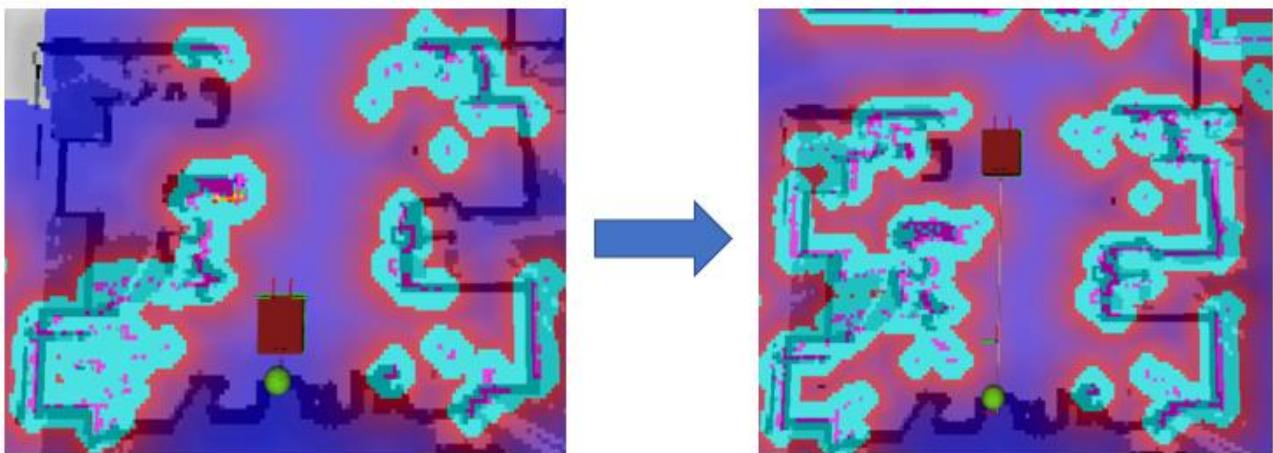
8-2-1 2D Nav Goal

RVIZ 화면에서 상단의 2D Nav Goal 을 클릭 후 원하는 위치에 마우스로 드래그하면 해당 위치로 TETRA 가이동하게 됩니다.



8-2-2 Joystick 사용

조이스틱을 움직여 TETRA 를 해당 위치로 이동시킬 수 있습니다.



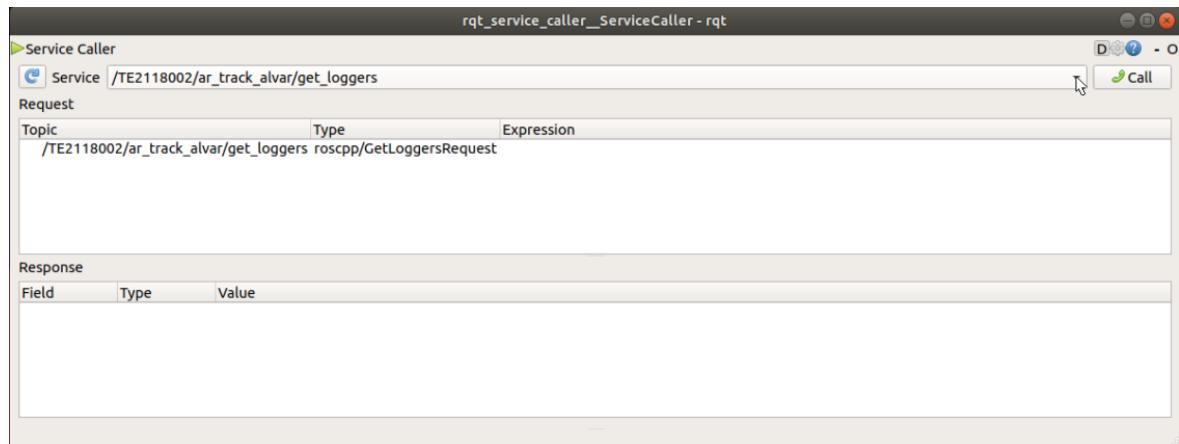
8-2-3서비스 호출

rqt service caller 실행

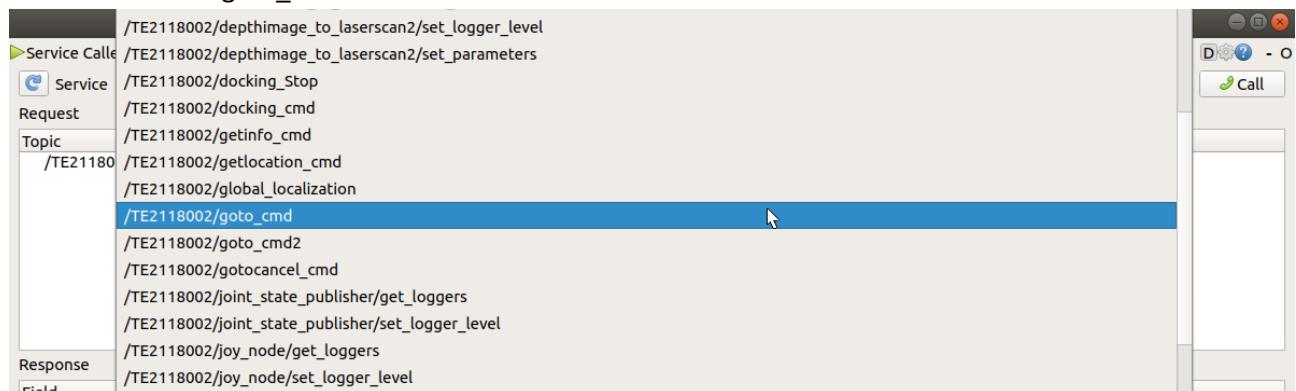
```
tetra@tetra:~$ rosrun rqt_service_caller rqt_service_caller
```

```
$ rosrun rqt_service_caller rqt_service_caller
```

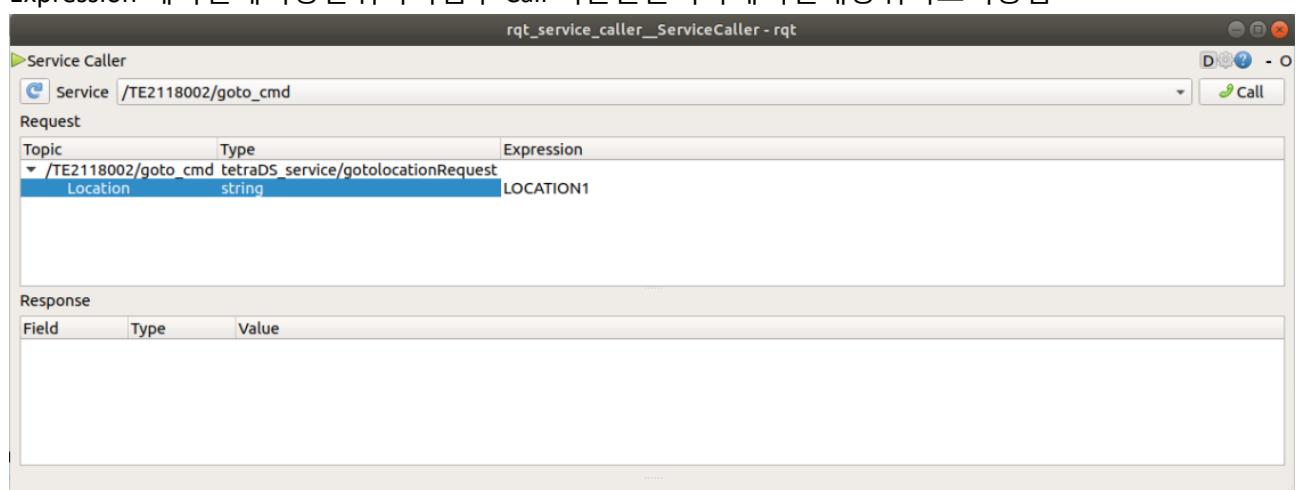
화살표를 눌러서 서비스 목록 확인



서비스 목록에서 goto_cmd 클릭



Expression에 사전에 저장한 위치 기입 후 Call 버튼을 클릭하게 되면 해당 위치로 이동함



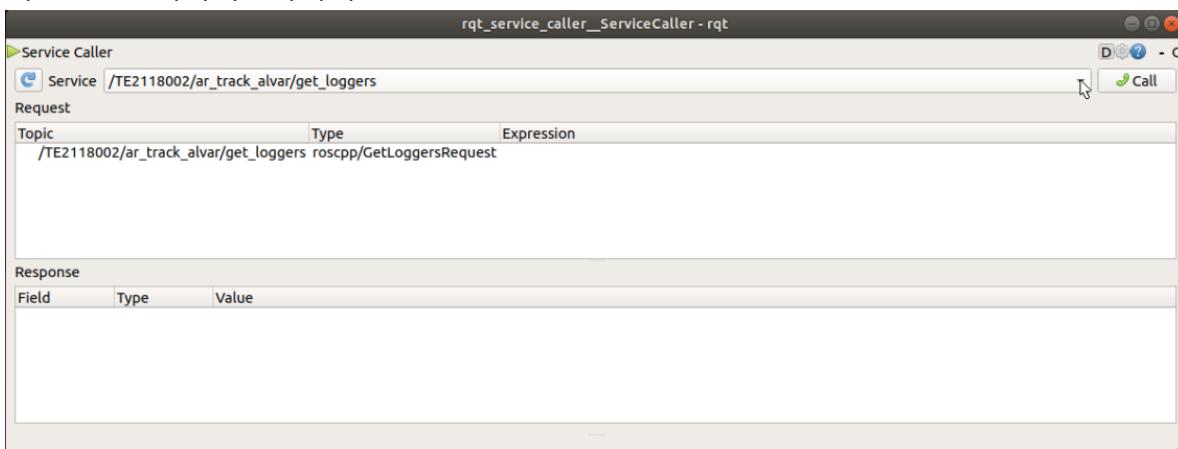
8-3. 위치 저장

위치이동후 해당위치를 저장하고자 할때 setlocation_cmd 서비스를 호출하여 저장할수 있습니다.

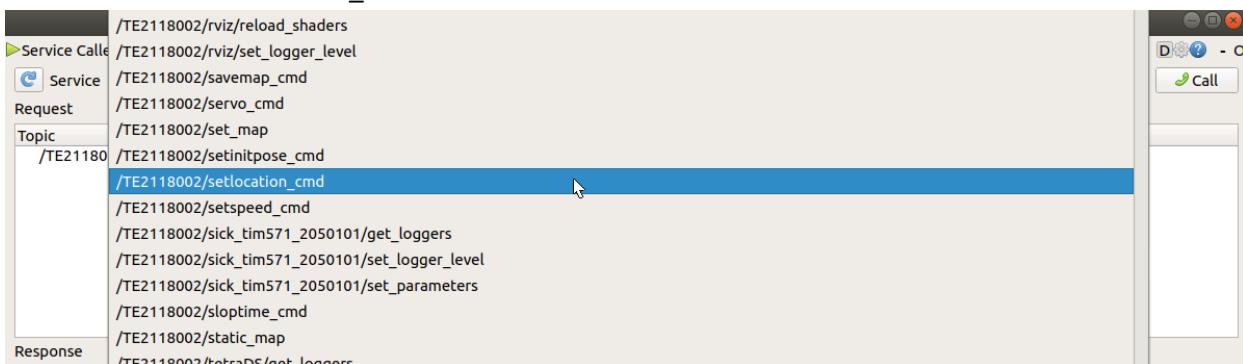
rqt service caller 실행

```
tetra@tetra:~$ rosrun rqt_service_caller rqt_service_caller
$ rosrun rqt_service_caller rqt_service_caller
```

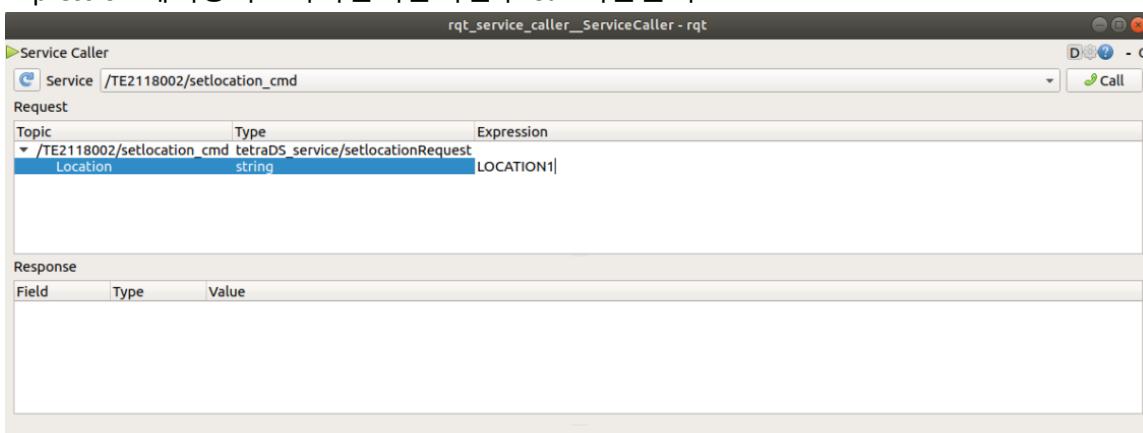
화살표를 눌러서 서비스 목록 확인



서비스 목록에서 setlocation_cmd 선택

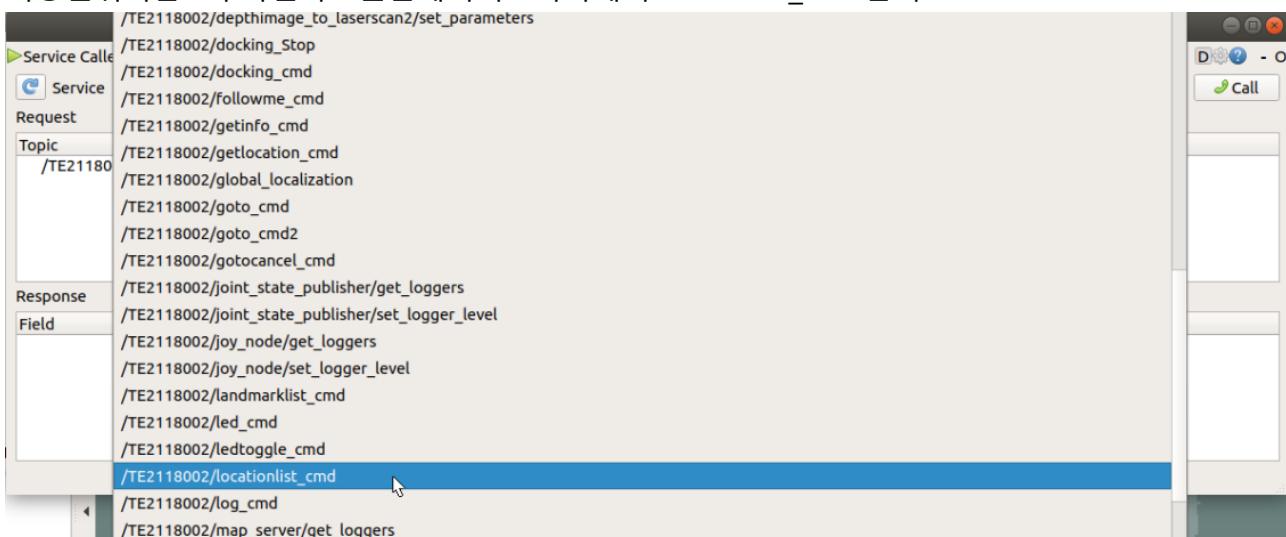


Expression에 저장하고자 하는 이름기입후 Call 버튼 클릭

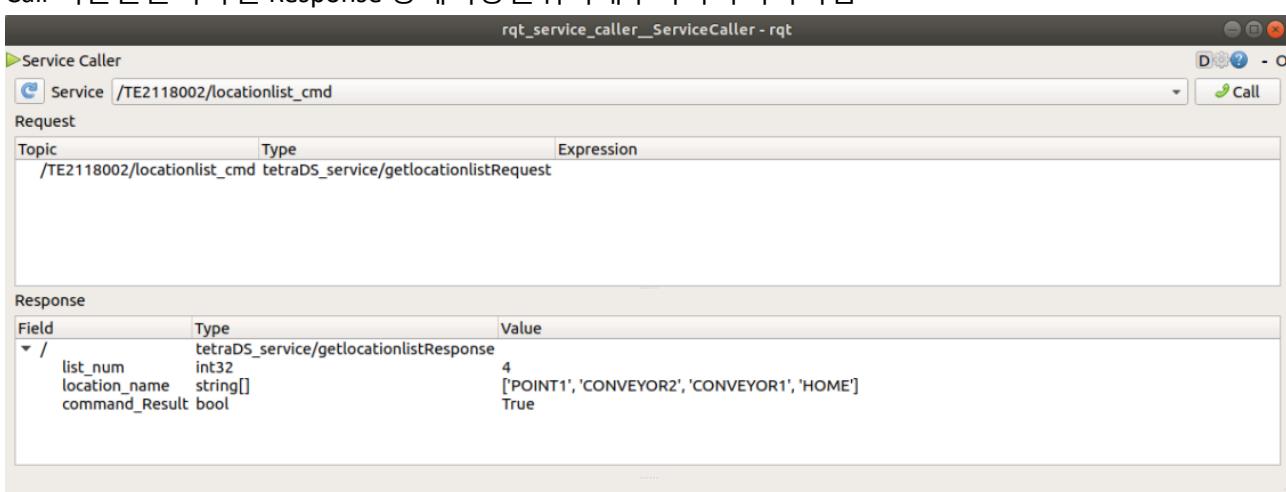


* 위치파일들은 /home/tetra/DATA에 저장됩니다

저장된 위치를 모두 확인하고 싶을 때 서비스 목록에서 locationlist_cmd 선택



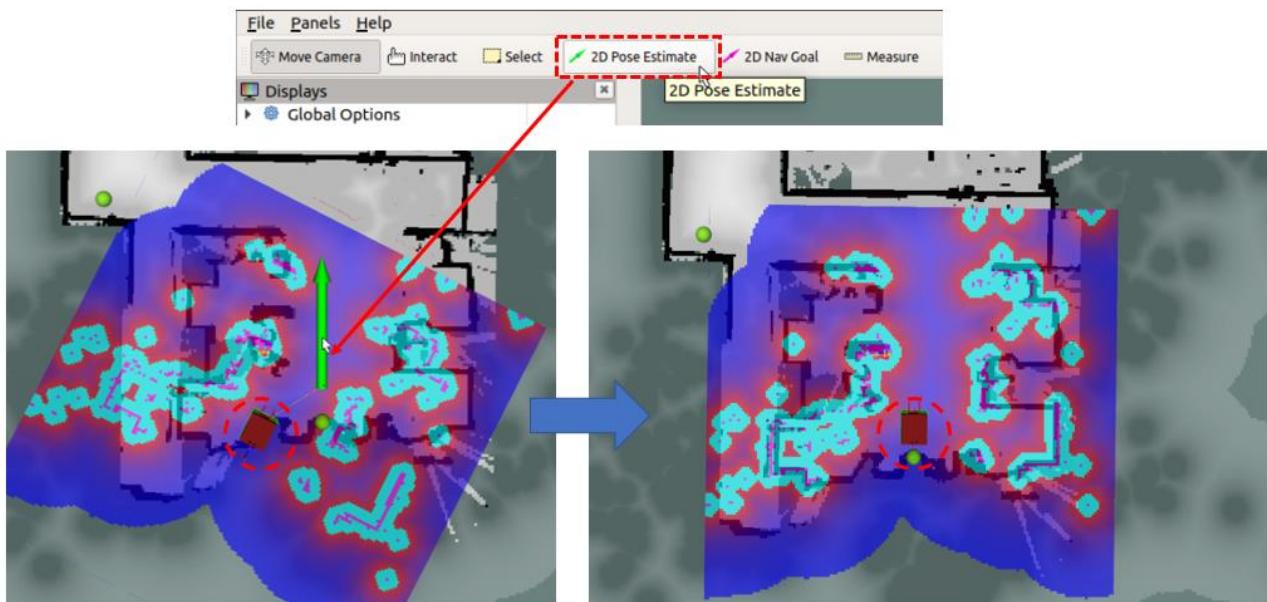
Call 버튼을 클릭하면 Response 창에 저장된 위치 개수와 목록이 나타남



8-4. 위치 보정

8-4-1 2D Pose Estimate

위치를 수동으로 보정하고 싶은 경우에는 상단에 위치한 '2D Pose Estimate' 버튼을 클릭한 후 map 상에서 현재 로봇 위치에 좌클릭 후로봇 방향으로 녹색화살표를 정하게 되면 해당 위치로 이동 후 보정하게 됩니다.

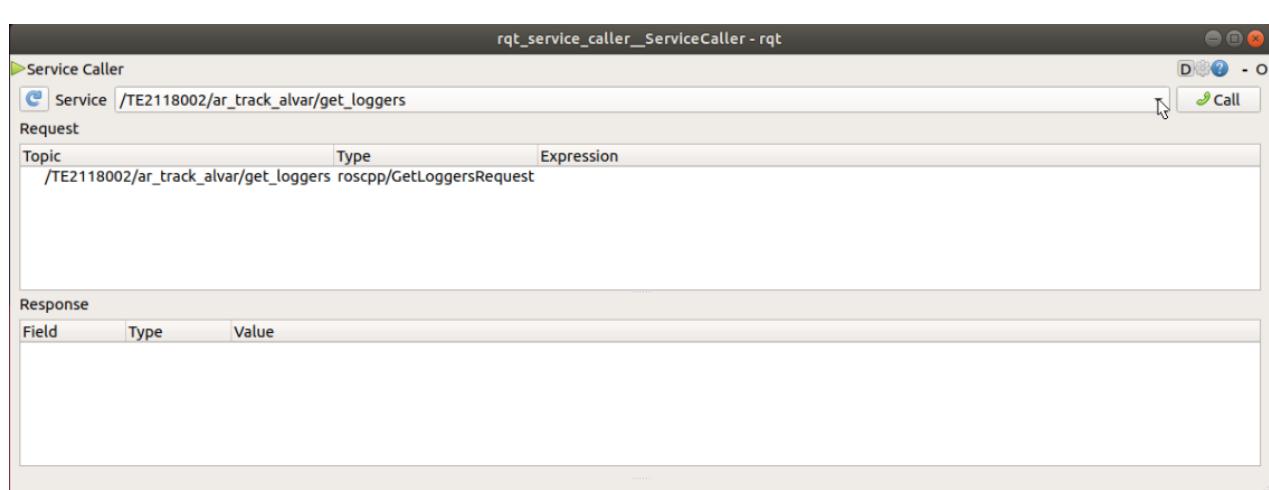


8-4-2 서비스 호출

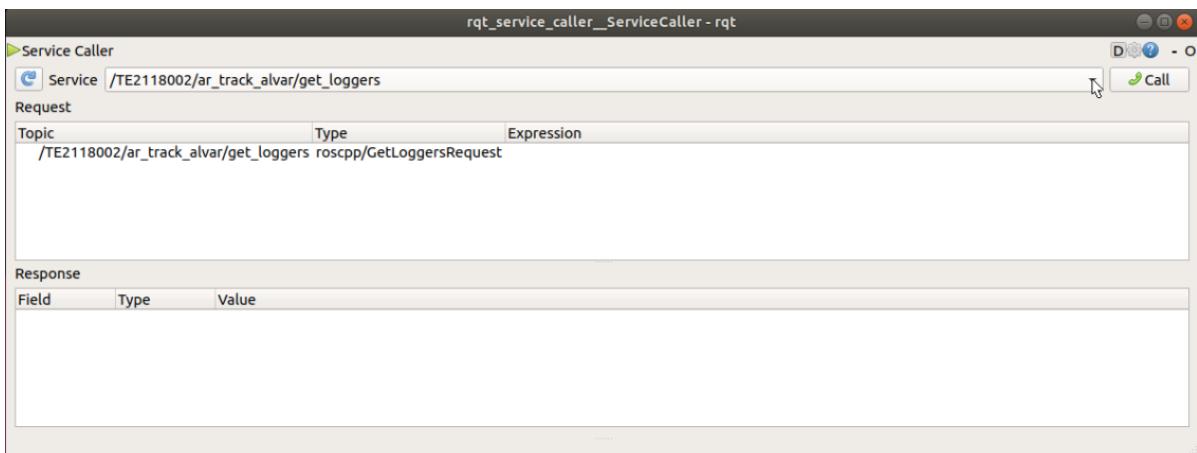
서비스 호출로 위치를 보정하고자 할 때에는 rqt service caller 실행한 후

```
tetra@tetra:~$ rosrun rqt_service_caller rqt_service_caller
```

```
$rosrun rqt_service_caller rqt_service_caller
```



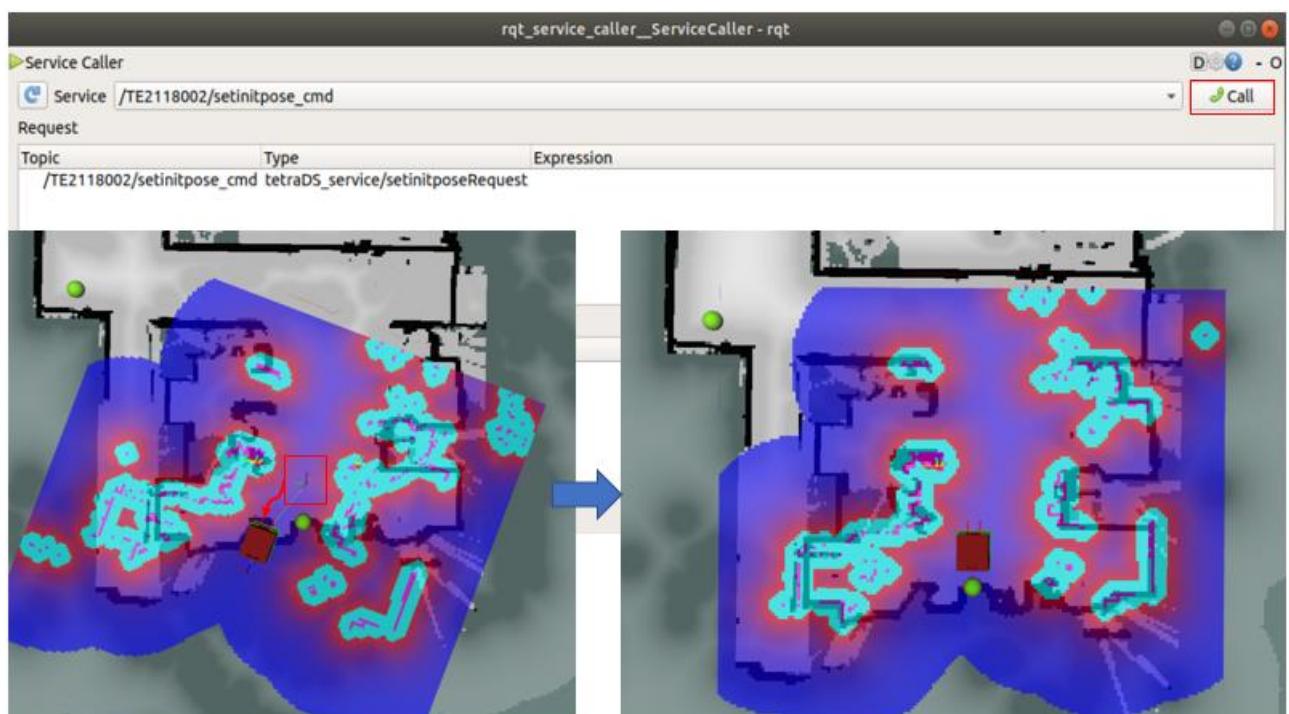
화살표를 눌러서 서비스 목록 확인



setinitpose_cmd 선택



Call 버튼을 클릭하면 위치보정을 실시함



Chapter 9. ROS

9-1. TETRA-DSV ROS Service

service name	Input	Output	Description
getlocation_cmd		<code>bool command_Result</code> <code>float32 poseAMCLx</code> <code>float32 poseAMCLy</code> <code>float32 poseAMCLqx</code> <code>float32 poseAMCLqy</code> <code>float32 poseAMCLqz</code> <code>float32 poseAMCLqw</code>	<p>로봇의 현재 위치정보데이터 확인명령.</p> <p>* 리턴으로는 X,Y좌표(m단위)와 쿼터니언(4개)가 수신.</p>
goto_cmd	<code>string</code> location	<code>bool command_Result</code> <code>float32 goal_positionX</code> <code>float32 goal_positionY</code> <code>float32 goal_quarterX</code> <code>float32 goal_quarterY</code> <code>float32 goal_quarterZ</code> <code>float32 goal_quarterW</code>	<p>로봇의 저장위치 이동 명령</p> <p>* 입력으로 저장된 위치(문자열)이름을 전송</p> <p>* 리턴으로는 호출결과와 저장위치에 X,Y좌표, 쿼터니언(4개)가 수신</p>
goto_cmd2		<code>float32 goal_positionX</code> <code>float32 goal_positionY</code> <code>float32 goal_quarterX</code> <code>float32 goal_quarterY</code> <code>float32 goal_quarterZ</code> <code>float32 goal_quarterW</code>	<code>bool command_Result</code> <p>로봇의 특정위치 이동 명령</p> <p>* 입력으로 X,Y좌표와 쿼터니언(4개)데이터를 전송</p> <p>* 리턴으로는 호출결과가 수신</p>
setlocation_cmd	<code>string</code> location	<code>bool command_Result</code> <code>float32 goal_positionX</code> <code>float32 goal_positionY</code> <code>float32 goal_quarterX</code> <code>float32 goal_quarterY</code> <code>float32 goal_quarterZ</code> <code>float32 goal_quarterW</code>	<p>로봇에 현재위치 저장 명령.</p> <p>* 입력으로는 현재위치를 저장할 위치이름(문자열) 전송.</p> <p>* 리턴으로는 호출결과와, 현재위치에 X,Y좌표, 쿼터니언(4개)가 수신.</p>
savemap_cmd	<code>string</code> map_name	<code>bool</code> command_Result	<p>작성된 현재 환경지도의 지도파일 저장 명령.</p> <p>* 입력으로는 저장할 지도의 이름(문자열)전송.</p> <p>* 리턴으로는 호출결과 수신.</p>
getinfo_cmd		<code>bool</code> command_Result <code>int32</code> battery <code>int32</code> Error_code <code>int32</code> EMG <code>int32</code> bumper <code>int32</code> charging <code>int32</code> running_mode	<p>로봇의 상태정보 확인 명령.</p> <p>* 리턴으로는 배터리 레벨, 에러코드, 비상정치스위치 체크, 범퍼체크, 충전상태 체크, 동작모드가 수신.</p> <ul style="list-style-type: none"> . 단, 에러코드는 여러 미발생 시 0을 리턴. . 동작모드(running_mode) 설명: 0: 기본상태 1 : 환경지도 작성 모드, 2: 자율주행 모드
locationlist_cmd		<code>int32</code> list_num <code>string[]</code> location_name <code>bool</code> command_Result	<p>로봇에 저장된 Waypoint정보 확인명령</p> <p>* 리턴으로는 Waypoint의 총 개수와 각각의 Waypoint 이름(문자열) 배열 수신</p>
delete_location_cmd	<code>string</code> location	<code>bool</code> command_Result	<p>로봇의 저장된 Waypoint삭제</p> <p>* 입력으로 Waypoint이름(문자열)을 전송</p> <p>* 리턴으로는 삭제여부 결과가 수신</p>
maplist_cmd	-	<code>int32</code> list_num <code>string[]</code> location_name <code>bool</code> command_Result	<p>'/home/tetra/catkin_ws/src/tetraDS_2dnav/maps 경로에 저장된 map정보확인 명령</p> <p>* 리턴으로는 map의 총 개수와 각각의 map이름(문자</p>

			<p>열) 배열 수신 * .pgm 확장자만 검색</p>
mapping_cmd	bool flag_mapping	bool command_Result	<p>cartographer를 이용한 지도작성 기능을 호출하는 명령 * 입력으로는 true를 전송 * 리턴으로는 호출결과 수신 * 주의사항: 자율주행 기능이 호출중이면, nodekill을 먼저 수행 해야함</p>
navigation_cmd	string map_name	bool command_Result	<p>로봇의 자율주행 기능을 호출하는 명령 * 입력으로는 자율주행을 실시할 map의 이름을 전송. * 리턴으로는 호출결과 수신. * 주의사항: 지도작성 기능이 호출중이면, nodekill을 먼저 수행 해야 함</p>
nodekill_cmd	-	bool command_Result	<p>tetra_configuration.launch를 제외하고 로봇에 실행중인 ROS node들을 모두 종료하는 명령. * 입력으로는 true를 전송. * 리턴으로는 호출결과 수신.</p>
delete_map_cmd	string map_name	bool command_Result	<p>저장된 환경지도를 지우는 명령. * 입력으로는 환경지도 이름을 전송. * 리턴으로는 호출결과 수신.</p>
turnon_cmd	int32 ID	bool command_Result	<p>로봇에 장착된 LED On/Off 제어명령. * 입력으로는 LED의 ID, On이되는 가속시간, On시의 밝기 전송. * 출력으로는 호출결과 수신.</p>
led_cmd	int32 ID int32 led_brightness	bool command_Result	<p>로봇에 장착된 LED On/Off 제어명령. * 입력으로는 LED의 ID 전송. (1: 좌, 2: 우, 3: 좌우On, 4: 좌우 Off) * 출력으로는 호출결과 수신.</p>
ledtoggle_cmd	int32 light_accel int32 led_High_brightness int32 light_decel int32 led_Low_brightness	bool command_Result	<p>로봇에 장착된 LED Toggle 제어명령. (깜박이 동작) * 입력으로는 LED의 On이되는 가속시간, On시의 밝기, Off되는 가속시간, Off시의 밝기 값을 전송. * 출력으로는 호출결과 수신.</p>
gotocancel_cmd	string location_id	bool command_Result	<p>로봇의 특정위치 이동취소 명령. * 입력으로는 공백문자열을 전송. * 출력으로는 호출결과 수신.</p>
setspeed_cmd	float32 speed	float32 set_vel bool command_Result	<p>로봇의 최대속도를 설정하는 명령. * 입력으로는 로봇의 최대 선속도 값(double)을 전송. (기본값: 0.5) 단위는 m/s * 출력으로는 호출결과와 적용된 선속도 값을 전송.</p>
servo_cmd	int32 data	bool command_Result	<p>로봇의 구동부 서보모터를 강제로 On/Off하는 명령. * 입력으로는 data(1이면 On, 2이면 Off)를 전송. * 출력으로는 호출결과를 전송.</p>
docking_cmd	int32 id int32 data	bool command_Result	<p>충전스테이션과의 Docking명령. * 입력으로 id는 0, data는 1을 전송 * 출력으로는 호출결과를 전송. # 충전스테이션의 id가 변경되면, 변경된 id번호를 넣어야 함.</p>
docking_stop			<p>로봇과 충전스테이션과의 docking동작을 강제로 정지시키는 명령. * 입력과 출력이 없습니다. * 명령이 호출되면 로봇의 선속도과 각속도를 0으로 만들고 정지합니다.</p>
virtual_obstacle_	int32[] list_count	bool command_Result	가상벽 /장애물을 그리는 명령.

cmd	<code>float32[] form_x</code> <code>float32[] form_y</code> <code>float32[] form_z</code>		<p>*입력:</p> <ul style="list-style-type: none"> - list_count: 가상벽/장애물의 개수 - mode: 장애물의 형태를 정하는 변수.(1: 점 또는 원, 2: 선, 그외: 다각형) - form_x: 장애물의 X좌표. - form_y: 장애물의 Y좌표. - form_z: mode가 1인 상태에서 해당값이 0보다 크면 원의 지름을 의미하며, 그 이외에는 모두 0.0을 사용 <p>* 출력으로는 호출결과를 전송.</p> <p>※ 좌표는 amcl_pose의 좌표기준이며, 단위는 m단위임.</p> <p>※ 서비스 사용방법 예시</p> <ul style="list-style-type: none"> - 우측 그림과 같이 로봇 전방에 지름0.5m사이즈의 원을 1, 2, 3m지점에 하나씩 생성할 경우 * list_count[]: [1, 1, 1] * form_x: [1.0, 2.0, 3.0] * form_y: [0.0, 0.0, 0.0] * form_z: [0.5, 0.5, 0.5]
landmarklist_cm d		<code>int32 list_num</code> <code>string[] landmark_name</code> <code>bool command_Result</code>	<p>로봇에 저장된 Landmark의 정보 리스트 확인</p> <ul style="list-style-type: none"> - 입력은 없고 출력만 있음 * list_num: Landmark정보 개수 * landmark_name: Landmark이름(문자열) 배열 수신. * command_Result: 함수 호출결과
delete_landmark _cmd	<code>stringLandmark</code>	<code>bool command_Result</code>	<p>로봇의 저장된 Landmark정보 삭제</p> <ul style="list-style-type: none"> * 입력으로 Landmark정보가 저장된 text file명을 입력 (예: marker_0 -> 0번 landmark) * 리턴으로는 삭제여부 결과가 수신.
setinitpose_cmd		<code>int8 m_iAR_tag_id</code> <code>float64init_position_x</code> <code>float64init_position_y</code> <code>float64init_position_z</code> <code>float64init_orientation_z</code> <code>float64init_orientation_w</code> <code>bool command_Result</code>	<p>해당 AR Tag ID를 확인 한 후그 위치로 위치보정 실시</p> <ul style="list-style-type: none"> - 입력은 없고 AR Tag 가 인식되면 출력으로 해당 AR Tag ID 와 위치 정보가 출력됨
output_cmd	<code>int8 Output0</code> <code>int8 Output1</code> <code>int8 Output2</code> <code>int8 Output3</code> <code>int8 Output4</code> <code>int8 Output5</code> <code>int8 Output6</code> <code>int8 Output7</code>	<code>bool command_Result</code>	<p>TETRA-DS5에 내장된 GPIO의 기능으로서 Digital Output 8개 채널을 개별 On/Off제어할 수 있는 기능.</p> <ul style="list-style-type: none"> * 입력으로 각각의 Digital Output에 1을 입력하면 On, 0을 입력하면 Off * 단, 1 또는 0 만 입력해야 함.
patrol_cmd	<code>bool on</code> <code>int32 list_count</code> <code>string[] location_name</code>	<code>bool command_Result</code>	<p>TETRA-DS5에 저장된 Location들을 Patrol하는 기능.</p> <p>*입력</p> <ul style="list-style-type: none"> .on: Patrol의 활성화 여부 (True는 enable, False는 disable) .list_count: location의 개수 .location_name: location문자열을 넣을 배열. <p>예시) P1과 P2를 Patrol하고 싶은 경우 on은 True, list_count는2, location_name은['P1','P2']를 입력</p> <p>* 출력으로는 호출결과를 리턴.</p>

gotoconveyor_cm d	<code>string</code> Location <code>int32</code> id <code>int32</code> movement	<code>bool</code> command_Result	TETRA-DS5를 고정형 Convetor위치로 이동하는 명령 (Conveyor위치에는 지정된 AR_tag ID가 있어야 함) * 입력 . Location: Conveyor위치 이름을 저장한 문자열 . id: Conveyor에 부착된 AR_tag ID 번호 . movement: loading or unloading의 선택 (0: nomal, 1: Loading, 2: Unloading) * 출력: 함수호출 결과를 리턴
loadingcheck_se rvice_cmd		<code>bool</code> command_Result	TETRA-DS5가 Conveyor에 Docking하여 Loading을 완료했는지를 확인하는 함수. * 출력으로 Loading완료여부를 리턴.
unloadingcheck _service_cmd		<code>bool</code> command_Result	TETRA-DS5가 Conveyor에 Docking하여 Unloading을 완료했는지를 확인하는 함수. * 출력으로 Unloading완료여부를 리턴.
savemark_cmd		<code>bool</code> command_Result	TETRA-DS5의 Mapping Mode에서만 사용이 가능합니다. * 해당 서비스를 호출하면, 확인된 AR_tag ID를 저장합니다.
deletedataall_cm d		<code>bool</code> command_Result	TETRA-DS5의 Map, Location, Landmark Data들을 모두 삭제합니다. * 해당 서비스를 호출하면, /home/tetra/catkin_ws/src/tetraDS_2dnav/maps/ 경로에 있는 Map data와 /home/tetra/DATA/ 경로에 있는 Location data, /home/tetra/LANDMARK/ 경로에 있는 Landmark data를 모두 삭제합니다.
CAL_cmd		<code>bool</code> command_Result	Conveyor에 장착된 Loadcell의 Calibration을 실시합니다. * 해당 함수를 호출한 상태에 무게를 0.0Kg으로 설정합니다. * 해당 함수는 초기에 한번만 호출하면 ROM에 저장됩니다.
Auto_Move_cm d	<code>int32</code> start	<code>bool</code> command_Result	TETRA의 Conveyor 자동조작 기능 함수입니다. * start는 Loading or Unloading을 Start/Stop하는 설정입니다. . start: 0 --> Loading or Unloading Stop . start: 1 --> Loading or Unloading Strat
Manual_Move_c md	<code>int32</code> mode	<code>bool</code> command_Result	TETRA의 Conveyor 수동조작 기능 함수입니다. * mode에는 동작 방향에 대한 명령값을 입력합니다. . mode: 0 --> Conveyor정지 . mode: 1 --> Unloading 동작 실행. . mode: 2 --> Loading 동작 실행.
parameter_read_ cmd	<code>int32</code> num	<code>int32</code> data <code>bool</code> command_Result	TETRA-DS5 Motor Drive관련 Parameter의 읽기 함수입니다. * Input: num에는 Parameter번호를 입력합니다. (0~30 번까지) * Output: . data에는 해당 Parameter번호의 데이터 값이 리턴됩니다. . command_Result는 호출결과가 리턴됩니다. 단, HW의 변경이 없는 한 해당 기능은 사용하지 마세요.

parameter_write_cmd	<code>int32 num</code> <code>int32 data</code>	<code>bool command_Result</code>	<p>TETRA-DS5 Motor Drive 관련 Parameter의 쓰기 함수입니다.</p> <p>* Input: . num에는 Parameter 번호를 입력합니다. (0~30번까지) . data에는 해당 Parameter 번호에 적용할 값을 입력합니다.</p> <p>* Output: command_Result는 호출 결과가 리턴됩니다. 단, HW의 변경이 없는 한 해당 기능은 사용하지 마세요.</p>
mode_change_cmd	<code>int32 mode</code>	<code>bool command_Result</code>	<p>TETRA-DS5 Motor Drive 관련 동작 모드 변환 함수입니다.</p> <p>* Input: mode가 1인 경우 Position Mode, 0인 경우 Velocity Mode입니다. (기본 0)</p> <p>* Output: command_Result는 호출 결과가 리턴됩니다. 단, 초기 설정 기능임으로 평상시에 사용하지 않습니다.</p>
linear_move_cmd	<code>int32 linear_position</code>	<code>bool command_Result</code>	<p>TETRA-DS5 Motor Drive 관련 전/후진 동작 함수입니다.</p> <p>* Input: linear_position에 위치 값을 입력합니다. (1m를 이동할 경우 10000을 입력)</p> <p>* Output: command_Result는 호출 결과가 리턴됩니다. 단, 초기 설정 기능임으로 평상시에 사용하지 않습니다.</p>
angular_move_cmd	<code>int32 Angular_degree</code>	<code>bool command_Result</code>	<p>TETRA-DS5 Motor Drive 관련 좌/우회전 동작 함수입니다.</p> <p>* Input: Angular_degree에 회전 각도 값을 입력합니다. (90도를 회전할 경우 9000을 입력)</p> <p>* Output: command_Result는 호출 결과가 리턴됩니다. 단, 초기 설정 기능임으로 평상시에 사용하지 않습니다.</p>
pose_estimate_cmd	<code>float64 estimate_position_x</code> <code>float64 estimate_position_y</code> <code>float64 estimate_position_z</code> <code>float64 estimate_orientation_x</code> <code>float64 estimate_orientation_y</code> <code>float64 estimate_orientation_z</code> <code>float64 estimate_orientation_w</code>	<code>bool command_Result</code>	<p>RVIZ에 있는 2D Pose Estimate를 서비스로 호출할 수 있도록 제작한 함수입니다.</p> <p>* Input: 로봇을 이동하고자 하는 위치와 자세 데이터 * Output: 호출 결과</p>
gpio_status_cmd		<code>uint8 Input0</code> <code>uint8 Input1</code> <code>uint8 Input2</code> <code>uint8 Input3</code> <code>uint8 Input4</code> <code>uint8 Input5</code> <code>uint8 Input6</code> <code>uint8 Input7</code> <code>uint8 Output0</code> <code>uint8 Output1</code> <code>uint8 Output2</code> <code>uint8 Output3</code> <code>uint8 Output4</code>	<p>TETRA-DS에 있는 GPIO의 상태를 확인하는 서비스입니다.</p> <p>* Input: 없음. * Output: Input 0~7 번, Output 0~7 번 까지의 상태 리턴</p>

		<code>uint8 Output5</code> <code>uint8 Output6</code> <code>uint8 Output7</code> <code>bool command_Result</code>	
set_ekf_cmd	<code>float64 init_position_x</code> <code>float64 init_position_y</code> <code>float64 init_position_z</code> <code>float64 init_orientation_x</code> <code>float64 init_orientation_y</code> <code>float64 init_orientation_z</code> <code>float64 init_orientation_w</code>	<code>bool command_Result</code>	<p>TETRA-DS에 현 위치로 EKF Pose와 Imu를 모두 초기화하는 함수입니다.</p> <p>*Input: map좌표 기준으로의 현 로봇 pose값</p>

9-2. TETRA-DSV ROS TOPIC

Topic name	Message Type	Description
odom	<code>nav_msgs/Odometry</code> [http://docs.ros.org/en/noetic/api/nav_msgs/html/msg/Odometry.html]	로봇 기준의 position 정보와 orientation 정보, 선속도와 각속도 정보.
amcl_pose	<code>geometry_msgs/PoseWithCovarianceStamped</code> [http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseWithCovarianceStamped.html]	공분산과 함께 맵에서 로봇의 추정된 포즈 지도상 기준의 로봇 position 정보와 orientation 정보.
joy	<code>sensor_msgs/Joy</code> <code>axes[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]</code> <code>buttons[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]</code>	조이스틱 입력에 대한 스틱과 버튼정보. (단, 조이스틱 입력이 있을 때만 Update)
PoseRest	<code>data</code>	<code>data</code> 가 1이되면, 로봇의 Odometry 가 모두 0으로 리셋. (리셋 후에는 0으로 변경됨) 초기 프로그램 실행 시와 충전스테이션 도킹완료 시 자동호출 됨.
cmd_vel	<code>data</code>	로봇의 선속도와 각속도가 입력되는 정보. (Navigation 동작 시 Update 됨)
accel_vel	<code>data</code>	주의! _ 가/감속 슬로프 시간을 주는 기능으로서 주행성능에 영향을 줄 수 있습니다.
Servo_ON	<code>data</code>	로봇의 모터 전원 On/Off 정보. (Data 가 1이면 On, 2이면 Off, 0이면 마지막 상태 유지)
tetra_battery	<code>Data</code>	로봇 배터리 정보 0~100까지의 배터리 용량 %가 Update 됨
docking_status	<code>data</code>	로봇과 충전스테이션 간의 도킹 상태를 표시. 1: Nomal 상태. 2: 로봇의 단자와 충전스테이션의 단자가 접촉된 상태. 3: 도킹동작 완료 후 충전을 시작하는 상태. 4: Error 가 발생된 상태. 5: 충전이 완료된 상태. 11: Conveyor Loading station Contact & Loading possible

		12: Conveyor Loading station Contact & Loading impossible 13: Conveyor Unloading station Contact & Unloading possible 14: Conveyor Unloading station Contact & Unloading impossible
emg_state	data	로봇에 장착된 비상정지 스위치의 On/Off 상태 정보. 1: 비상정지 스위치 On 2: 비상정지 스위치 Off
bumper_data	data	로봇 전방에 장착된 범퍼스위치의 On/Off 상태 정보. 1: 범퍼입력 On 2: 범퍼입력 Off
bumper_pointcloud	data[](PointCloud2)	로봇 전방에 장착된 범퍼스위치의 On/Off 상태를 Pointcloud2 데이터로 변환한 정보
move_base/result	move_base_msgs/MoveBaseActionResult - status.status - status.text	로봇의 Navigation 동작에 대한 상태 표시 status: 상태별 번호가 리턴 text: 상태 설명(영문 문자열)
move_base/TebLocalPlanner ROS/global_plan	http://docs.ros.org/en/api/nav_msgs/html/msg/Path.html	로봇 Navigation 글로벌 계획 (시각화 목적으로만 사용)
move_base/TebLocalPlanner ROS/local_plan	http://docs.ros.org/en/api/nav_msgs/html/msg/Path.html	로봇 Navigation 로컬 계획 (시각화 목적으로만 사용)
move_base/TebLocalPlanner ROS/obstacles	http://docs.ros.org/en/api/costmap_converter/html/msg/ObstacleArrayMsg.html	장애물의 정보제공 토픽
Ultrasonic_D_L	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 좌측 하단에 장착된 초음파 센서 데이터 (단위는 m)
Ultrasonic_D_R	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 우측 하단에 장착된 초음파 센서 데이터 (단위는 m)
Ultrasonic_R_L	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 후방 좌측에 장착된 초음파 센서 데이터 (단위는 m)
Ultrasonic_R_R	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 후방 우측에 장착된 초음파 센서 데이터 (단위는 m)
range_points_DL	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 좌측 하단에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
range_points_RL	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 우측 하단에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
range_points_RR	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 후방 좌측에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
range_points_DR	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 후방 우측에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
map	http://docs.ros.org/en/api/nav_msgs/html/msg/OccupancyGrid.html	환경지도 정보
docking_progress	data	도킹상태를 알 수 있는 토픽 -1: 도킹실패.

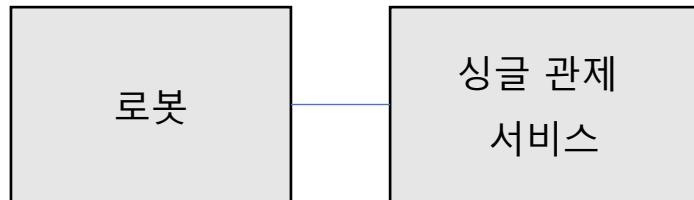
		1: 도킹명령 시작. 2: 마커의 위치까지 접근. 3: 자세에 대한 보정_1 4: 자세에 대한보정_2 5: 스테이션 위치까지 직진. 6: 도킹완료.
ar_pose_marker	ar_pose_marker (ar_track_alvar/AlvarMarkers)	검출된 ar_tag 의 포즈 목록이 출력됩니다. - ar_tag 의 ID 정보, 위치/자세 정보
camera1/.. camera2/..	https://github.com/IntelRealSense/realsense-ros	TETRA-DS5전방 하단 좌/우에 장착된 각각의 RGBD-Camera 관련 정보.
depthimage_to_laserscan1/.. depthimage_to_laserscan2/..	http://wiki.ros.org/depthimage_to_laserscan	TETRA-DS5전방 하단 좌/우에 장착된 각각의 RGBD-Camera 의 Pointcloud 정보를 Laser scan 정보로 변환한 정보.
particlecloud	geometry_msgs/PoseArray http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	필터에서 유지 관리하는 포즈 추정값 세트입니다.
scan	sensor_msgs/LaserScan http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	SICK TIM571에서 획득한 Laser Scan Data 정보입니다.
pcl_1	sensor_msgs/LaserScan http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	TETRA 의 전방 하단 좌측에 위치한 RGBD-Camera(camera1)의 Point Cloud 를 Laser Scan Data 로 변환한 정보입니다.
pcl_2	sensor_msgs/LaserScan http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	TETRA 의 전방 하단 우측에 위치한 RGBD-Camera(camera2)의 Point Cloud 를 Laser Scan Data 로 변환한 정보입니다.
usb_cam/..	http://wiki.ros.org/usb_cam	후방 USB 카메라 관련 정보입니다.
marker/node	http://docs.ros.org/en/api/visualization_msgs/html/msg/Marker.html	작성된 환경지도(map)상에 기록된 Landmark(ar_tag)의 정보가 담겨있습니다. 저장한 Landmark 의 개수 대로 출력됩니다.
gpio_status	GPIO_msg Input0, Input1,Input2,Input3,Input4,Input5,Input6, Input7, Output0,Output1,Output2,Output3,Output4,Output5,Output6,Output7	TETRA-DS5에 내장된 Digital Input8개와 Digital Output8개의 상태 정보. * 1은 On, 0은 Off 를 의미합니다. * GPIO 라는 custom msg 를 사용합니다.
Camera1/Imu	sensor_msgs/Imu http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Imu.html	Realsense D435i 1번으로부터 전달받은 IMU 정보입니다. * geometry_msgs/Quaternion orientation: N/A * geometry_msgs/Vector3 angular_velocity: x, y, z axis * geometry_msgs/Vector3 linear_acceleration: x, y, z axis
battery_voltage	data (Float64)	TETRA-DS5 배터리의 전압(Voltage)정보입니다. * 0.1V 단위로 값이 업데이트 됩니다.
battery_current	data (Float64)	TETRA-DS5 배터리의 전류(Current)정보입니다. * 0.1A 단위로 값이 업데이트 됩니다.
conveyor_loadcell	data (Float64)	Conveyor 에 내장된 Loadcell 에서 측정한 weight 정보입니다. * 0.1Kg 단위로 업데이트 되며, 일정 시간평균을 낸 값을 업데이트 합니다.

		* 단, 물건이 없는 상태는 0.0~ 0.3Kg 값을 출력합니다. (노이즈 포함)
conveyor_sensor	data	TETRA 에 장착된 Conveyor 의 입구/출구 센서 상태값을 업데이트 합니다. * data: 0 --> 입구/출구 센서 모두 Off 상태. * data: 1 --> 입구 센서 On 상태. * data: 2 --> 출구 센서 On 상태. * data: 3 --> 입구/출구 센서 모두 On 상태.
conveyor_movement	data(Int32)	TETRA 에 장착된 Conveyor 의 동작 상태값을 업데이트 합니다. * data: 0 or 1 --> 동작정지 상태 * data: 3 --> 로딩 동작 중. * data: 5 --> 언로딩 동작 중.
imu/data	http://docs.ros.org/en/melodic/api/sensor_msgs/html/msg/Imu.html	TETRA 에 장착된 IMU 센서 데이터

Chapter 10. TETRA DS V 싱글 관제 서비스 사용 가이드

플랫폼 별 주요 운영 기능 요약

→ TETRA-DS V 의 싱글 관제 서비스는 [싱글 관제 서비스], [제어 대상 로봇]으로 구성됩니다.

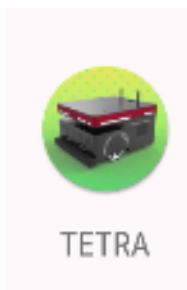


태블릿을 통한 [싱글 관제 서비스] 서비스에서 운영에 관련된 모든 기능 제어

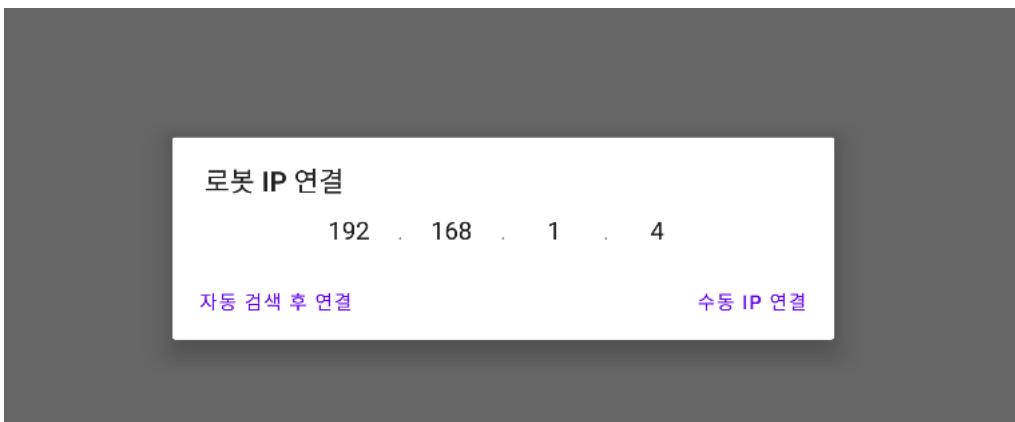
1. 모니터링 : 로봇의 실시간 주행 상태, 위치를 지도에서 확인하며 이동을 제어할 수 있습니다.
2. 지도관리 : 복수의 지도를 매핑하고 관리할 수 있습니다.
3. 위치관리 : 지도 내에 논리적 위치 정보 등을 등록, 관리할 수 있습니다.
4. 태스크 매니저 : 지정된 업무를 등록하여 태스크를 생성, 수행, 관리할 수 있습니다.
5. 로봇 설정 : 로봇의 속도, 충전 기준, 업무 수행 기준 등을 설정할 수 있습니다.

10-1. TETRA APP 실행

1. 설정 --> 연결 --> Wi-Fi 에 들어가 접속할 로봇의 Wi-Fi 에 연결합니다.
2. 밑의 그림과 같은 설치된 싱글 관제 서비스 APP 을 실행합니다.



3. 을 실행하면, 수동 IP 연결 화면이 나옵니다. IP 입력 후 '수동 IP 연결'을 누르면 메인화면으로 진입합니다.



4. '자동 검색 후 연결'을 클릭 시 해당 IP 대역을 내의 IP 를 검색하고 연결합니다.

자동 검색 연결

- ▶ 웹서버 검색중.. 자동으로 연결됩니다.
192.168.200.9

자동 연결 취소

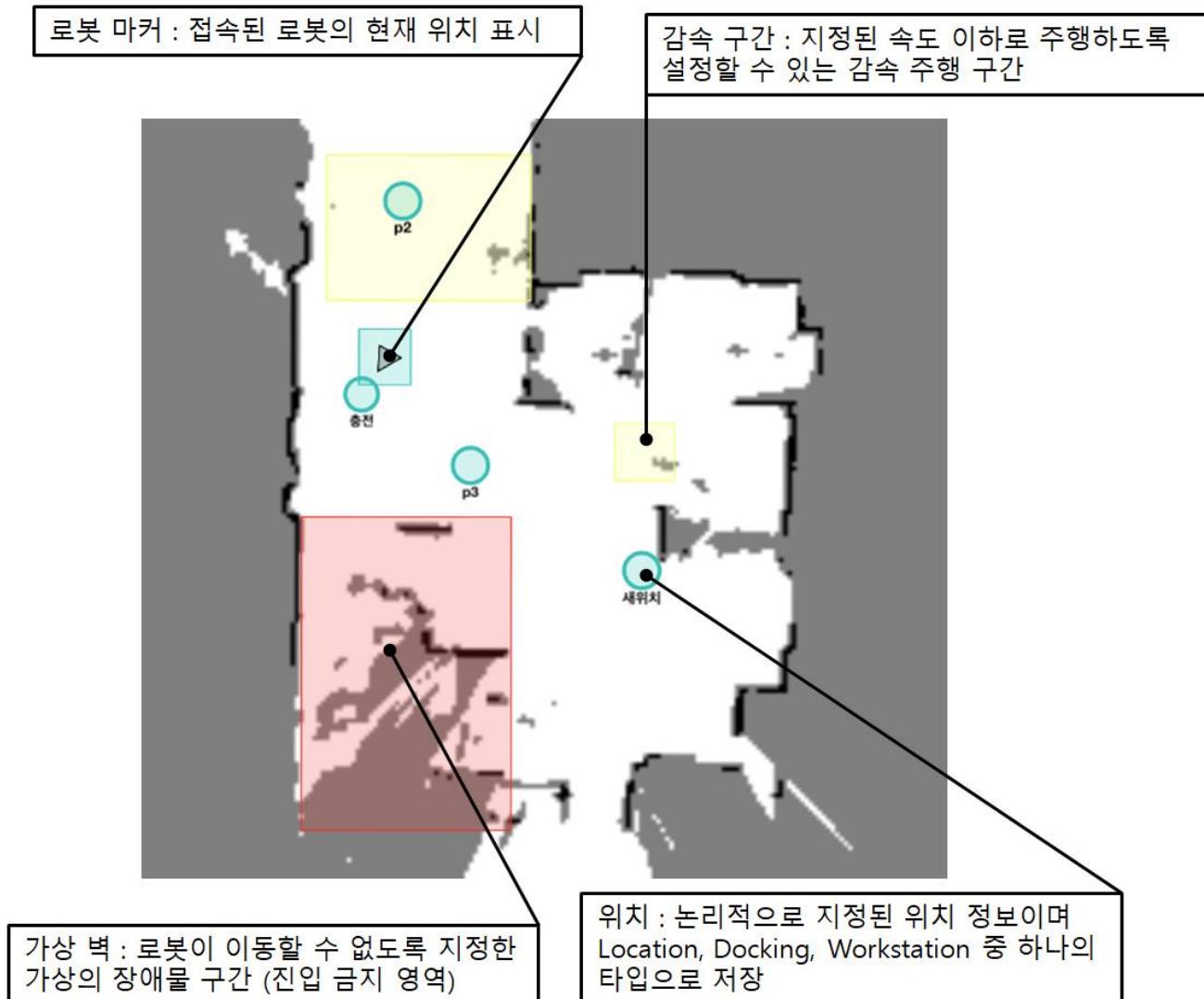
10-2. APP 메인화면

APP 메인화면에서는 현재 활성화된 지도 및 지도 상의 논리적 위치 정보, 로봇의 주행 상태가 표시됩니다. 또한, 우측의 위치 목록을 클릭하여 로봇을 이동시킬 수 있습니다.



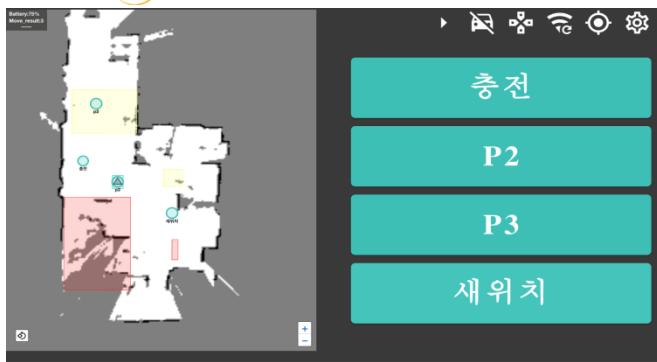
1. 로봇 모터 On/Off 버튼 : 로봇의 구동 모터를 켜거나 끌 수 있습니다. 로봇 구동 모터를 Off 할 경우, 로봇은 이동 및 태스크 수행을 할 수 없습니다.
2. 조이스틱 버튼 : 조이스틱을 켜거나 끌 수 있습니다. 조이스틱을 활성화하면 해당 버튼을 통해 로봇을 직접 운전할 수 있습니다.
3. 네트워크 재 연결 버튼 : 어플리케이션과 로봇의 연결이 끊어질 경우, 해당 버튼을 눌러 다시 재 연결을 시도할 수 있습니다.
4. 위치 조정 버튼 : 지도상의 로봇 위치와 실제 로봇의 위치가 다를 경우, 해당 버튼을 눌러 로봇의 지도상 위치를 조정할 수 있습니다.
5. 설정 버튼 : 로봇의 설정 메뉴로 이동합니다.
6. 위치 이동 버튼 : 특정 위치를 누르면 로봇이 해당 위치로 주행을 시작합니다.
7. 등록된 플랜리스트 중 선택한 플랜을 태스크에 추가 가능합니다.

10-2-1. APP 메인화면 – 지도상 표시 정보



1. 로봇 마커 : 접속된 로봇의 현재 위치 표시
2. 감속 구간 : 지정된 속도 이하로 주행하도록

10-2-2. APP 메인화면 – 로봇을 특정 위치로 이동시키기



1. 이동하고자 하는 [목적지] 를 터치



2. 이동이 시작되면 [이동정지] 버튼이 활성화 됨.

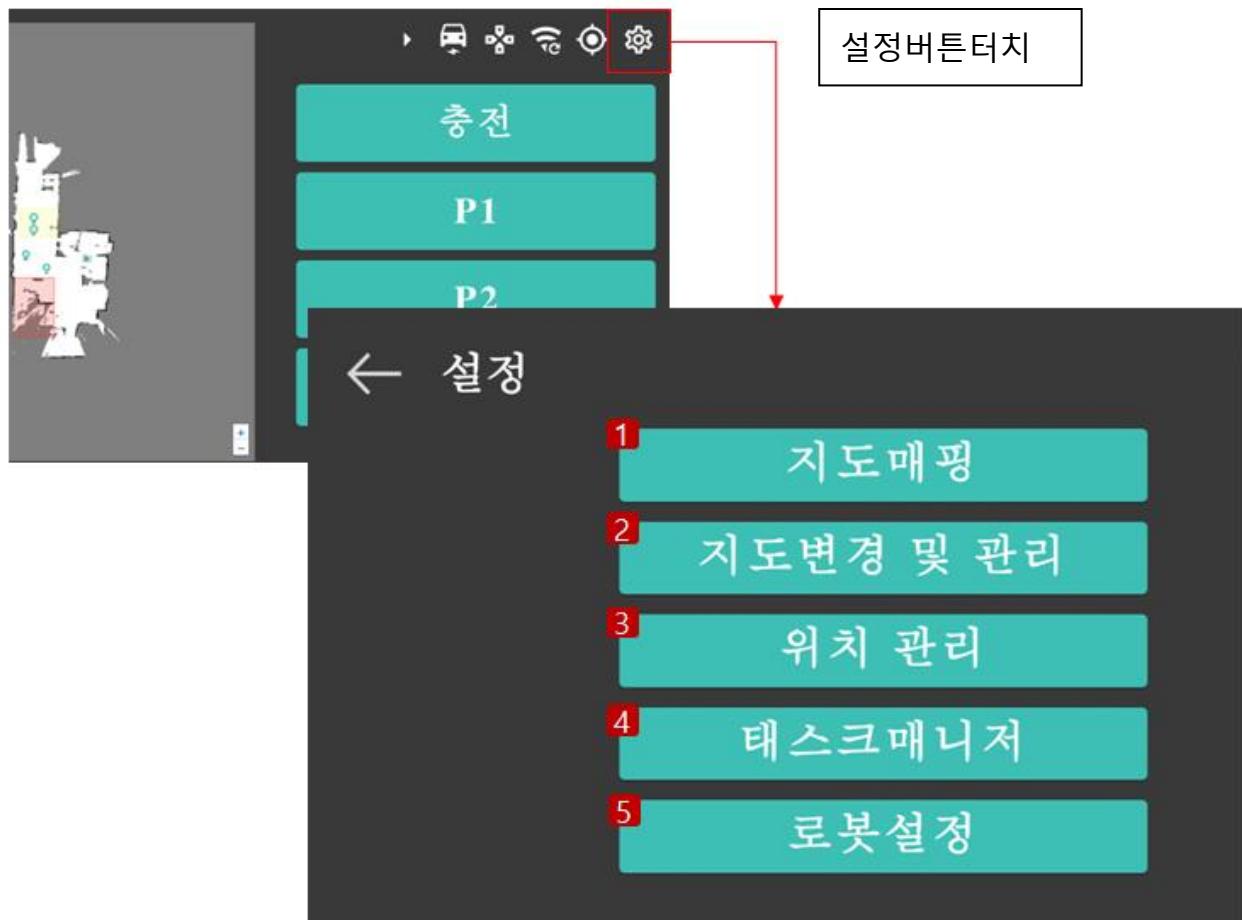


3. 이동 정지 버튼을 터치하면 [이동취소] 또는 [이동계속] 을 선택할 수 있음.



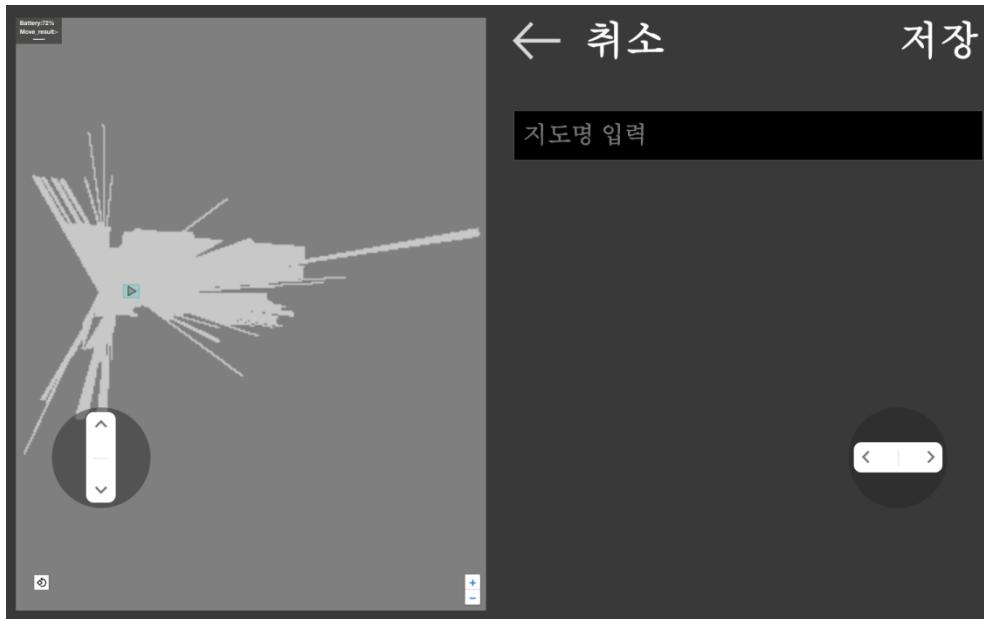
4. 목적지에 도착하면 도착 안내 화면과 함께 [메인메뉴] 로 복귀하는 버튼이 활성화됨.

10-3. 설정

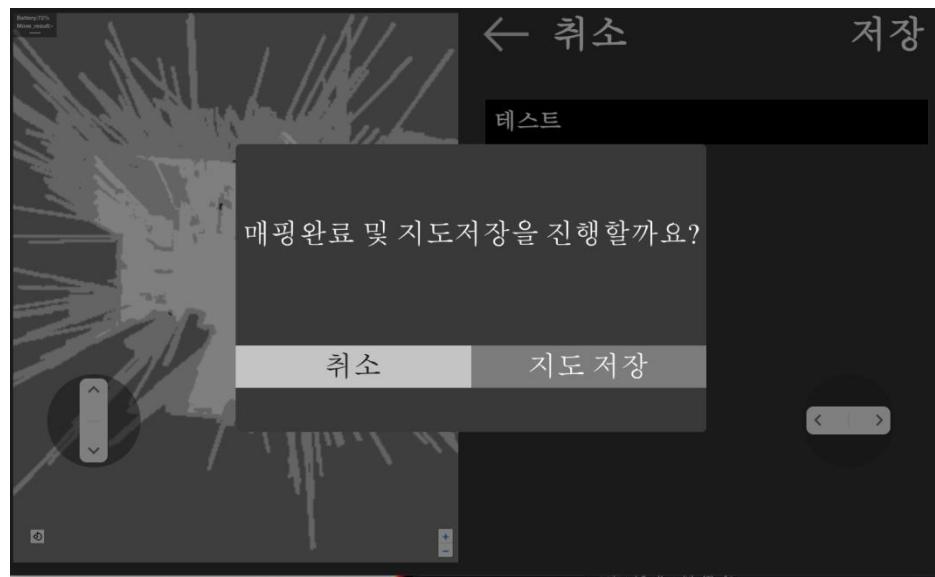


1. 지도 매핑 : 로봇으로 지도를 매핑하는 매핑 모드로 전환
2. 지도 변경 및 관리 : 저장된 전체 지도 목록으로 이동
3. 위치 관리 : 현재 지도 위에 위치 정보를 등록, 관리하는 화면으로 이동
4. 태스크 매니저 : 업무 계획을 작성하고 업무를 등록하는 태스크 매니저 화면으로 이동
5. 로봇 설정 : 로봇의 주요 설정 정보를 관리하는 화면으로 이동

10-3-1. 설정 – 지도 매핑



1. 매핑 모드가 활성화되면 좌우의 조이스틱 버튼을 조종하여 로봇으로 해당 지역을 이동하며 매핑을 시작합니다.
2. 매핑이 진행될 수록 로봇이 지나갈 수 없는 장애물 영역이 회색 > 검은색으로 매핑 됩니다.

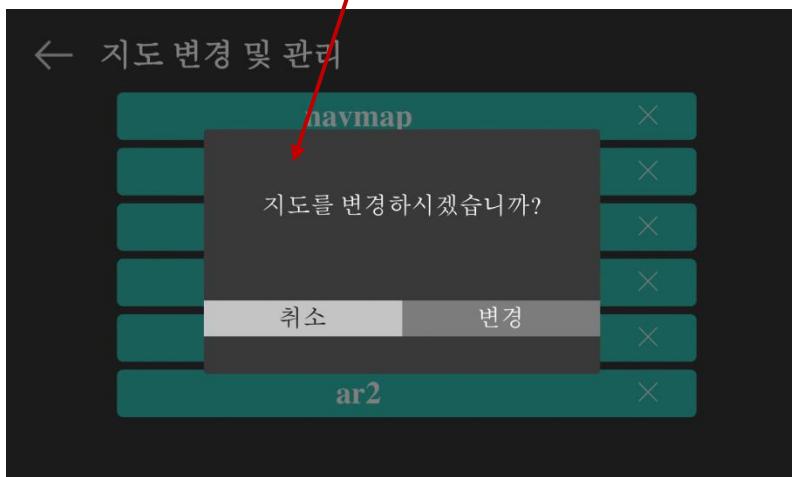


3. 모든 지도 매핑이 완료되면 지도 이름을 입력합니다.
4. 지도 매핑 완료 버튼을 터치하면 지도가 저장됩니다.

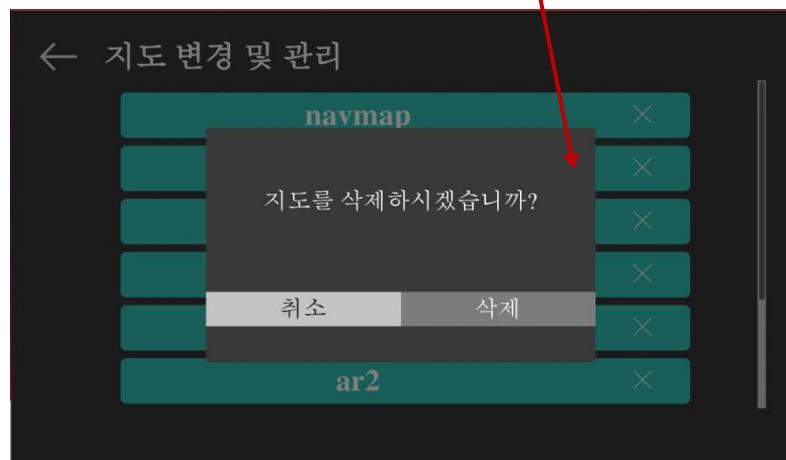
10-3-2. 설정 – 지도 변경 및 관리



1. 전체 저장된 지도 목록이 표시됨.

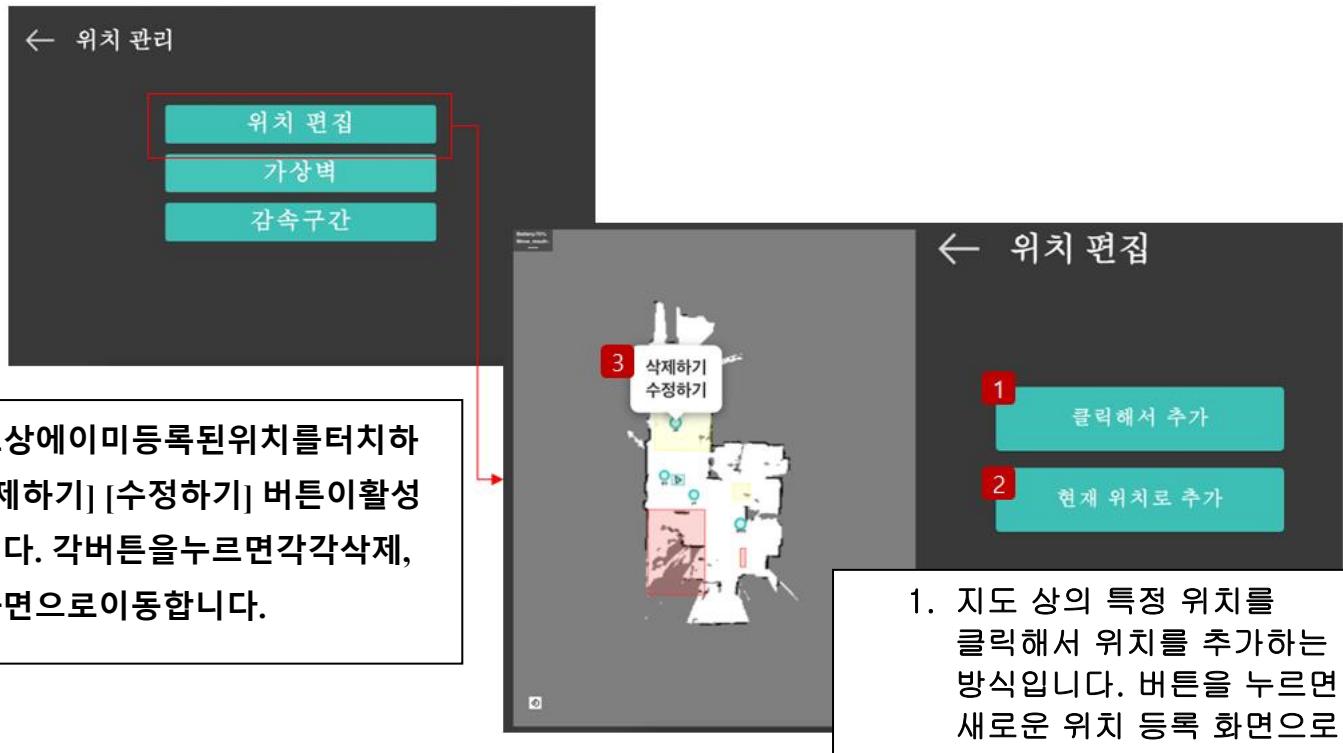


2. 현재 선택되지 않은 다른 지도를 선택하면 해당 지도로
지도가 변경됨.



3. 지도 우측의 [X] 버튼을 눌러서 해당 지도를 목록에서
삭제할 수 있음.

10-3-3. 설정 – 위치 관리 – 위치 편집



3. 지도상에 이미 등록된 위치를 터치하면 [삭제하기] [수정하기] 버튼이 활성화됩니다. 각 버튼을 누르면 각각 삭제, 수정화면으로 이동합니다.

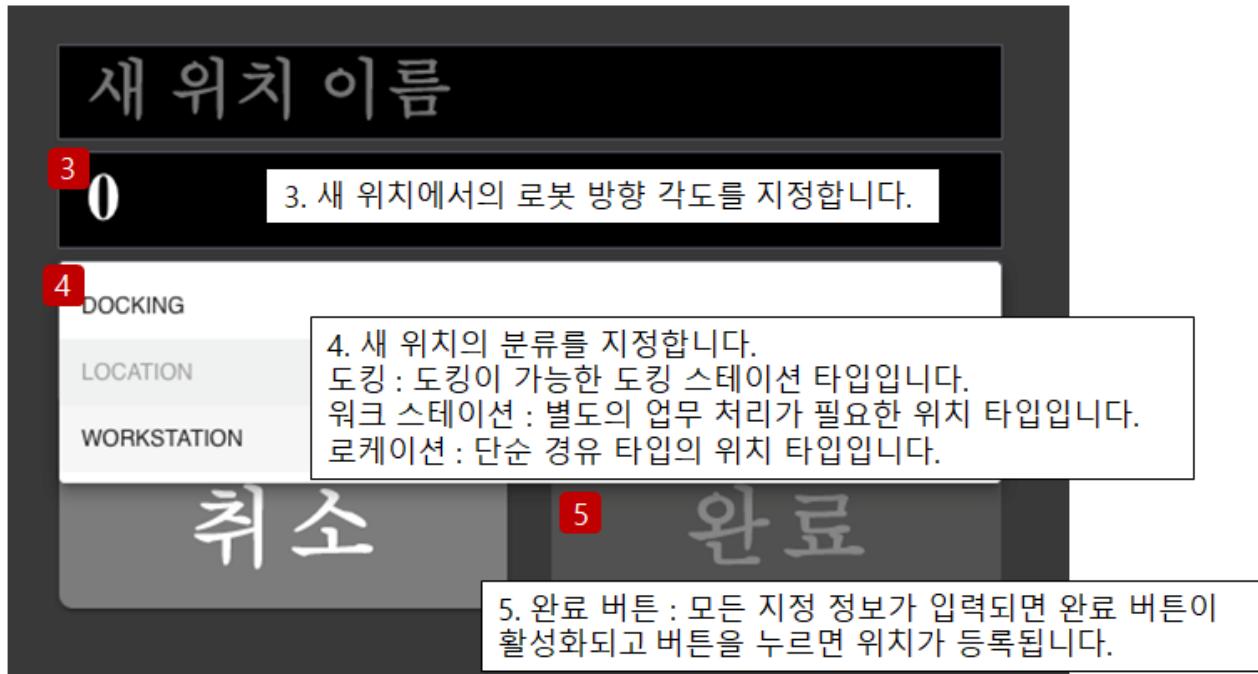
1. 지도 상의 특정 위치를 클릭해서 위치를 추가하는 방식입니다. 버튼을 누르면 새로운 위치 등록 화면으로 이동합니다.

2. 현재 로봇의 실제 위치를 추가하는 방식입니다. 버튼을 누르면 새로운 위치 등록 화면으로 이동합니다.

10-3-3 (1). 설정 – 위치 관리 – 위치 편집 - 클릭해서 추가

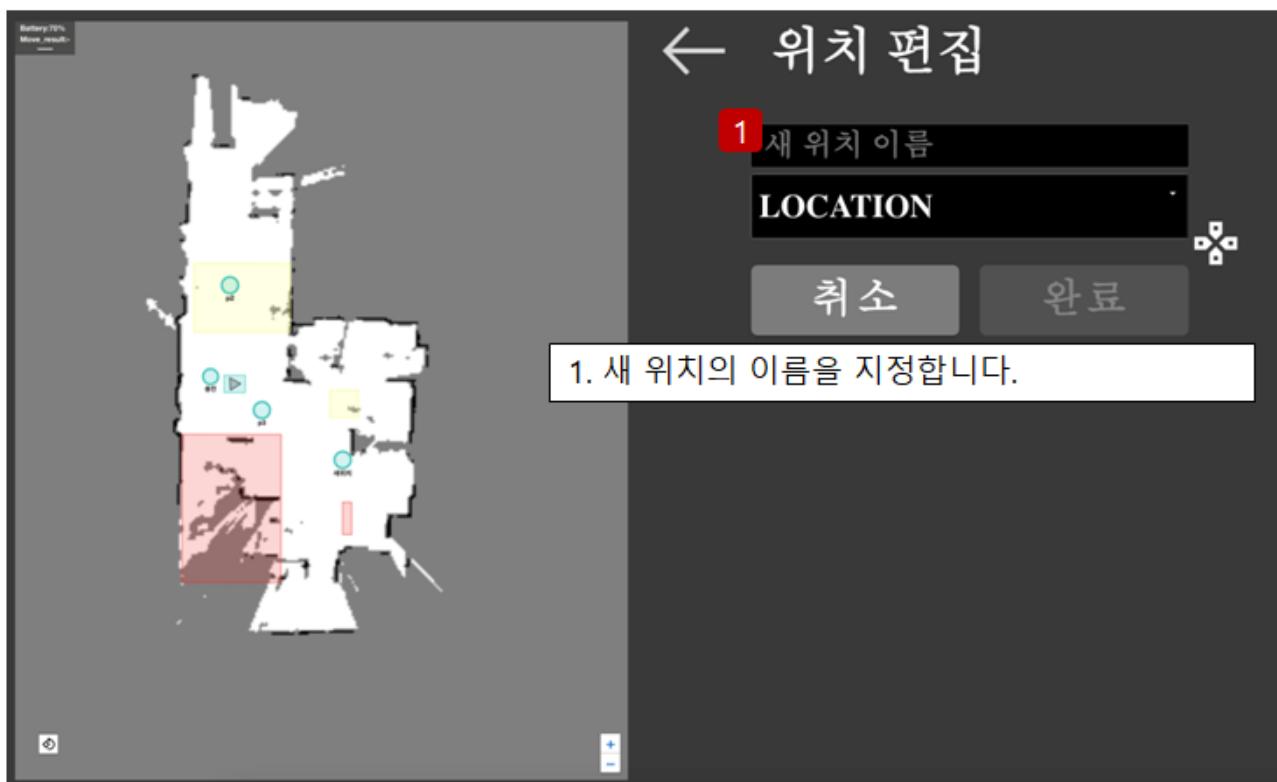
지도상의 지점을 터치하여 위치를 저장할 경우,
해당 지점을 저장하고도 착시로 봇이 정지할 각도를 입력해야 합니다.

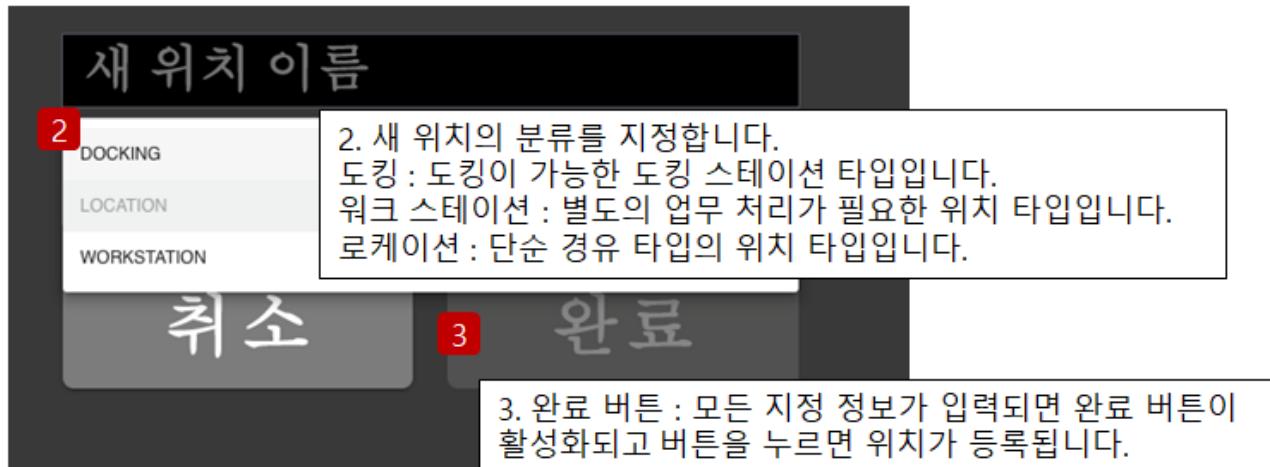




10-3-3 (2). 설정 – 위치 관리 – 위치 편집 – 로봇 위치로 추가

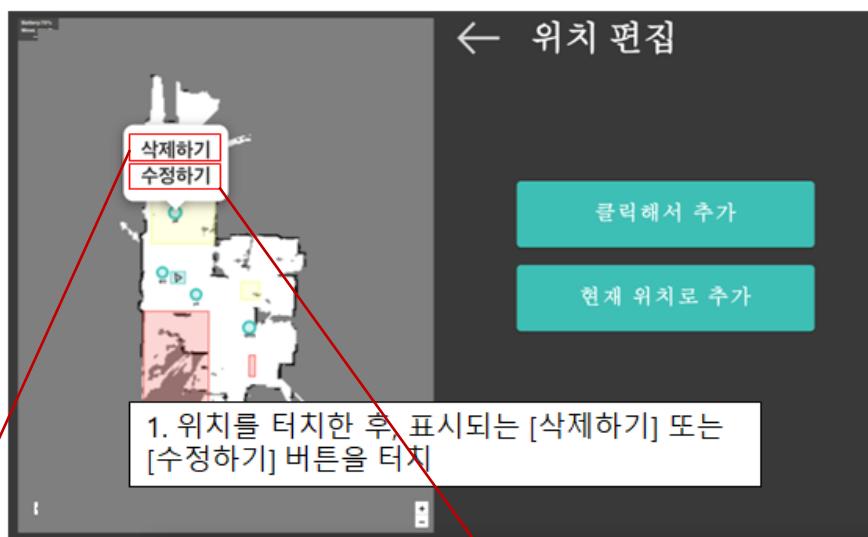
로봇위치로위치를추가할경우,
실제로봇의위치및각도를그대로저장합니다.





10-3-3 (3). 설정 – 위치 관리 – 위치 편집 – 위치 수정 및 삭제

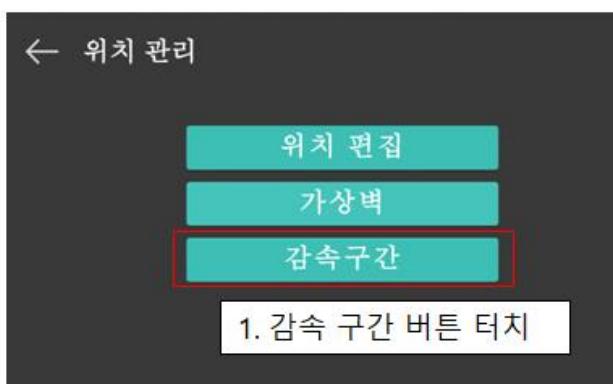
로봇위치로위치를추가할경우,
 실제로봇의위치및각도를그대로저장합니다.

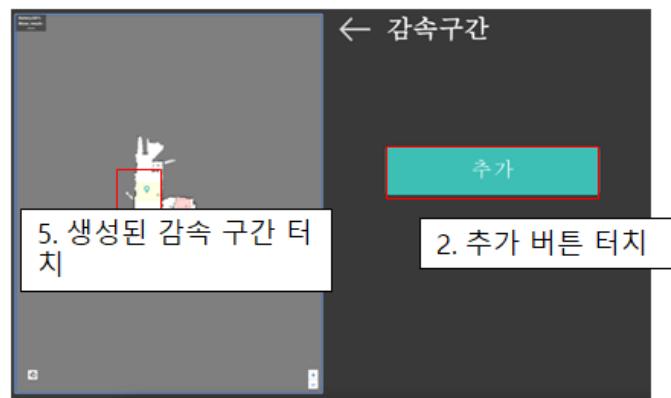




10-3-3 (4). 설정 – 위치 관리 – 감속 구간

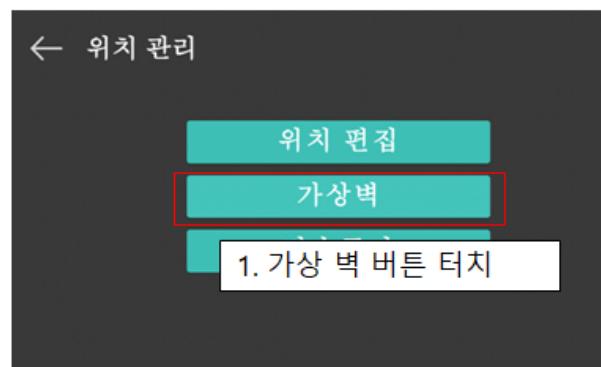
감속구간을 추가하여 로봇이 지정 속도 이하로 주행하는 지역을 설정할 수 있습니다.

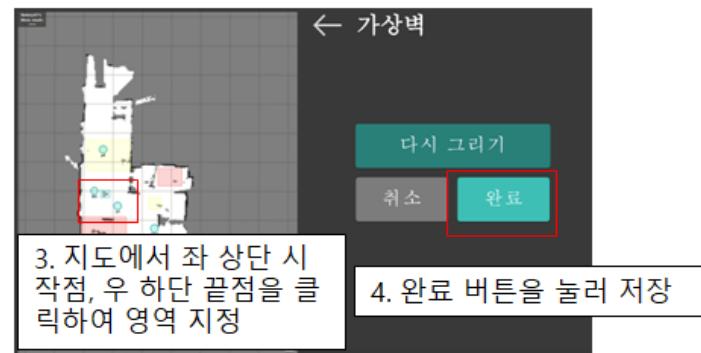




10-3-3 (5). 설정 – 위치 관리 – 가상벽

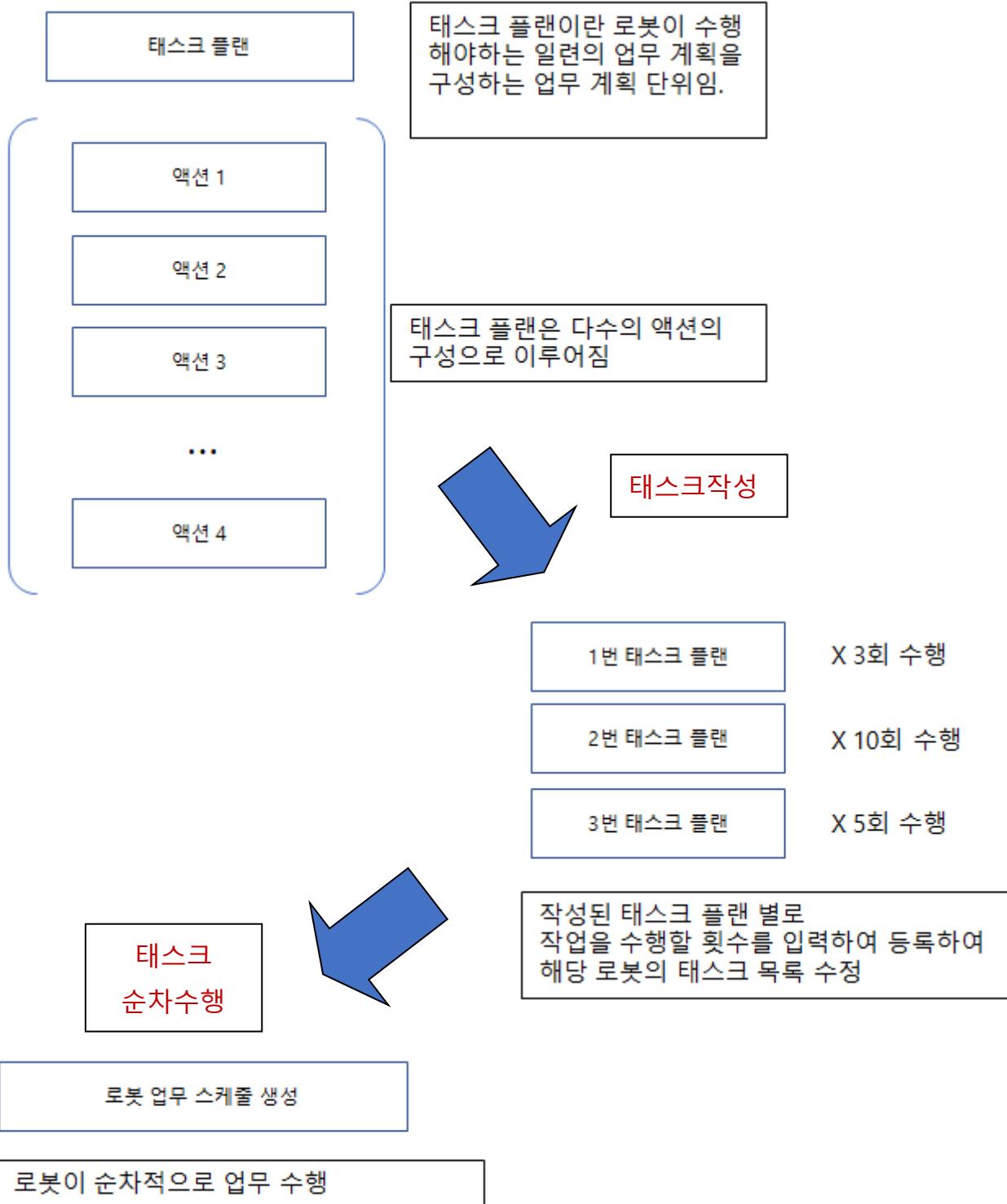
가상벽을 추가하여로봇이주행해서는안되는지역을지정할수있습니다.





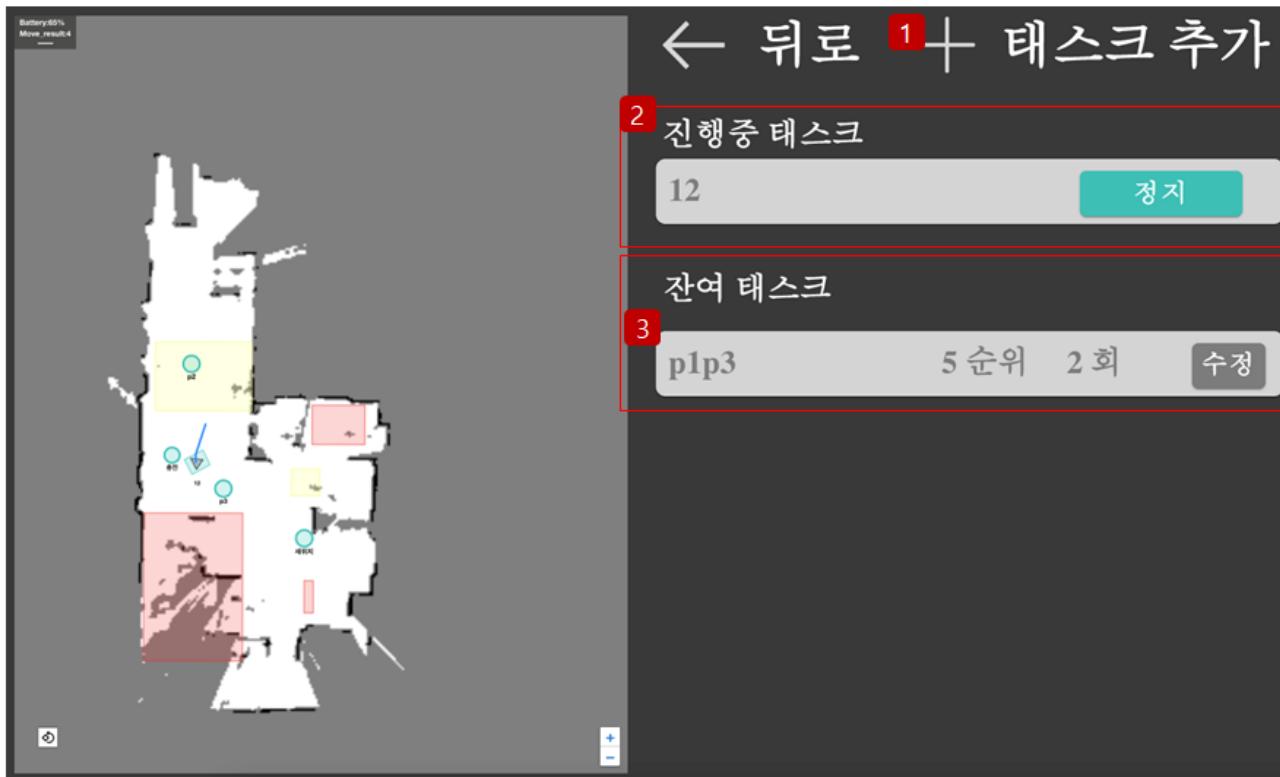
10-3-4. 설정 – 태스크 플랜

태스크플랜은 개별 액션으로 구성되어 하나의 작업 계획이 되고
구성된 작업 계획인 태스크플랜을 수행할 작업 횟수, 우선 순위와 함께 등록하면 해당 규칙에 따라
로봇이 태스크를 순차 처리함.



10-3-5. 설정 – 태스크 매니저

태스크매니저는 로봇을 지정된 업무 플랜에 따라 순차적으로 태스크를 수행하도록 설정하고 관리하는 기능입니다.

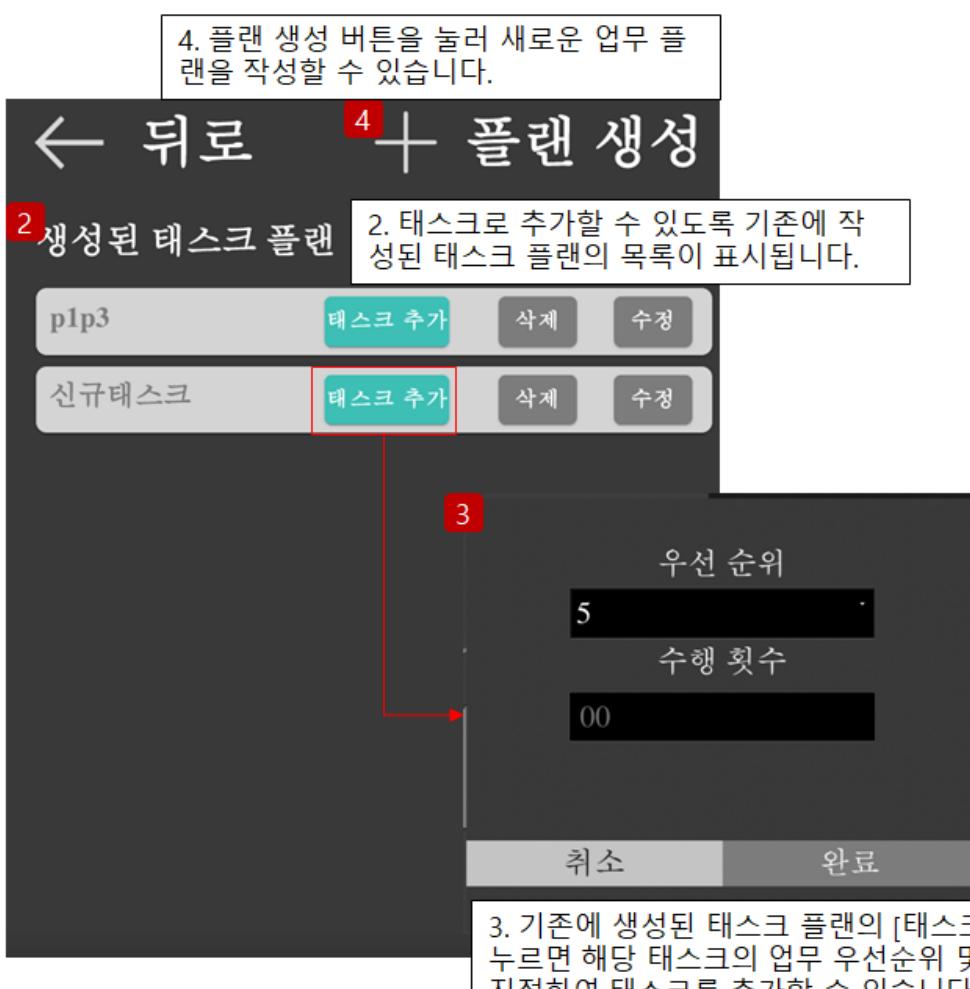


1. 태스크 추가 버튼을 터치하여 새로운 태스크 목록을 생성하는 화면으로 이동합니다.
2. 현재 로봇이 진행하고 있는 태스크입니다. 정지 버튼을 눌러 태스크를 정지 후, 재개 및 취소가 가능합니다.
3. 전체 작성된 태스크 목록입니다. 업무 우선순위에 따라 업무를 순차 진행하며, 지정된 횟수만큼 수행 시 해당 태스크는 잔여 태스크 목록에서 삭제됩니다. 또한 [수정] 버튼을 눌러 해당 태스크의 수량을 수정할 수 있습니다.

10-3-5 (1). 설정 – 태스크 매니저 – 태스크 추가

새로운업무플랜을작성하거나

기존의업무플랜에우선순위, 수행횟수를지정하여태스크목록으로추가할수있습니다.



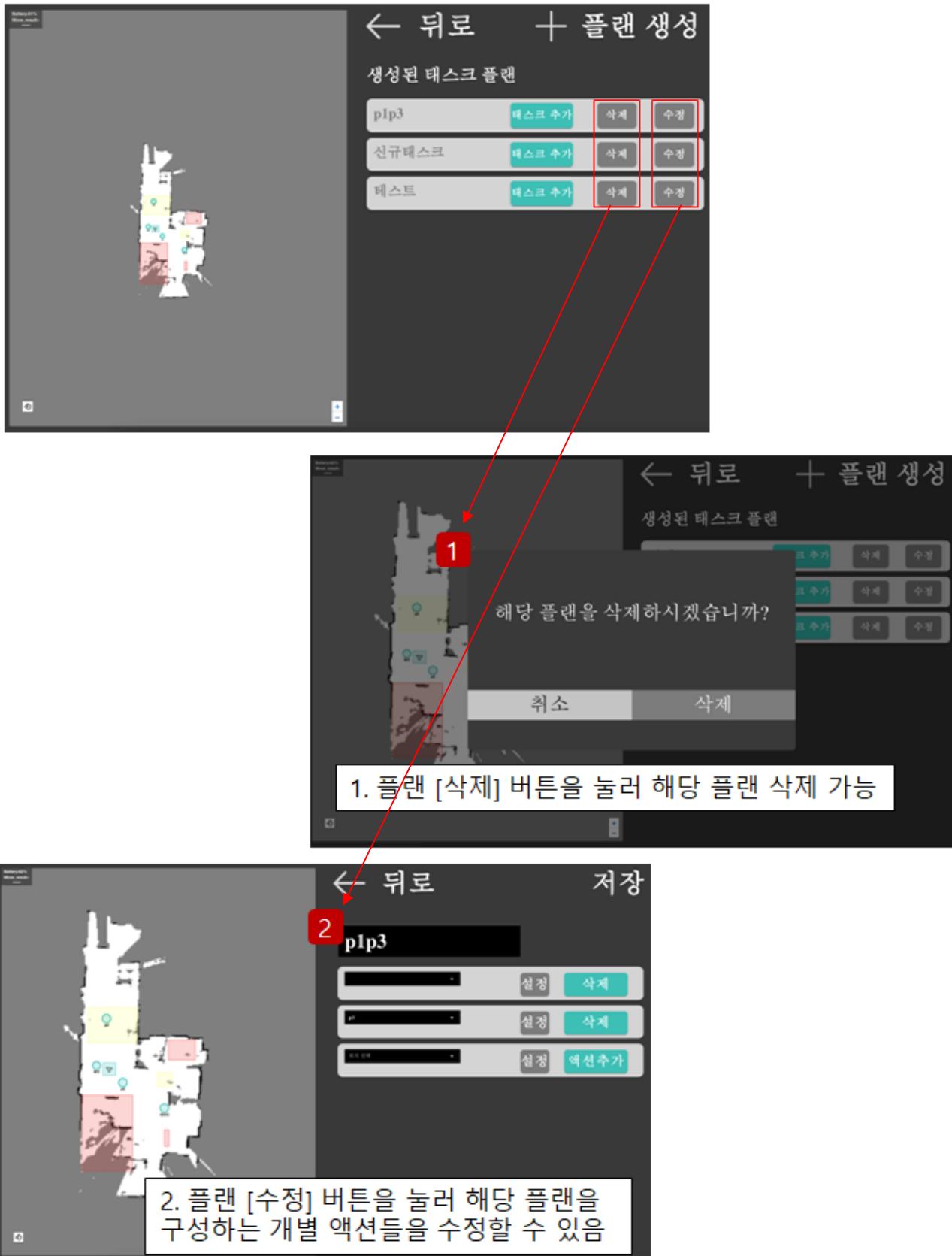
10-3-5 (2). 설정 – 태스크 매니저 – 태스크 추가 - 플랜생성

다수의 위치별 액션을 조합하여
하나의 업무 플랜을 생성할 수 있습니다.



10-3-5 (3). 설정 – 태스크 매니저 – 태스크 추가 – 플랜 수정 및 삭제

기존에 만들어진 플랜을 수정 또는 삭제 할 수 있습니다.



10-3-6. 설정 – 로봇 설정

해당 로봇의 기본 설정을 변경할 수 있습니다.



1. **로봇 기본 속도 설정** : 해당 로봇의 주행 속도를 지정합니다.
2. **자동 충전 배터리 설정** : 자동 충전을 시작할 배터리 수치를 지정합니다.
3. **자동 수행 배터리 설정** : 충전 후, 다시 업무를 재개할 배터리 수치를 지정합니다.

감사합니다