

TR Manual

Wego & Industrial Robot

1. Hardware Specification
2. Control TR Using ROS
3. TR ROS Topic

01

Hardware Specification

01 Hardware Specification

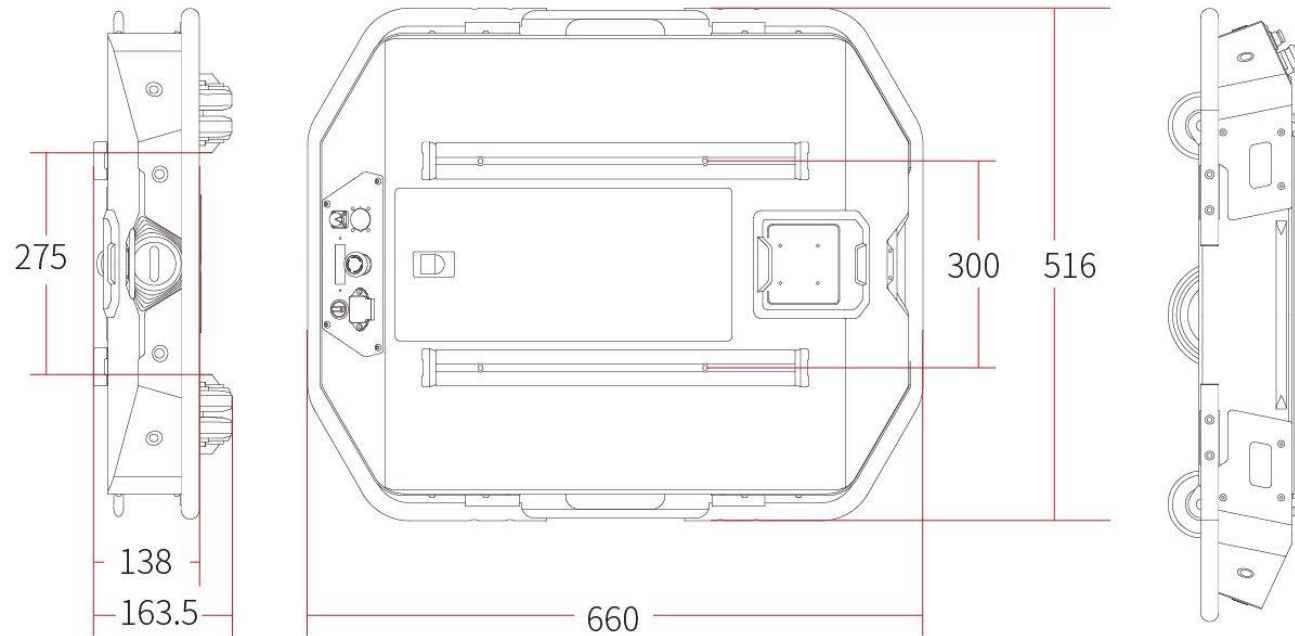
- TR 스펙사항

Dimensions(L X W X H)	• 685 X 570 X 155 (mm)
Overall weight	• 28~30 (Kg)
Track	• 360 (mm)
Rated Progressive Load	• 100 (Kg)
Minimum Turning Radius	• 0m (Can Rotate In place)
Climbing angle	• < 8°
Obstacle crossing ability	• 10MM
Max. speed	• 6KM/H
Max. stroke	• 15KM
Suspension form	• Independent Suspension With Rocker Arm



01 Hardware Specification

- Dimensions



01 Hardware Specification

Q1 : RS232(현재
미지원)

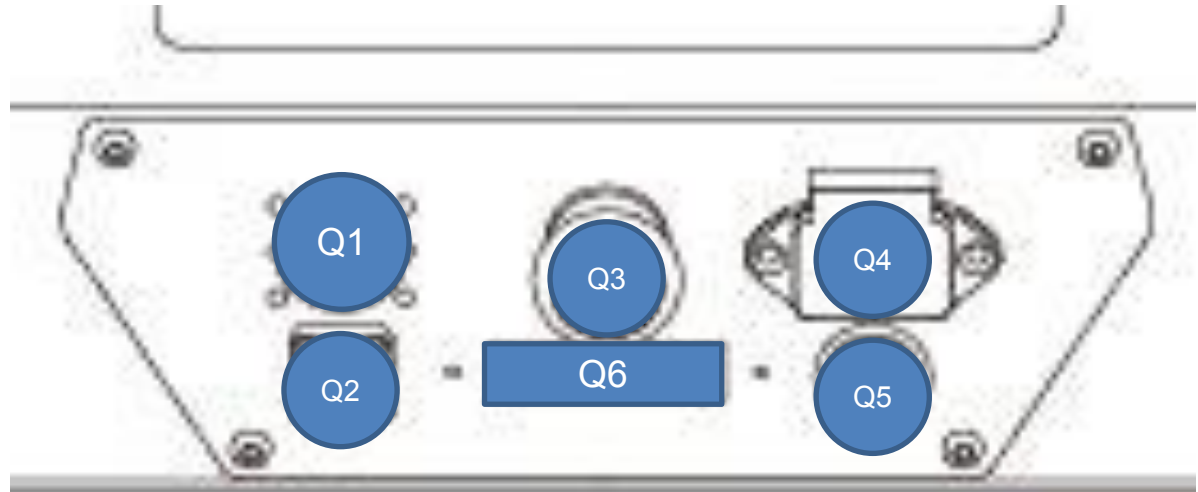
Q2 : CAN_24V Port

Q3 : E-STOP

Q4 : 전원 포트

Q5 : 전원 스위치

Q6 : 전력 LED 패널



01 조종기 작동법

- 전원 On / Off를 위해 7, 8 버튼을 길게 입력(1, 2, 3, 4는 위 올린 상태로 구동)
- 수동 모드 조작을 위해 2를 중앙으로 이동한 후, 5를 이용하여 전, 후방 이동, 6을 이용하여 좌우 회전 조작 가능
- Serial or CAN을 이용한 주행을 위해서는 2를 위(CAN) 또는 아래(Serial)로 이동하면 자율 주행 모드 실행 가능
- 3을 이용하여, 수동 모드 시, TR의 조명을 변경 가능
- 4를 이용하여, TR 의 최대 이동 속도 변경 가능(Speed Mode, Normal Mode)

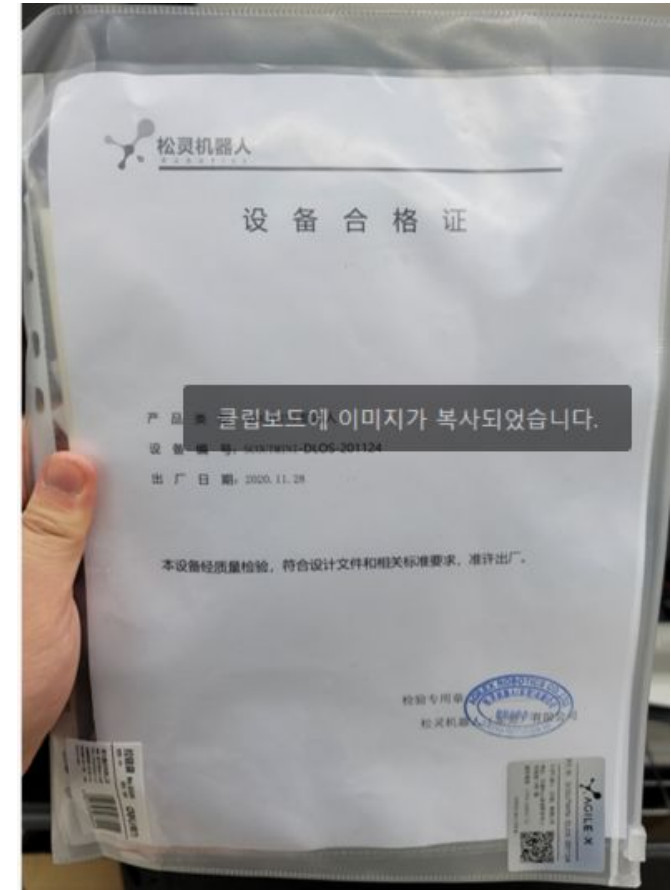


- | | |
|-----------------|---|
| 1. Lever SWA | 7. Power switch key 1 |
| 2. Lever SWB | 8. Power switch key 2 |
| 3. Lever SWC | 9. Mobile/Tablet fixing support interface |
| 4. Lever SWD | 10. Ring interface |
| 5. Left rocker | 11. LCD panel |
| 6. Right rocker | |

*When the user gets the RC transmitter, the settings have been available without having to be set separately.

01 CAN 케이블 사용

1. 전원을 사용하기 앞서 제품과 함께 제공된 오른쪽 사진과 같이 생긴 물건이 있습니다.
2. 해당 패키지 안에는 CAN 어댑터, CAN to USB 젠더, Serial Cable 등 다양한 악세사리들이 들어있습니다.



3. 모든 차량에는 오른쪽 사진과 같이 생긴 포트가 나와있습니다. (특정 차량은 후면 혹은 측면에서 나오는 제품도 있습니다.)
4. 해당 포트는 **CAN통신**과 **24V DC 전원**이 나오는 포트입니다.
5. 패키지에 동봉된 액세서리를 활용하여 결합해줍니다.



6. 선이 총 4개가 나오는데 여기서 노란색과 파란색은 Can to USB 컨버터에 들어가며 노란색이 CAN_H, 파란색이 CAN_L 입니다.
7. 그리고 빨간색이 Vcc(+), 검은색이 GND(-)이며, 24V(오차범위 2.5V)가 나옵니다.
8. 꼭 다른 센서에 사용하실 때 컨버터와 분배기를 활용하여 일정한 출력을 유지시켜 사용하는 걸 권장합니다.



02

Control TR
Using ROS

02 Control TR Using ROS

- Install ROS Package & Build Package

1. `$ mkdir -p ~/catkin_ws/src`
2. `$ cd ~/catkin_ws/src`
3. `$ git clone https://github.com/agilexrobotics/ugv_sdk.git`
4. `$ git clone https://github.com/agilexrobotics/tracer_ros.git`

- Install Dependencies

1. `$ sudo apt install ros-melodic-teleop-twist-keyboard`
2. `$ sudo apt install ros-melodic-joint-state-publisher-gui`
3. `$ sudo apt install ros-melodic-ros-controllers`
4. `$ cd ~/catkin_ws`
5. `$ catkin_make`

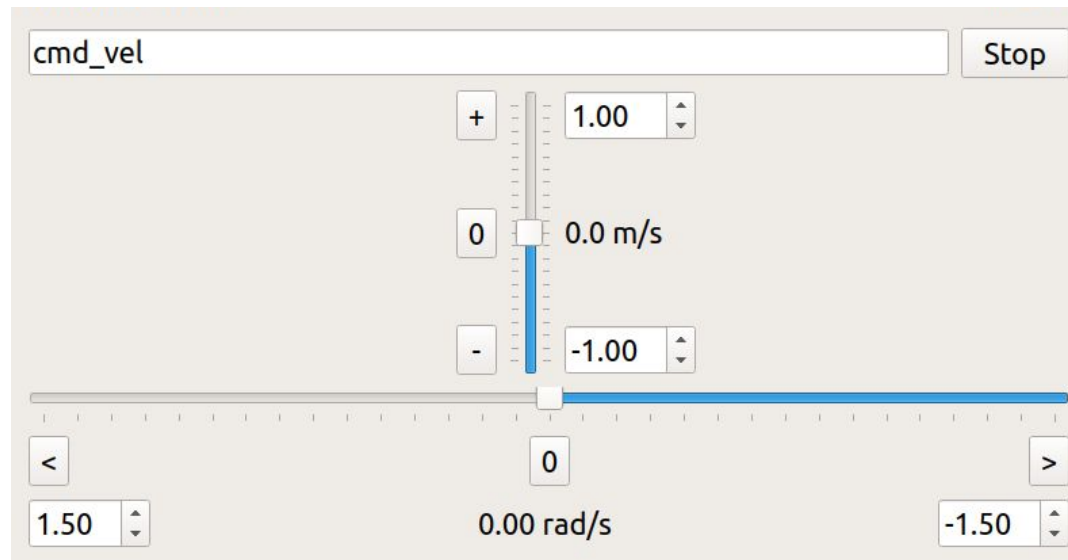
- Setup CAN-To-USB

1. `$ sudo modprobe gs_usb`
2. `$ sudo ip link set can0 up type can bitrate 500000`
3. `$ ifconfig -a` (설정 확인을 위한 부분)
4. `$ sudo apt install can-utils` (최초 실행에만 필요)
5. `$ candump can0` (데이터 입출력 확인을 위해 사용)
6. `$ rosrn tracer_bringup setup_can2usb.bash` (위의 내용을 한 번에 실행)
7. `$ rosrn tracer_bringup bringup_can2usb.bash` (재부팅 or USB 재연결시 실행)

- Start Node
 1. `$ roslaunch tracer_bringup tracer_robot_base.launch (using CAN)`
- 자동 모드 테스트
- 테스트 전 rqt graph 실행하여, driver가 cmd_vel을 subscribe하고 있는지 확인
 - `$ rqt_graph`
- 방법 1 : Robot-Steering 이용
 - `$ sudo apt install ros-melodic-rqt-robot-steering`

02 Control TR Using ROS

- 자동 모드 테스트
- 방법 2 : Robot-Steering 이용
 - \$ sudo apt install ros-melodic-rqt-robot-steering
 - \$ rosrun rqt_robot_steering rqt_robot_steering
- 위의 cmd_vel을 확인 후, 조종기의 모드를 변경한 후, 아래의 값 변경을 통해 제어 가능



02 Control TR Using ROS

- 자동 모드 테스트
- 방법 3 : Keyboard Teleop 이용
- \$ roslaunch tracer_bringup tracer_teleop_keyboard.launch
- 아래의 그림에서 설명을 통해 제어 가능(q, z, w, x, e, c를 통해 속도 변경)
- u, i, o, j, k, l, m, , , .을 통해 이동 방향 및 회전 제어 가능

```
Moving around:
  u   i   o
  j   k   l
  m   ,   .

For Holonomic mode (strafing), hold down the shift key:
-----
  U   I   O
  J   K   L
  M   <   >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5      turn 1.0
█
```


03

TR ROS Topic

- Tracer base Node
- Publishing Topic : /tracer_status, /odom, /tf
- Subscribing Topic : /tracer_light_control, /cmd_vel

- /tracer_status Msg Type (FR = Front Right, RL = Rear Left)
- std_msgs/Header header
 - uint32 seq
 - time stamp
 - string frame_id
- float64 linear_velocity
- float64 angular_velocity
- uint8 base_state
- uint8 control_mode
- uint16 fault_code
- float64 battery_voltage
- tracer_msgs/TracerMotorState[2] motor_states
 - float64 rpm
- bool light_control_enabled
- tracer_msgs/TracerLightState front_light_state
 - uint8 mode
 - uint8 custom_value

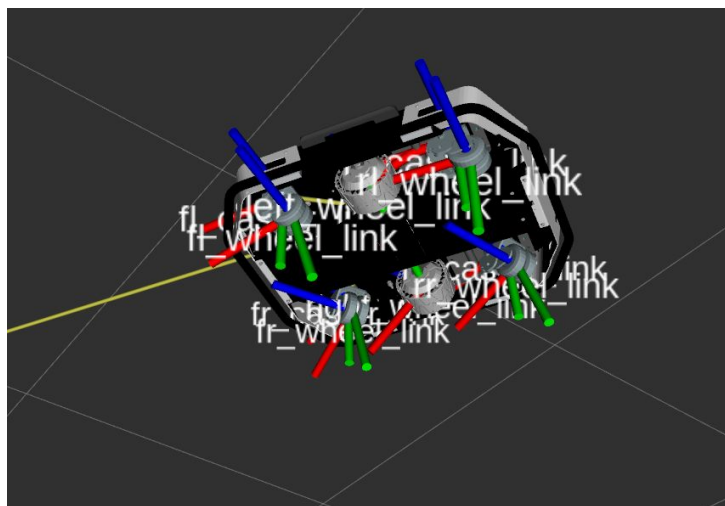
- /tracer_status → 실제 TR의 상태를 출력
- linear_velocity → TR의 속도
- angular_velocity → TR의 회전 속도
- base_state → 사용하지 않음
- control_mode : 0이면 수동, 1이면 CAN, 2이면 RS232
- fault_code : 0이면 정상, 문제 발생 시 다른 숫자
- motor_states : 각 모터의 상태 출력
- light_control_enabled : 자동 모드 시, light 제어 상태 확인 가능
- front_light_state - mode : 0이면 꺼짐, 1이면 고정값, 2이면 숨쉬기 모드

- /odom Msg Type
- std_msgs/Header header
 - uint32 seq
 - time stamp
 - string frame_id
- child_frame_id
- pose
 - pose
 - position
 - x,y,z
 - orientation
 - x,y,z,w
 - covariance
- twist
 - twist
 - linear
 - x,y,z
 - angular
 - x,y,z
 - covariance

- /odom → 실제 TR 가 ROS와 연결된 후, 바퀴를 통해 이동한 현재 위치 및 현재 이동 상태를 확인 가능
- pose : 출발 이후, 변화한 x, y 및 yaw에 해당하는 값 확인 가능
- pose - covariance : 사용하지 않음
- twist - 현재 TR의 선속도 및 회전 속도 확인 가능
- twist - covariance : 사용하지 않음
- rviz를 통해 출발 위치인 odom과 그 child frame인 base_link를 통해, 출발 지점에서 변화한 정도를 확인 가능

- /tf Msg Type
- std_msgs/Header header
 - uint32 seq
 - time stamp
 - string frame_id
- child_frame_id
- transform
 - translation
 - x, y, z
 - rotation
 - x, y, z, w

- /tf → ROS Frame 사이의 변화량에 대한 값 확인 가능
- child_frame_id : 변환할 frame의 이름
- transform : 부모 Frame에서 자식 Frame 사이의 변화량 확인 가능
- transform – translation : 부모 자식 Frame, x, y, z 사이의 변화량 확인 가능
- transform – rotation : 부모 자식 Frame 사이의 roll, pitch, yaw 변화량 확인 가능
- rviz의 tf 및 axes를 통해서 변환된 값을 직접 시각적으로 확인 가능



- /TR_light_control Msg Type
- bool enable_cmd_light_control
- uint8 front_mode
- uint8 front_custom_value
- uint8 rear_mode
- uint8 rear_custom_value

- /TR_light_control → 자동 모드에서 TR의 전방 light control 가능
- enable_cmd_light_control : 제어할지 안할지를 입력 (true, false)
- front_mode : 0일 경우 Off, 1일 경우, 현재 값으로 고정, 2일 경우, 숨쉬기 모드, 3일 경우 아래에서 입력하는 custom value로 설정
- front_custom_value : Mode가 3일 경우 사용하는 Value
- rear_mode – 사용할 수 없음

- /cmd_vel Msg Type
- geometry_msgs/Vector3 linear
 - float64 x
 - float64 y
 - float64 z
- geometry_msgs/Vector3 angular
 - float64 x
 - float64 y
 - float64 z

- /cmd_vel → 자동 모드 시, 속도 및 회전 속도 제어
- linear – x값만 사용하며, m/s단위로 이동 속도 입력
- angular – z값만 사용하며, 회전 속도를 rad/s단위로 입력



Q & A

go.support@wego-robotics.com

go.sales@wego-robotics.com