

LIMO Utils

WeGo & LIMO

목 차

1. 동작 환경
2. Method

01

동작 환경

01 동작 환경

- Github
 - https://github.com/WeGo-Robotics/limo_examples
- 테스트 환경
 - Ubuntu 18.04
 - ROS Melodic
 - Python 2.7 (Default version for Ubuntu 18.04 & ROS Melodic)
 - OpenCV 4.1.1

02

Method

02 Method

- startApplication()
 - Input : 없음
 - Output : 없음
 - Description
 - 해당 모듈 사용을 위해 필수적으로 선행되어야 하는 메서드
 - Processing Procedure
 - 동작 시, Python Script와 ROS가 연결 및 LIMO 제어 준비가 완료됨
 - 정상적으로 수행 시, 'limo application start' 라는 문구가 출력됨
 - 수행 후, 2초간 대기

02 Method

- goStraight(_linear_speed=0.2)
 - Input : _linear_speed (Float)→ LIMO의 선속도 (m/s), Default는 0.2
 - Output : 정상 완료 시, True, 실패 시 False
 - Description
 - LIMO 전, 후진을 담당하는 메서드
 - Processing Procedure
 - 입력받은 _linear_speed (m/s)에 맞춰 전진 또는 후진
 - 지속적인 입력을 하지 않도록 0.1초의 대기 기간 있음
 - 중간에 문제 없이 성공 시 True
 - 오류 발생 시, False

02 Method

- `turn(_angular_speed=0.5, _unit="rad")`
 - Input
 - `_angular_speed` (Float) → LIMO의 회전속도, Default는 0.5 rad/s
 - `_unit` (String) → `_angular_speed`의 단위 명시, “rad” or “deg” 중 하나로 설정
 - Output : 정상 완료 시, True, 실패 시 False
 - Description
 - LIMO 회전을 담당하는 메서드 (Ackermann Mode에서는 조향 담당)
 - Processing Procedure
 - 입력받은 `_angular_speed` 에 맞춰 회전 or 조향
 - 지속적인 입력을 하지 않도록 0.1초의 대기 기간 있음
 - 중간에 문제 없이 성공 시 True, 오류 발생 시, False

02 Method

- `moveCmd(_linear_speed=0.2, _angular_speed=0.5, _unit="rad")`
 - Input
 - `_linear_speed` (Float) → LIMO의 선속도 (m/s)
 - `_angular_speed` (Float) → LIMO의 회전속도, Default는 0.5 rad/s
 - `_unit` (String) → `_angular_speed`의 단위 명시, “rad” or “deg” 중 하나로 설정
 - Output : 정상 완료 시, True, 실패 시 False
 - Description
 - LIMO 전후진 및 회전을 동시에 실행하기 위한 메서드
 - Processing Procedure
 - 입력받은 `_linear_speed`, `_angular_speed` 에 맞춰 전진 or 후진 및 회전 or 조향
 - 지속적인 입력을 하지 않도록 0.1초의 대기 기간 있음
 - 중간에 문제 없이 성공 시 True, 오류 발생 시, False

02 Method

- `getRosTopicData(_topic_name, _msg_type)`
 - Input
 - `_topic_name (String)` → 받으려고 하는 Topic 이름
 - `_msg_type (String)` → 받으려고 하는 Topic의 메시지 타입
 - Output : 정상 완료 시, Topic Data, 3초 이상 데이터가 없을 경우 None
 - Description
 - ROS Topic Data를 받기 위한 메서드
 - Processing Procedure
 - 3초 간 입력한 메시지 타입을 가진 Topic을 받기를 시도
 - 성공하면, 해당 데이터를 반환, 3초 이상 데이터가 없을 경우 None을 반환

02 Method

- getLidar()
 - Input : 없음
 - Output : 정상 완료 시, LiDAR Data, 3초 이상 데이터가 없을 경우 None
 - Description
 - LIMO LiDAR 데이터를 받기 위한 메서드
 - Processing Procedure
 - 3초 간 LaserScan Type의 “/scan” 토픽을 받기를 시도
 - 성공하면, LiDAR 데이터를 반환, 3초 이상 데이터가 없을 경우 None을 반환

02 Method

- `getLidarRangeUsingAngle(_lidar, _angle=0.0, _type="deg")`
 - Input
 - `_lidar` (LaserScan) → `getLidar()` 메서드를 통해 받은 데이터
 - `_angle` (Float) → 거리 값이 필요한 특정 각도
 - `_type` (String) → “deg” or “rad” 입력한 `_angle`의 단위, Default는 “deg”
 - Output : 정상 완료 시, (Angle, Range) 반환, 물체가 너무 가깝거나 멀면 (`_angle`, 0.0)
 - Description
 - LiDAR의 특정 각도의 거리값을 받기 위한 메서드
 - Processing Procedure
 - 입력된 각도를 degree 단위로 변환
 - 입력된 각도 양쪽으로 1.5도를 탐색하여, 값이 있으면 해당 각도와 거리를 반환
 - 값이 없으면 입력한 `_angle`과 0.0을 함께 반환

02 Method

- getImage(_type="opencv")
 - Input
 - _type (String) → “opencv” 일 경우, 출력 데이터를 opencv Image 타입으로, “ROS”인 경우, 출력 데이터를 ROS CompressedImage 타입으로 반환, Default는 “opencv”
 - Output : 정상 완료 시, Image Data, 3초 이상 데이터가 없을 경우 None
 - Description
 - LIMO Camera 데이터를 받기 위한 메서드
 - Processing Procedure
 - 3초 간 CompressedImage Type의 “/camera/rgb/image_raw/compressed” 토픽을 받기를 시도
 - 성공하면, Image 데이터를 반환, 3초 이상 데이터가 없을 경우 None을 반환

02 Method

- `showOpenCvImage(_image, _window_name="myImage")`
 - Input
 - `_image` (OpenCV Image) → 확인하려는 이미지
 - `_window_name` (String) → 확인하는 창의 이름
 - Output : 없음
 - Description
 - OpenCV Image를 확인하기 위한 메서드
 - Processing Procedure
 - `_window_name` 이름의 창을 생성
 - 생성된 창에 `_image`를 띄우기



WeGo Robotics

Tel. 031 – 229 – 3553

Fax. 031 – 229 – 3554



제품 문의: go.sales@wego-robotics.com

기술 문의: go.support@wego-robotics.com