# Turtlebot3

# 이동

# 배치파일 확인

- 배치파일 확인

```
$ code ~/.bashrc
```

```
export ROS_DOMAIN_ID=30
export TURTLEBOT3_MODEL=waffle_pi
export LDS_MODEL=LDS-02
source /opt/ros/foxy/setup.bash
source ~/Workspaces/ros2_ws/install/setup.bash
```

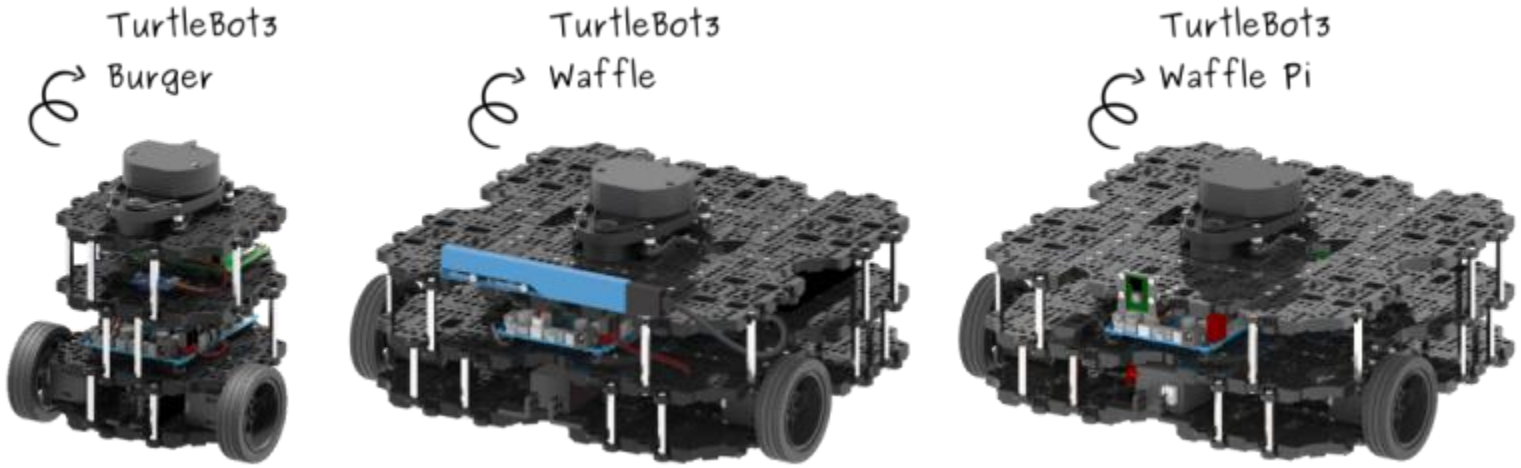
LDS-01 

LDS-02 

- 배치파일 적용

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

# Turtlebot3 Model



# Turtlebot3

# 이동

# CLI 실습

[teleop_turtle]

Publisher
메시지 발행

[/turtle1/cmd_vel]

geometry_msgs/msg/Twist
메시지


[turtlesim]

Subscriber
메시지 구독

**[geometry_msgs/msgs/Twist]**

Vector3 linear
Vector3 angular

**[geometry_msgs/msgs/Vector3]**

float64 x
float64 y
float64 z

**[geometry_msgs/msgs/Vector3]**

float64 x
float64 y
float64 z

출처: https://velog.io/@dumok

# 토픽 /cmd_vel

로봇의 병진 및 회전 속도 제어
명령을 내릴 때 사용하는 토픽

- 토픽 리스트 확인

```
$ ros2 topic list
```

- 토픽 타입 확인

```
$ ros2 topic type /cmd_vel
```

- 인터페이스 확인

```
$ ros2 interface show geometry_msgs/msg/Twist
```

- 인터페이스의 기본 타입 표시

```
$ ros2 interface proto geometry_msgs/msg/Twist
```

# geometry_msgs/msg/Twist

| 변수명 | 단위 | 설명 |
| --- | --- | --- |
| Vector3 **linear** | m/s | x, y, z축 방향으로의 병진속도(linear) |
| Vector3 **angular** | rad/s | x, y, z축에 대한 회전속도(angular) |

# geometry_msgs/msg/Twist

| 변수명 | 단위 | 설명 |
| --- | --- | --- |
| float64 **x** | | x축 |
| float64 **y** | | y축 |
| float64 **z** | | z축 |

https://github.com/ros2/common_interfaces/blob/foxy/geometry_msgs/msg/Twist.msg

# 토픽 /cmd_vel 발행(publish)

- [창1] Turtlebot3 Gazebo 실행

```
$ ros2 launch turtlebot3_gazebo empty_world.launch.py
```

- [창2] 토픽 정보 확인

```
$ ros2 topic info /cmd_vel
```

- [창3] 토픽 내용 확인

```
$ ros2 topic echo /cmd_vel
```

- [창4] 토픽 발행

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}' --once
```

# Turtlebot3 이동

- [창4] 토픽 발행 - 앞으로

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear: {x: 0.2, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}' --once
```

- [창4] 토픽 발행 - 뒤으로

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear: {x: -0.2, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}' --once
```

- [창4] 토픽 발행 - 정지

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}' --once
```

# Turtlebot3 회전

- [창4] 토픽 발행 - 왼쪽으로

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear:
{x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.2}}' --once
```

- [창4] 토픽 발행 - 오른쪽으로

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear:
{x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: -0.2}}' --once
```

- [창4] 토픽 발행 - 정지

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist '{linear:
{x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}' --once
```

# Turtlebot3

# 이동

# Code 실습

# move_pkg

- 패키지 생성

```
$ cd ~/Workspaces/ros2_ws/src
$ ros2 pkg create move_pkg --build-type ament_python
  --dependencies rclpy geometry_msgs
```

- 코드 생성

```
$ cd move_pkg/move_pkg
$ code go.py
$ code stop.py
```

- 설정파일 수정

```
$ cd ..
$ code setup.py
```

# Turtlebot 앞으로 가기

src/move_pkg/move_pkg/go.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist
#https://github.com/ros2/common_interfaces/blob/foxy/geometry_msgs/msg/Twist.msg

class Go(Node):

    def __init__(self):
        super().__init__('go_node')
        self.counter = 0
        self.pub = self.create_publisher(Twist, 'cmd_vel', 10)
        self.timer = self.create_timer(0.1, self.pub_cb)

    def pub_cb(self):
        msg = Twist()
        msg.linear.x = 0.2

        self.pub.publish(msg)
        self.get_logger().info('Published message: ' + str(msg.linear.x))
```

# Turtlebot 앞으로 가기 src/move_pkg/move_pkg/go.py

```
def main(args=None):
    rclpy.init(args=args)
    node = Go()

    try:
        rclpy.spin_once(node)
    except KeyboardInterrupt:
        node.get_logger().info('Keyboard Interrupt')
    finally:
        node.destroy_node()
        rclpy.shutdown()


if __name__ == '__main__':
    main()
```

# Turtlebot 멈추기

src/move_pkg/move_pkg/stop.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist
#https://github.com/ros2/common_interfaces/blob/foxy/geometry_msgs/msg/Twist.msg

class Stop(Node):

    def __init__(self):
        super().__init__('stop_node')
        self.counter = 0
        self.pub = self.create_publisher(Twist, 'cmd_vel', 10)
        self.timer = self.create_timer(0.1, self.pub_cb)

    def pub_cb(self):
        msg = Twist()
        msg.linear.x = 0.0

        self.pub.publish(msg)
        self.get_logger().info('Published message: ' + str(msg.linear.x))
```

# Turtlebot 멈추기

src/move_pkg/move_pkg/stop.py

```
def main(args=None):
    rclpy.init(args=args)
    node = Stop()

    try:
        rclpy.spin_once(node)
    except KeyboardInterrupt:
        node.get_logger().info('Keyboard Interrupt')
    finally:
        node.destroy_node()
        rclpy.shutdown()


if __name__ == '__main__':
    main()
```

# 설정파일 수정

src/move_pkg/setup.py

```
'console_scripts': [
    'go = move_pkg.go:main',
    'stop = move_pkg.stop:main'
],
```

빌드

```
$ cd ~/Workspaces/ros2_ws
$ colcon build --symlink-install --packages-select move_pkg
$ source install/setup.bash
```

노드 실행(ros2 run)

```
$ ros2 pkg executables move_pkg
$ ros2 run move_pkg go
$ ros2 run move_pkg stop
```

# 빌드 & 실행

- 빌드

```
$ cd ~/Workspaces/ros2_ws
$ colcon build --symlink-install --packages-select
move_pkg
```

- 설정 적용

```
$ source install/setup.bash
```

- 패키지 확인

```
$ ros2 pkg executables move_pkg
```

- 코드 실행

```
$ ros2 run move_pkg go
$ ros2 run move_pkg stop
```