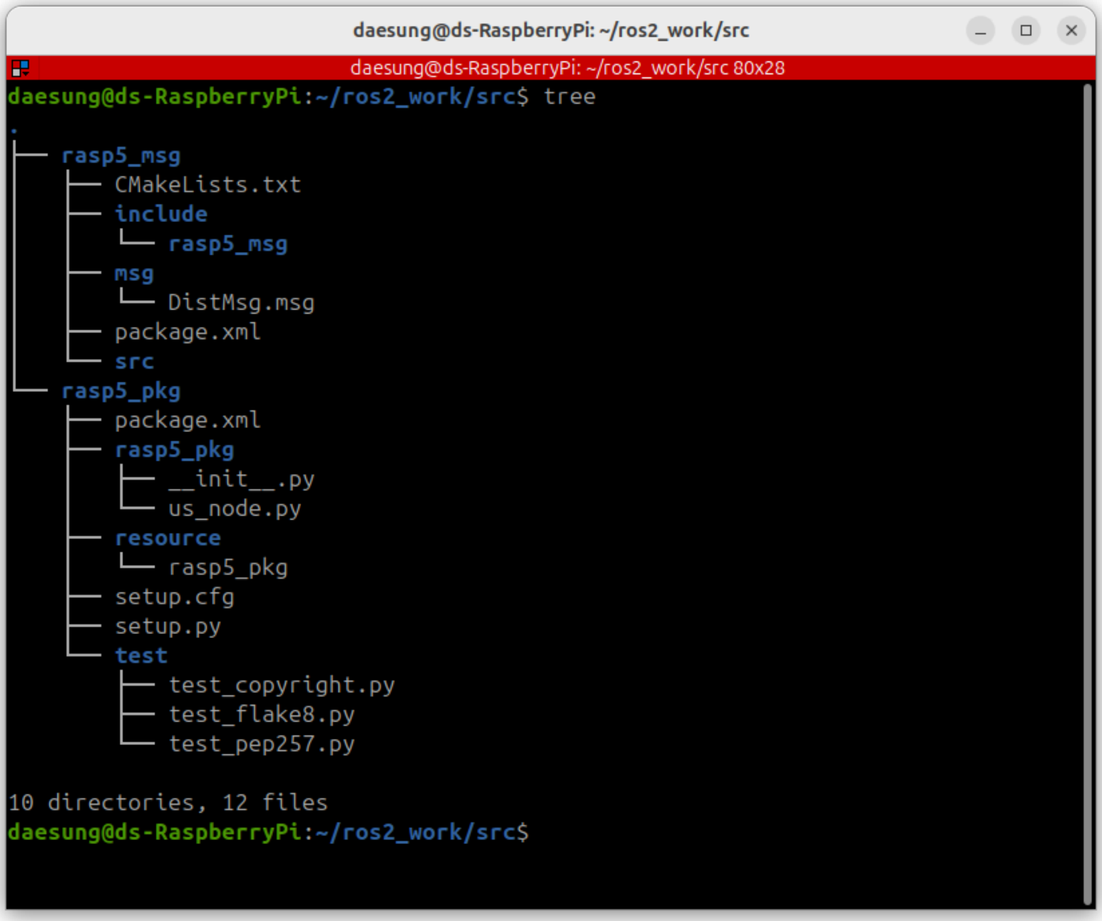
UntraSonic Sensor Pub. node in Raspberry Pi 5

▼ Raspberry Pi ROS2



daesung@ds-RaspberryPi: ~/ros2_work/src
daesung@ds-RaspberryPi: ~/ros2_work/src 80x28

```
daesung@ds-RaspberryPi:~/ros2_work/src$ tree
.
├── rasp5_msg
│   ├── CMakeLists.txt
│   ├── include
│   │   └── rasp5_msg
│   ├── msg
│   │   └── DistMsg.msg
│   ├── package.xml
│   └── src
└── rasp5_pkg
    ├── package.xml
    ├── rasp5_pkg
    │   ├── __init__.py
    │   └── us_node.py
    ├── resource
    │   └── rasp5_pkg
    ├── setup.cfg
    ├── setup.py
    └── test
        ├── test_copyright.py
        ├── test_flake8.py
        └── test_pep257.py

10 directories, 12 files
daesung@ds-RaspberryPi:~/ros2_work/src$
```

- DistMsg.msg

```
float32 dist
```

- CMakeLists.txt

```
cmake_minimum_required(VERSION 3.8)
project(rasp5_msg)
```

```
if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
  add_compile_options(-Wall -Wextra -Wpedantic)
endif()

# find dependencies
find_package(ament_cmake REQUIRED)
# uncomment the following section in order to fill in
# further dependencies manually.
# find_package(<dependency> REQUIRED)

# Edited from here
find_package(rosidl_default_generators REQUIRED)
rosidl_generate_interfaces(${PROJECT_NAME} "msg/DistMsg.msg")
# Edited to here

if(BUILD_TESTING)
    find_package(ament_lint_auto REQUIRED)
    # the following line skips the linter which checks for copyrights
    # comment the line when a copyright and license is added to all source files
    set(ament_cmake_copyright_FOUND TRUE)
    # the following line skips cpplint (only works in a git repo)
    # comment the line when this package is in a git repo and when
    # a copyright and license is added to all source files
    set(ament_cmake_cpplint_FOUND TRUE)
    ament_lint_auto_find_test_dependencies()
endif()

ament_package()
```

- package.xml

```
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
    <name>rasp5_msg</name>
    <version>0.0.0</version>
    <description>TODO: Package description</description>
    <maintainer email="daesung@todo.todo">daesung</maintainer>
    <license>TODO: License declaration</license>

  <buildtool_depend>ament_cmake</buildtool_depend>

  <test_depend>ament_lint_auto</test_depend>
  <test_depend>ament_lint_common</test_depend>

  <build_depend>rosidl_default_generators</build_depend>
  <exec_depend>rosidl_default_runtime</exec_depend>
  <member_of_group>rosidl_interface_packages</member_of_group>

  <export>
      <build_type>ament_cmake</build_type>
    </export>
</package>
```

- us_node.py

```
import rclpy as rp
from rclpy.node import Node
from rasp5_msg.msg import DistMsg

from gpiozero import DistanceSensor
from time import sleep

class UsDistPublisher(Node):
    def __init__(self):
        super().__init__('us_dist_publisher')
        self.publisher_ = self.create_publisher(DistMsg, 'us_dist', 10)
        timer_period = 0.5  # seconds
        self.timer = self.create_timer(timer_period, self.timer_callback)
        self.sensor = DistanceSensor(echo=24, trigger=23)
        self.get_logger().info('US Distance Publisher Node has been starte
d.')
```

```
    def timer_callback(self):
        msg = DistMsg()
        distance_cm = self.sensor.distance * 100
        msg.dist = distance_cm  # Assuming 'dist' is the field in your DistM
sg.msg
        self.publisher_.publish(msg)
        self.get_logger().info(f'Publishing: "{msg.dist:.2f} cm"')


def main(args=None):
    rp.init(args=args)
    us_dist_publisher = UsDistPublisher()
    rp.spin(us_dist_publisher)
    us_dist_publisher.destroy_node()
    rp.shutdown()


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

- setup.py

```
from setuptools import find_packages, setup

package_name = 'rasp5_pkg'

setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='daesung',
    maintainer_email='daesung@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    extras_require={
        'test': [
            'pytest',
        ],
    },
    entry_points={
        'console_scripts': [
            'us_node = rasp5_pkg.us_node:main'
        ],
    },
)
```