

Project PacMan

1.0.0

Generated by Doxygen 1.9.1

1 PacMan - Swarm Robotics Algorithm	1
1.1 Project Descripton	1
1.1.1 Overview	1
1.1.2 Method	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Master Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 Master()	10
5.1.3 Member Function Documentation	10
5.1.3.1 process_callback()	10
5.1.3.2 traj()	10
5.2 RoomBa Class Reference	11
5.2.1 Detailed Description	11
5.2.2 Constructor & Destructor Documentation	12
5.2.2.1 RoomBa()	12
5.2.3 Member Function Documentation	12
5.2.3.1 set_goal()	12
5.2.3.2 stop()	12
5.3 TestPubSub Class Reference	13
5.3.1 Detailed Description	13
5.3.2 Member Data Documentation	13
5.3.2.1 node_	14
5.4 Trajectory Class Reference	14
5.4.1 Detailed Description	14
5.4.2 Constructor & Destructor Documentation	14
5.4.2.1 Trajectory()	14
5.4.3 Member Function Documentation	15
5.4.3.1 circleTrajectory()	15
5.4.3.2 getNumberOfRobots()	15
5.4.3.3 setCenter()	15
5.4.3.4 setNumberOfRobots()	15
5.4.3.5 squareTrajectory()	17
5.4.3.6 triangleTrajectory()	17

6 File Documentation	19
6.1 docs/introduction.txt File Reference	19
6.2 include/master.hpp File Reference	19
6.2.1 Detailed Description	20
6.2.2 Typedef Documentation	20
6.2.2.1 ODOM	20
6.2.2.2 TWIST	20
6.3 include/pubsub.hpp File Reference	21
6.3.1 Detailed Description	22
6.3.2 Typedef Documentation	22
6.3.2.1 ODOM	22
6.3.2.2 TWIST	22
6.4 include/trajectory.hpp File Reference	23
6.4.1 Detailed Description	23
6.5 src/main.cpp File Reference	24
6.5.1 Detailed Description	24
6.5.2 Function Documentation	25
6.5.2.1 main()	25
6.6 test/main.cpp File Reference	25
6.6.1 Detailed Description	25
6.6.2 Function Documentation	26
6.6.2.1 main()	26
6.7 src/master.cpp File Reference	26
6.7.1 Detailed Description	27
6.7.2 Typedef Documentation	27
6.7.2.1 ODOM	27
6.7.2.2 RCL_NODE_PTR	27
6.7.2.3 TWIST	27
6.8 src/pubsub.cpp File Reference	28
6.8.1 Detailed Description	28
6.9 src/trajectory.cpp File Reference	28
6.9.1 Detailed Description	29
6.10 test/pubsub_test.cpp File Reference	29
6.10.1 Detailed Description	30
6.10.2 Function Documentation	30
6.10.2.1 TEST_F() [1/2]	31
6.10.2.2 TEST_F() [2/2]	31
6.11 test/test.cpp File Reference	31
6.11.1 Detailed Description	32
6.11.2 Function Documentation	32
6.11.2.1 TEST() [1/3]	32
6.11.2.2 TEST() [2/3]	32

6.11.2.3 TEST() [3/3]	32
Index	33

Chapter 1

PacMan - Swarm Robotics Algorithm

1.1 Project Descripton

1.1.1 Overview

The aim of this project is to design and simulate a swarm robotics algorithm for Acme Robotics. The simulation is created in ROS2 Gazebo. The Turtlebot3 burger model is used to simulate our agents. Our algorithm sets a target location, and then directs 25 agents to surround the target based on the selected formation. The swarm can also transition between the different formations.

1.1.2 Method

The simulation is started and the agents are spawned into the gazebo world by running a python launch file. We set up a class to initialize nodes for each swarm agent to publish to its command velocity and subscribe to the position provided through Odometry. Next, the master class will ask the user to input desired formation, and then the trajectory will be computed by the methods in the trajectory class. The trajectory class computes goal positions for each agent based on the set target location. The goal location and live position of each agent is then used to direct it towards its goal location. A two sprint Agile Iterative Process approach and test driven development style is utilized in the making of this project.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

rclcpp::Node	
Master	9
RoomBa	11
testing::Test	
TestPubSub	13
Trajectory	14

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Master	Master class to control the robots and spawn nodes	9
RoomBa	Class for the pubsub node	11
TestPubSub	Test fixture class for the project	13
Trajectory	Class for the trajectory	14

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/ master.hpp	
Master class to control the robots	19
include/ pubsub.hpp	
Header file for the pubsub class	21
include/ trajectory.hpp	
Header file for the trajectory class	23
src/ main.cpp	
Main file for the project which creates the nodes	24
src/ master.cpp	
Master class for the project which creates a master node and executes the trajectory algorithm	26
src/ pubsub.cpp	
RoomBa class for the project which creates a node and publishes and subscribes to the required topics based on the robot number and namespace	28
src/ trajectory.cpp	
Trajectory class for the project which creates a shape trajectory for the robots to follow	28
test/ main.cpp	
Main file for the project which runs the tests	25
test/ pubsub_test.cpp	
Test file for the project which tests the publisher and subscriber	29
test/ test.cpp	
Test file for the project to test the algorithms	31

Chapter 5

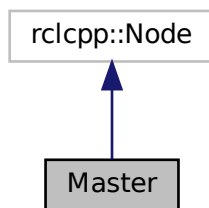
Class Documentation

5.1 Master Class Reference

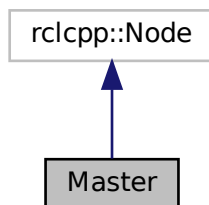
[Master](#) class to control the robots and spawn nodes.

```
#include <master.hpp>
```

Inheritance diagram for Master:



Collaboration diagram for Master:



Public Member Functions

- [Master](#) (std::vector< std::shared_ptr< [RoomBa](#) >> const &robot_array)
Construct a new [Master](#) object.
- void [process_callback](#) ()
Empty function to be called by publisher.
- void [traj](#) ()
Function to select the robot trajectory to be followed and call the respective function.

5.1.1 Detailed Description

[Master](#) class to control the robots and spawn nodes.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Master()

```
Master::Master (
    std::vector< std::shared_ptr< RoomBa >> const & robot_array ) [explicit]
```

Construct a new [Master](#) object.

Parameters

<i>robot_array</i>	Vector of robot objects
--------------------	-------------------------

5.1.3 Member Function Documentation

5.1.3.1 process_callback()

```
void Master::process_callback ( )
```

Empty function to be called by publisher.

5.1.3.2 traj()

```
void Master::traj ( )
```

Function to select the robot trajectory to be followed and call the respective function.

The documentation for this class was generated from the following files:

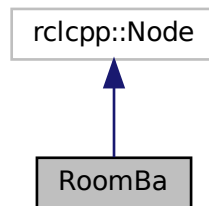
- include/[master.hpp](#)
- src/[master.cpp](#)

5.2 RoomBa Class Reference

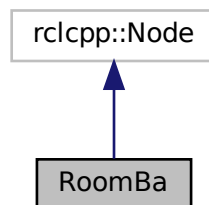
Class for the pubsub node.

```
#include <pubsub.hpp>
```

Inheritance diagram for RoomBa:



Collaboration diagram for RoomBa:



Public Member Functions

- [RoomBa](#) (std::string const &node_name, std::string const &robot_name)
Construct a new [RoomBa](#) object.
- void [set_goal](#) (double x, double y)
Set the goal object.
- void [stop](#) ()
Stop the robot by publishing 0 velocity.

5.2.1 Detailed Description

Class for the pubsub node.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 RoomBa()

```
RoomBa::RoomBa (
    std::string const & node_name,
    std::string const & robot_name ) [inline]
```

Construct a new [RoomBa](#) object.

Parameters

<i>node_name</i>	name of the node
<i>robot_name</i>	name of the robot

5.2.3 Member Function Documentation

5.2.3.1 set_goal()

```
void RoomBa::set_goal (
    double x,
    double y ) [inline]
```

Set the goal object.

Parameters

<i>x</i>	x coordinate of the goal
<i>y</i>	y coordinate of the goal

5.2.3.2 stop()

```
void RoomBa::stop ( )
```

Stop the robot by publishing 0 velocity.

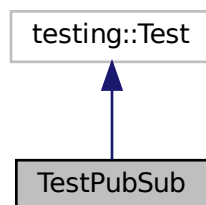
The documentation for this class was generated from the following files:

- [include/pubsub.hpp](#)
- [src/pubsub.cpp](#)

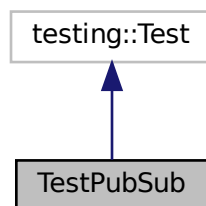
5.3 TestPubSub Class Reference

Test fixture class for the project.

Inheritance diagram for TestPubSub:



Collaboration diagram for TestPubSub:



Protected Attributes

- `roclcpp::Node::SharedPtr` [node_](#)

5.3.1 Detailed Description

Test fixture class for the project.

5.3.2 Member Data Documentation

5.3.2.1 node_

```
rc1cpp::Node::SharedPtr TestPubSub::node_ [protected]
```

The documentation for this class was generated from the following file:

- [test/pubsub_test.cpp](#)

5.4 Trajectory Class Reference

Class for the trajectory.

```
#include <trajectory.hpp>
```

Public Member Functions

- [Trajectory](#) ()
Construct a new [Trajectory](#) object and initialize the variables.
- void [setCenter](#) (double x, double y)
Set the Center of the object.
- void [setNumberOfRobots](#) (double n)
Set the Number Of Robots object.
- int [getNumberOfRobots](#) ()
Get the Number Of Robots object.
- std::vector< std::vector< double > > [circleTrajectory](#) ()
Function to calculate the circle shape trajectory.
- std::vector< std::vector< double > > [squareTrajectory](#) ()
Function to calculate the square shape trajectory.
- std::vector< std::vector< double > > [triangleTrajectory](#) ()
Function to calculate the triangle shape trajectory.

5.4.1 Detailed Description

Class for the trajectory.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 Trajectory()

```
Trajectory::Trajectory ( )
```

Construct a new [Trajectory](#) object and initialize the variables.

5.4.3 Member Function Documentation

5.4.3.1 circleTrajectory()

```
vector< vector< double > > Trajectory::circleTrajectory ( )
```

Function to calculate the circle shape trajectory.

Returns

std::vector<std::vector<double>> vector of vectors containing the x and y coordinates of the robots

5.4.3.2 getNumberOfRobots()

```
int Trajectory::getNumberOfRobots ( )
```

Get the Number Of Robots object.

Returns

int number of robots

5.4.3.3 setCenter()

```
void Trajectory::setCenter (
    double x,
    double y )
```

Set the Center of the object.

Parameters

<i>x</i>	x coordinate of the center
<i>y</i>	y coordinate of the center

5.4.3.4 setNumberOfRobots()

```
void Trajectory::setNumberOfRobots (
    double n )
```

Set the Number Of Robots object.

Parameters

n	number of robots
-----	------------------

5.4.3.5 squareTrajectory()

```
vector< vector< double > > Trajectory::squareTrajectory ( )
```

Function to calculate the square shape trajectory.

Returns

`std::vector<std::vector<double>>` vector of vectors containing the x and y coordinates of the robots

5.4.3.6 triangleTrajectory()

```
vector< vector< double > > Trajectory::triangleTrajectory ( )
```

Function to calculate the triangle shape trajectory.

Returns

`std::vector<std::vector<double>>` vector of vectors containing the x and y coordinates of the robots

The documentation for this class was generated from the following files:

- [include/trajectory.hpp](#)
- [src/trajectory.cpp](#)

Chapter 6

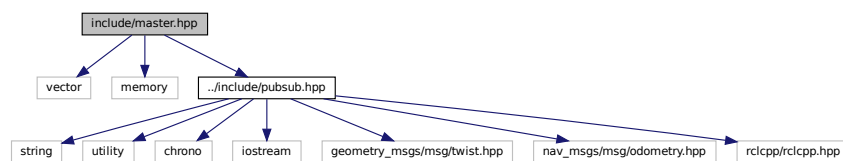
File Documentation

6.1 docs/introduction.txt File Reference

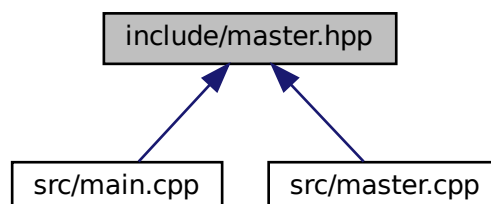
6.2 include/master.hpp File Reference

[Master](#) class to control the robots.

```
#include <vector>
#include <memory>
#include "../include/pubsub.hpp"
Include dependency graph for master.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Master](#)
Master class to control the robots and spawn nodes.

Typedefs

- using [TWIST](#) = geometry_msgs::msg::Twist
- using [ODOM](#) = nav_msgs::msg::Odometry

6.2.1 Detailed Description

[Master](#) class to control the robots.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.2.2 Typedef Documentation

6.2.2.1 ODOM

```
using ODOM = nav_msgs::msg::Odometry
```

6.2.2.2 TWIST

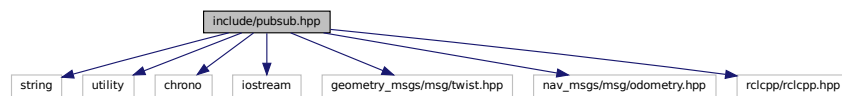
```
using TWIST = geometry_msgs::msg::Twist
```

6.3 include/pubsub.hpp File Reference

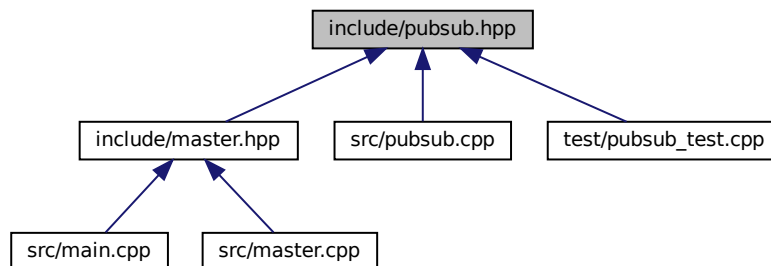
Header file for the pubsub class.

```
#include <string>
#include <utility>
#include <chrono>
#include <iostream>
#include <geometry_msgs/msg/twist.hpp>
#include <nav_msgs/msg/odometry.hpp>
#include <rclcpp/rclcpp.hpp>
```

Include dependency graph for pubsub.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [RoomBa](#)
Class for the pubsub node.

Typedefs

- using [TWIST](#) = geometry_msgs::msg::Twist
- using [ODOM](#) = nav_msgs::msg::Odometry

6.3.1 Detailed Description

Header file for the pubsub class.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.3.2 Typedef Documentation

6.3.2.1 ODOM

```
using ODOM = nav_msgs::msg::Odometry
```

6.3.2.2 TWIST

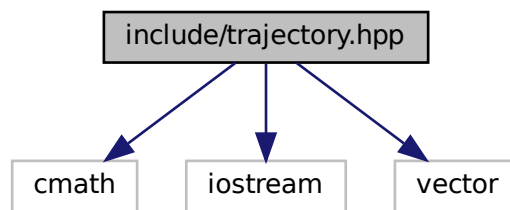
```
using TWIST = geometry_msgs::msg::Twist
```

6.4 include/trajectory.hpp File Reference

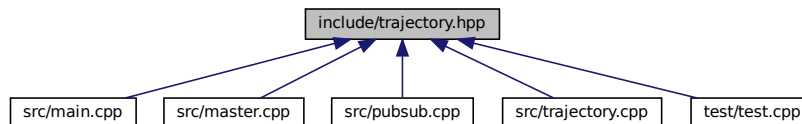
Header file for the trajectory class.

```
#include <cmath>
#include <iostream>
#include <vector>
```

Include dependency graph for trajectory.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Trajectory](#)

Class for the trajectory.

6.4.1 Detailed Description

Header file for the trajectory class.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu.)

Version

0.1

Date

2022-12-15

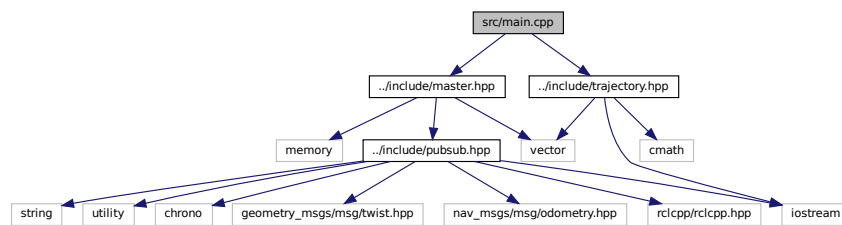
Copyright

Copyright (c) 2022

6.5 src/main.cpp File Reference

Main file for the project which creates the nodes.

```
#include "../include/master.hpp"
#include "../include/trajectory.hpp"
Include dependency graph for main.cpp:
```



Functions

- `int main (int argc, char **argv)`

6.5.1 Detailed Description

Main file for the project which creates the nodes.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.5.2 Function Documentation

6.5.2.1 main()

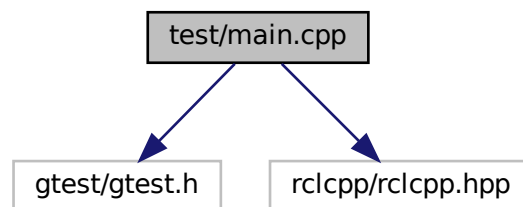
```
int main (
    int argc,
    char ** argv )
```

6.6 test/main.cpp File Reference

Main file for the project which runs the tests.

```
#include <gtest/gtest.h>
#include <rclcpp/rclcpp.hpp>
```

Include dependency graph for main.cpp:



Functions

- int [main](#) (int argc, char **argv)
Main function for the project.

6.6.1 Detailed Description

Main file for the project which runs the tests.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.6.2 Function Documentation

6.6.2.1 main()

```
int main (
    int argc,
    char ** argv )
```

Main function for the project.

Parameters

<i>argc</i>	Number of arguments
<i>argv</i>	Arguments

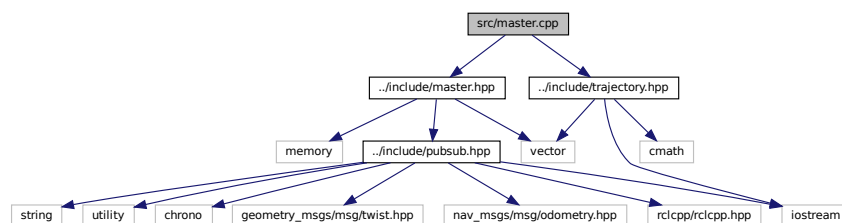
Returns

int Returns result of the tests

6.7 src/master.cpp File Reference

[Master](#) class for the project which creates a master node and executes the trajectory algorithm.

```
#include "../include/master.hpp"
#include "../include/trajectory.hpp"
Include dependency graph for master.cpp:
```



Typedefs

- using [TWIST](#) = geometry_msgs::msg::Twist
- using [ODOM](#) = nav_msgs::msg::Odometry
- using [RCL_NODE_PTR](#) = std::shared_ptr< rclcpp::Node >

6.7.1 Detailed Description

`Master` class for the project which creates a master node and executes the trajectory algorithm.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.7.2 Typedef Documentation

6.7.2.1 ODOM

```
using ODOM = nav_msgs::msg::Odometry
```

6.7.2.2 RCL_NODE_PTR

```
using RCL_NODE_PTR = std::shared_ptr<rclcpp::Node>
```

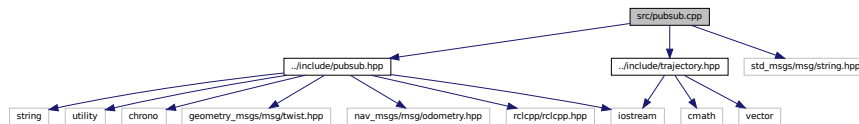
6.7.2.3 TWIST

```
using TWIST = geometry_msgs::msg::Twist
```

6.8 src/pubsub.cpp File Reference

[RoomBa](#) class for the project which creates a node and publishes and subscribes to the required topics based on the robot number and namespace.

```
#include "../include/pubsub.hpp"
#include "../include/trajectory.hpp"
#include <std_msgs/msg/string.hpp>
Include dependency graph for pubsub.cpp:
```



6.8.1 Detailed Description

[RoomBa](#) class for the project which creates a node and publishes and subscribes to the required topics based on the robot number and namespace.

Author

Driver: Tanmay Haldankar (tanmayh@umd.edu) Navigator: Sanchit Kedia (sanchit@terpmail.umd.edu)

Version

0.2

Date

2022-12-15

Copyright

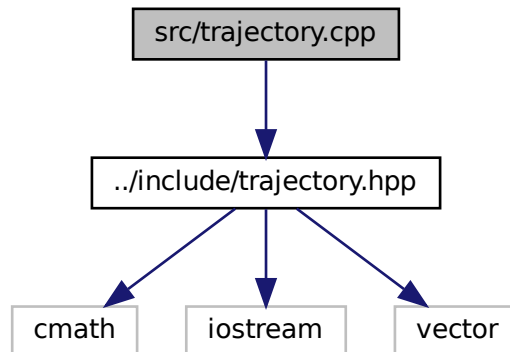
Copyright (c) 2022

6.9 src/trajectory.cpp File Reference

[Trajectory](#) class for the project which creates a shape trajectory for the robots to follow.

```
#include "../include/trajectory.hpp"
```

Include dependency graph for trajectory.cpp:



6.9.1 Detailed Description

`Trajectory` class for the project which creates a shape trajectory for the robots to follow.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

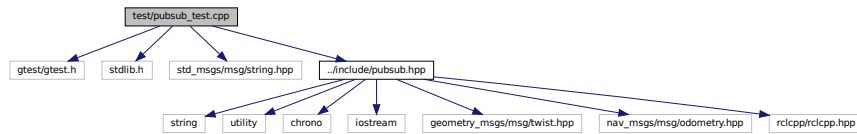
6.10 test/pubsub_test.cpp File Reference

Test file for the project which tests the publisher and subscriber.

```
#include <gtest/gtest.h>
#include <stdlib.h>
#include "std_msgs/msg/string.hpp"
```

```
#include "../include/pubsub.hpp"
```

Include dependency graph for pubsub_test.cpp:



Classes

- class [TestPubSub](#)
Test fixture class for the project.

Functions

- [TEST_F](#) ([TestPubSub](#), test_num_publishers)
Construct a new test f object to test the publisher.
- [TEST_F](#) ([TestPubSub](#), test_num_subscribers)
Construct a new test f object to test the subscriber.

6.10.1 Detailed Description

Test file for the project which tests the publisher and subscriber.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu.)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.10.2 Function Documentation

6.10.2.1 TEST_F() [1/2]

```
TEST_F (
    TestPubSub ,
    test_num_publishers )
```

Construct a new test f object to test the publisher.

6.10.2.2 TEST_F() [2/2]

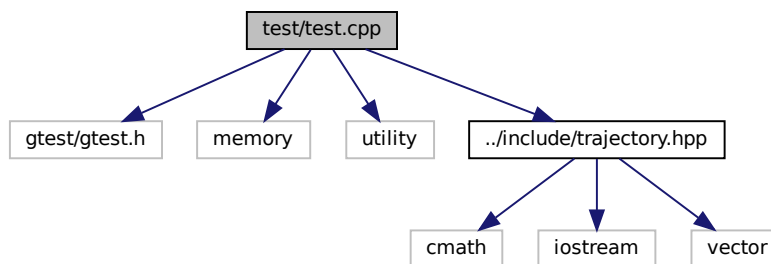
```
TEST_F (
    TestPubSub ,
    test_num_subscribers )
```

Construct a new test f object to test the subscriber.

6.11 test/test.cpp File Reference

Test file for the project to test the algorithms.

```
#include <gtest/gtest.h>
#include <memory>
#include <utility>
#include "../include/trajectory.hpp"
Include dependency graph for test.cpp:
```



Functions

- **TEST** (AlgorithmTest, CircleTest)
Construct a new TEST object for the circle trajectory.
- **TEST** (AlgorithmTest, SquareTest)
Construct a new TEST object for the square trajectory.
- **TEST** (AlgorithmTest, TriangleTest)
Construct a new TEST object for the triangle trajectory.

6.11.1 Detailed Description

Test file for the project to test the algorithms.

Author

Driver: Sanchit Kedia (sanchit@terpmail.umd.edu) Navigator: Tanmay Haldankar (tanmayh@umd.edu)

Version

0.1

Date

2022-12-15

Copyright

Copyright (c) 2022

6.11.2 Function Documentation

6.11.2.1 TEST() [1/3]

```
TEST (
    AlgorithmTest ,
    CircleTest )
```

Construct a new TEST object for the circle trajectory.

6.11.2.2 TEST() [2/3]

```
TEST (
    AlgorithmTest ,
    SquareTest )
```

Construct a new TEST object for the square trajectory.

6.11.2.3 TEST() [3/3]

```
TEST (
    AlgorithmTest ,
    TriangleTest )
```

Construct a new TEST object for the triangle trajectory.

Index

- circleTrajectory
 - Trajectory, [15](#)
- docs/introduction.txt, [19](#)
- getNumberOfRobots
 - Trajectory, [15](#)
- include/master.hpp, [19](#)
- include/pubsub.hpp, [21](#)
- include/trajectory.hpp, [23](#)
- main
 - main.cpp, [25](#), [26](#)
- main.cpp
 - main, [25](#), [26](#)
- Master, [9](#)
 - Master, [10](#)
 - process_callback, [10](#)
 - traj, [10](#)
- master.cpp
 - ODOM, [27](#)
 - RCL_NODE_PTR, [27](#)
 - TWIST, [27](#)
- master.hpp
 - ODOM, [20](#)
 - TWIST, [20](#)
- node_
 - TestPubSub, [13](#)
- ODOM
 - master.cpp, [27](#)
 - master.hpp, [20](#)
 - pubsub.hpp, [22](#)
- process_callback
 - Master, [10](#)
- pubsub.hpp
 - ODOM, [22](#)
 - TWIST, [22](#)
- pubsub_test.cpp
 - TEST_F, [30](#), [31](#)
- RCL_NODE_PTR
 - master.cpp, [27](#)
- RoomBa, [11](#)
 - RoomBa, [12](#)
 - set_goal, [12](#)
 - stop, [12](#)
- set_goal
 - RoomBa, [12](#)
- setCenter
 - Trajectory, [15](#)
- setNumberOfRobots
 - Trajectory, [15](#)
- squareTrajectory
 - Trajectory, [17](#)
- src/main.cpp, [24](#)
- src/master.cpp, [26](#)
- src/pubsub.cpp, [28](#)
- src/trajectory.cpp, [28](#)
- stop
 - RoomBa, [12](#)
- TEST
 - test.cpp, [32](#)
- test.cpp
 - TEST, [32](#)
- test/main.cpp, [25](#)
- test/pubsub_test.cpp, [29](#)
- test/test.cpp, [31](#)
- TEST_F
 - pubsub_test.cpp, [30](#), [31](#)
- TestPubSub, [13](#)
 - node_, [13](#)
- traj
 - Master, [10](#)
- Trajectory, [14](#)
 - circleTrajectory, [15](#)
 - getNumberOfRobots, [15](#)
 - setCenter, [15](#)
 - setNumberOfRobots, [15](#)
 - squareTrajectory, [17](#)
 - Trajectory, [14](#)
 - triangleTrajectory, [17](#)
- triangleTrajectory
 - Trajectory, [17](#)
- TWIST
 - master.cpp, [27](#)
 - master.hpp, [20](#)
 - pubsub.hpp, [22](#)