

# チャレンジサイト・メカニックカモノハシ 2019

## micro mouse simulator API

ER17045 立道壱太郎

2019 年 8 月 21 日

### Mouse

Mouse( topic\_device, topic\_sensor)

mouse のコンストラクタです。

- topic\_device mouse が配信する Twist 型のトピック名を指定します。初期設定のような物なので、これは"cmd\_vel"と指定してください。
- topic\_sensor mouse が購読する float64 型のトピック名を指定します。初期設定のような物なので、これはサンプル通りに指定してください。

move( fwd, ang, [cnt])

シミュレータ上の mouse を動かします。

- fwd  
直進速度 [m/s] を指定します。
- ang  
回転速度 [rad/s] を指定します。
- [cnt]  
実行速度みたいな物です。デフォルトで 10 に指定されているので、指定しない場合は 10 になります。基本的に指定しないでください。

### サンプルコード

ソースコード 1 script/chapters/1/moving\_mouse.py

---

```
1 #!/usr/bin/env python
2 #coding: utf-8 # 日本語を使えるようにする
3
4 import roslib.packages
5 pk_path = roslib.packages.get_pkg_dir('micro_mouse')
6 import sys
```

```

7 sys.path.append(pk_path + '/script/lib')
8
9 import math
10 import rospy # include<ros/ros.h>のようなもの
11
12 import mouse
13
14
15 if __name__ == '__main__': # int main()みたいな
16     rospy.init_node("node_name") # ノード設定
17
18     mouse = mouse.Mouse( "cmd_vel", ["scan/R",
19                                     "scan/FR",
20                                     "scan/F",
21                                     "scan/FL",
22                                     "scan/L"])
23
24     try:
25         loop = rospy.Rate(10) # 10[loop/s]の設定をする。
26         while not rospy.is_shutdown():
27             #-----ここにプログラムを書く-----
28             #-----ロボットを動かす-----
29             # mouse.move( 直進速度 [ m/s], 回転速度 [ rad/s])
30             # 直進速度は 1[m/s]、回転速度は /2 [ rad/s]が限界です。
31             print("-----")
32             print("マウス動きます。")
33             print("マウス 1 マス前進します。")
34             mouse.move( 1, 0.0)
35             print("マウス 9 0 度回転します。")
36             mouse.move(0.0, 1 * math.pi/2)
37             print("マウス 1 マス前進します。")
38             mouse.move( 1, 0.0)
39             print("マウス 1 マス後退します。")
40             mouse.move( -1, 0.0)
41             print("マウス - 9 0 度回転します。")
42             mouse.move(0.0, -1 * math.pi/2)
43             print("マウス 1 マス後退します。")
44             mouse.move( -1, 0.0)
45
46             #-----ここまで-----
47             loop.sleep() # 10[loop/s]になるよう調整する。
48         except KeyboardInterrupt: # 実行中 (try:中)にCTRL-C が押されればプログラムが終了する
49             print("キーボード割り込み！CTRL-C 終了")

```

---

## wallPublisher

### setData( data)

- data

2次元配列変数を渡してください。値によって Rviz 上で壁の色が変化します。  
値と色の対応表は以下の通りです。

0	透明	walls.WALL_INVISIBLE
1	白	walls.WALL_WHITE
2	赤	walls.WALL_RED
3	オレンジ	walls.WALL_ORANGE
4	黄色	walls.WALL_YELLOW
5	緑	walls.WALL_GREEN
6	青	walls.WALL_BLUE
7	紫	walls.WALL_PURPLE
8	灰色	walls.WALL_GREY

### publish()

Rviz 上の壁を更新します。

## サンプルコード

### ソースコード 2 script/chapters/2/wall\_publisher.py

```
1  #!/usr/bin/env python
2  #coding: utf-8 # 日本語を使えるようにする
3
4  import roslib.packages
5  pk_path = roslib.packages.get_pkg_dir('micro_mouse')
6  import sys
7  sys.path.append(pk_path + '/script/lib')
8
9  import math
10 import numpy as np
11 import rospy # include<ros/ros.h>のようなもの
12
13 import wall
14
15
16 if __name__ == '__main__': # int main()みたいな
17     rospy.init_node("node_name") # ノード設定
18
19     cell_matrix = [ 11, 11]
20     cell_size = [ 0.200, 0.200]
21     walls = wall.wallPublisher("wall_marker", cell_matrix, cell_size)
22     map_data = np.zeros( cell_matrix, dtype=int)
```

```

23 map_data_colored = np.zeros( cell_matrix, dtype=int)
24
25 try:
26     loop = rospy.Rate(10) # 10[loop/s]の設定をする。
27     while not rospy.is_shutdown():
28         #-----ここにプログラムを書く-----
29         #-----地図情報を更新する-----
30         #-----壁更新-----
31         for iy in range(5):
32             for ix in range(5):
33                 map_data[ iy*2+0, ix*2+0] = 1 # [ 偶数, 偶数]
34                 map_data[ iy*2+1, ix*2+1] = 2 # [ 奇数, 奇数]
35                 map_data[ iy*2+0, ix*2+1] = 3 # [ 偶数, 奇数]
36                 map_data[ iy*2+1, ix*2+0] = 4 # [ 奇数, 偶数]
37         #-----マップ色変え & マップ送信-----
38         map_data_colored[ map_data == 0] = walls.WALL_INVISIBLE
39         map_data_colored[ map_data == 1] = walls.WALL_BLUE
40         map_data_colored[ map_data == 2] = walls.WALL_WHITE
41         map_data_colored[ map_data == 3] = walls.WALL_RED
42         map_data_colored[ map_data == 4] = walls.WALL_GREEN
43         walls.setData( map_data_colored)
44         #-----地図情報を表示する-----
45         walls.publish()
46         #-----ここまで-----
47         loop.sleep() # 10[loop/s]になるよう調整する。
48     except KeyboardInterrupt: # try:中にCTRL-C が押されればココを実行する。
49         print("キーボード割り込み！CTRL-C 終了")

```

## サンプルコードの実行結果

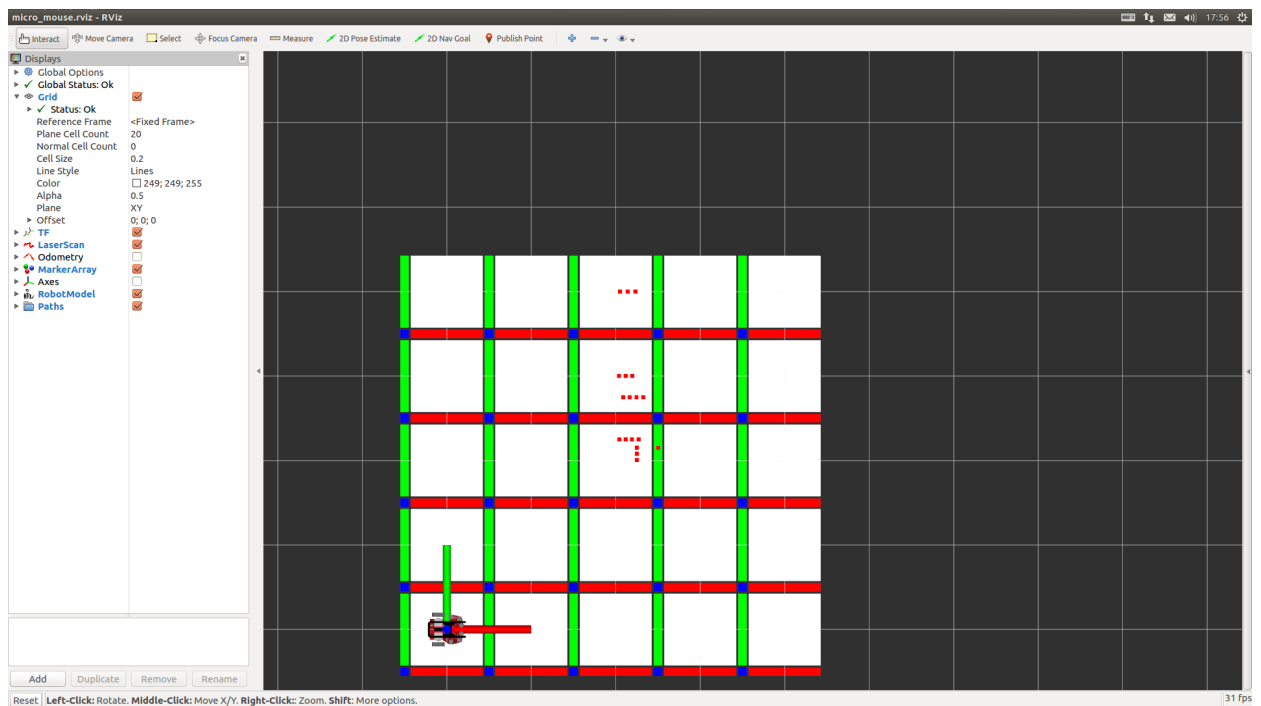


図1 wall\_publisher.py の実行結果

## pathPublisher

### setData( data)

- data

[[x1, y1], [x2, y2], [x3, y3].....] というような 2 次元配列変数を渡してください。値によって Rviz 上で path が変化します。

### publish()

Rviz 上の Path を更新します。

## サンプルコード

### ソースコード 3 script/chapters/2/path\_publisher.py

---

```
1  #!/usr/bin/env python
2  #coding: utf-8 # 日本語を使えるようにする
3
4  import roslib.packages
5  pk_path = roslib.packages.get_pkg_dir('micro_mouse')
6  import sys
7  sys.path.append(pk_path + '/script/lib')
8
9  import rospy # include<ros/ros.h>のようなもの
10
11  import path
12
13
14  if __name__=="__main__":
15      rospy.init_node("node_name")
16      path1 = path.pathPublisher( "path_red" , [ 0.200, 0.200])
17      path2 = path.pathPublisher( "path_green", [ 0.200, 0.200])
18      path3 = path.pathPublisher( "path_blue" , [ 0.200, 0.200])
19      path1.setData( [[0,0], [1,0], [2,1], [3,0]])
20      path2.setData( [[0,0], [0,1], [1,2], [0,3]])
21      path3.setData( [[0,0], [1,1], [2,4], [4,2]])
22      loop = rospy.Rate(10)
23      try:
24          while not rospy.is_shutdown():
25              path1.publish()
26              path2.publish()
27              path3.publish()
28              loop.sleep()
29              pass
30      except KeyboardInterrupt:
31          print(" キーボード割り込み！CTRL-C 終了")
```

---

## サンプルコードの実行結果

わかりやすいように色味を変えています。

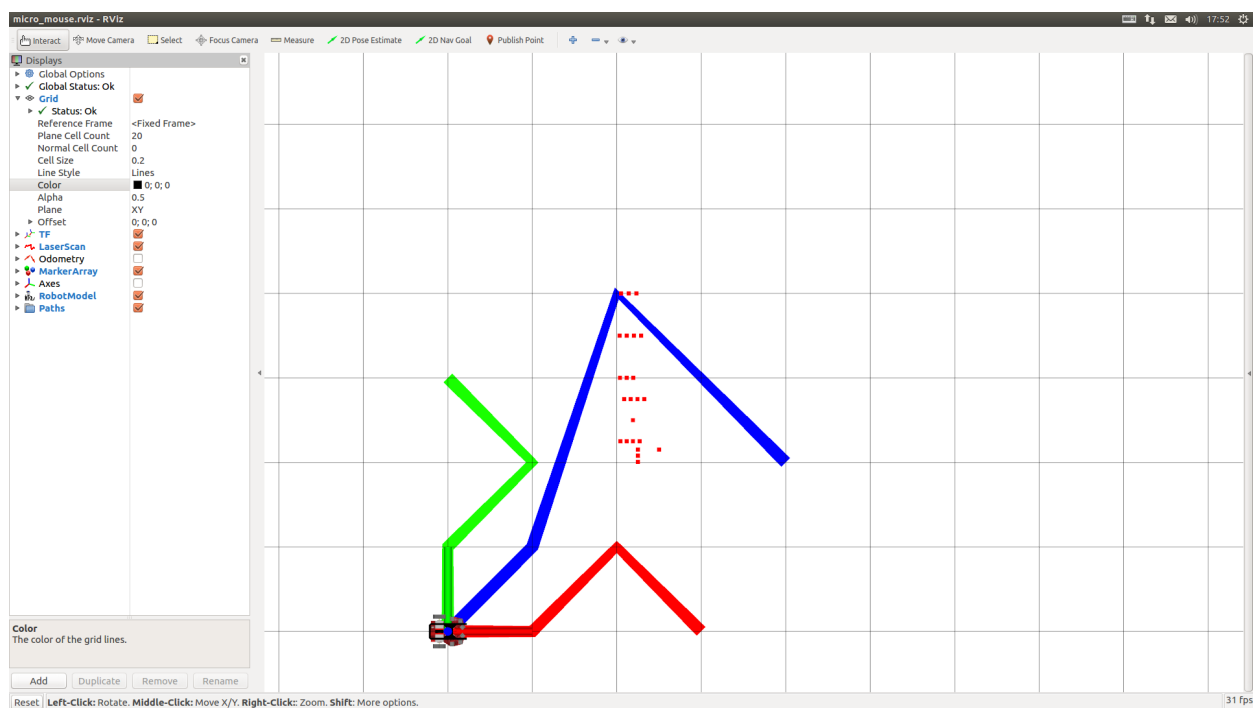


図 2 path\_publisher.py の実行結果