

به نام تو

سید عارف طباطبایی 9831040

در این پروژه، یک فایل از نوع `msg` در فولدر مربوطه شامل جهت ها قرار گرفته است. همچنین در فولدر `src`، چهار فایل از نوع پایتون به نام های `sensor`، `controller`، `motor1` و `motor2` قرار دارد.

مطابق دستور کار، در فایل `sensor`، برای جهات مختلف، عدد رندم تولید کرده و به آن ها نسبت می دهیم:

```
src > sensor.py > ...
1  #!/usr/bin/python3
2
3  import rospy
4  from random import randint
5  from std_msgs.msg import String
6  from hwl.msg import proximity
7
8  def talker():
9
10     pub = rospy.Publisher('distance', proximity, queue_size=10)
11     rospy.init_node('sensor', anonymous=True)
12     rate = rospy.Rate(1) #Hz
13
14     while not rospy.is_shutdown():
15         msg = proximity()
16         msg.up = randint(10,200)
17         msg.right = randint(10,200)
18         msg.down = randint(10,200)
19         msg.left = randint(10,200)
20         rospy.loginfo(msg)
21         pub.publish(msg)
22         rate.sleep()
23
24
25 if __name__ == '__main__':
26     try:
27         talker()
28     except rospy.ROSInterruptException:
29         pass
30
```

در کد کنترلر، ابتدا کمترین مقدار نسبت داده شده به جهت ها را بدست می آوریم و در اینجا برای سادگی فرض را بر این گرفتیم که به سمت آن کمترین مقدار تغییر جهت دهیم. در ادامه این تغییر جهت را `publish` می کنیم:

```
src > controller.py > ...
1  #!/usr/bin/env python
2
3  import rospy
4  from std_msgs.msg import String, Int16
5  from hwl.msg import proximity
6
7  direction_index = 0 # Up
8
9  def get_min(data_list):
10     min_value = min(data_list)
11
12     min_index = [i for i, val in enumerate(data_list) if val==min_value]
13     return min_index[0]
14
15 def direction(data):
16     global direction_index
17     pub1 = rospy.Publisher('direction1', Int16, queue_size=10)
18     pub2 = rospy.Publisher('direction2', Int16, queue_size=10)
19
20     data_list = [data.up, data.right, data.down, data.left]
21     min_distance = get_min(data_list) # new direction
22     tempt = min_distance - direction_index
23     rospy.loginfo("last: %s, now: %s, The mean = %s", direction_index, min_distance, tempt*90)
24     direction_index = min_distance
25
26     if not rospy.is_shutdown():
27         pub1.publish(direction_index*90)
28         pub2.publish(direction_index*90)
29
30
31 def listener():
32     rospy.init_node('control', anonymous=True)
33     rospy.Subscriber("distance", proximity, direction)
34
35     rospy.spin()
36
37
38
39 if __name__ == '__main__':
40     listener()
41
```

برای مثال در تصویر زیر، مقادیر رندم تولید شده در هر iteration را مشاهده می‌کنید:

```
aref@TB:~/Documents/catkin$ rosrun hw1 sensor.py
[INFO] [1680203993.548271]: up: 53
right: 65
down: 69
left: 14
[INFO] [1680203994.550084]: up: 121
right: 177
down: 34
left: 184
[INFO] [1680203995.549794]: up: 116
right: 78
down: 97
left: 157
[INFO] [1680203996.549683]: up: 86
right: 38
down: 58
left: 50
^Caref@TB:~/Documents/catkin$
```

و خواهیم داشت:

```
aref@TB:~/Documents/catkin$ rosrun hw1 controler.py
[INFO] [1680203993.557902]: last: 0, now: 3, The mean = 270
[INFO] [1680203994.555924]: last: 3, now: 2, The mean = -90
[INFO] [1680203995.555739]: last: 2, now: 1, The mean = -90
[INFO] [1680203996.552162]: last: 1, now: 1, The mean = 0
^
```

منظور از last، جهت قبلی و منظور از now جهتی که به سمت آن می‌رویم می‌باشد که در اینجا مقادیر 0 تا 3 را به صورت زیر به جهت‌ها نسبت داده ایم:

$0 \rightarrow up$

$1 \rightarrow right$

$2 \rightarrow down$

$3 \rightarrow left$

و منظور از مقدار مثبت حرکت ساعتگرد و برعکس می‌باشد.

گراف مربوطه نیز به صورت زیر خواهد بود:

