

نام و نام خانوادگی: حسنا اویارحسینی	گزارش تمرین اول اصول علم ربات
شماره دانشجویی: ۹۸۲۳۰۱۰	تاریخ: اسفند - ۱۴۰۱

"تمرین اول"

بررسی تست و خروجی:

با توجه به تصویر صفحه بعد که خروجی یکبار اجرا را نشان میدهد صحت کد را بررسی میکنیم:

۱- خروجی را بررسی میکنیم. فرض شده جهت اولیه ربات در تست زیر به نوعی بوده است که پشت ربات به سمت بالا است. در بار اول سنسور عدد های زیر را تولید کرده است:

Up: ۸۴, down: ۶۸, right: ۶۶, left: ۳۶

که عدد مربوط به فاصله چپ از همه کمتر است پس لازم است ربات به نحوی بچرخد که پشت آن به سمت چپ قرار گیرد. یعنی باید ۹۰ درجه در جهت خلاف عقربه های ساعت بچرخد که مشاهده میکنیم کنترلر به درستی تصمیم گرفته است که ۹۰ درجه در جهت خلاف عقربه های ساعت بچرخد و این پیام را به تایپیک مربوط به موتور ۱ و ۲ ارسال کرده است و این دو موتور نیز پیغام را برداشته و چاپ کرده اند.

۲- سپس ربات که در جهتی است که پشت آن به سمت چپ است از طرف سنسور فاصله های زیر را دریافت میکند:

Up: ۶۲, down: ۳۳, right: ۱۰۰, left: ۳۶

این بار عدد سمت پایین از همه کوچکتر است پس باید به سمت پایین بچرخد یعنی از چپ ۹۰ درجه خلاف جهت عقربه های ساعت، این تصمیم را کنترلر میگیرد و در غالب پیام twist به موتور ۱ و ۲ میدهد و آنها پیام را از تایپیک مربوطه برداشته و چاپ میکنند.

۳- در مرحله بعدی به سمت راست میپیچد باز هم با ۹۰ درجه در خلاف جهت عقربه های ساعت.

۴- در مرحله بعد با توجه به فاصله های زیر باید از راست که موقعیت فعلی باشد به چپ تغییر جهت دهد این بار باید ۱۸۰ درجه بچرخد و زاویه آن فرقی نمیکند که اینجا ساعتگرد در نظر گرفته شده.

Up: ۱۸۱, down: ۱۲۰, right: ۶۲, left: ۲۹

و به همین صورت با گرفتن فاصله های جدید ربات تغییر جهت خواهد داد.

Terminal

Mar 16 03:41

roscore

Motor1

Motor2

```

=====
PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.16.0

NODES

auto-starting new master
process[master]: started with pid [2076]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to a1ac253e-c3e1-11ed-9724-559284d3f184
process[rosout-1]: started with pid [2110]
started core service [/rosout]

```

```

hosna@ubuntu:~/Desktop/catkin_ws$ rosrn hw0 motor1.py
[INFO] [1678963205.555629]: 90
[INFO] [1678963205.562409]: counter clock wise
[INFO] [1678963210.553750]: 90
[INFO] [1678963210.557927]: counter clock wise
[INFO] [1678963215.550004]: 90
[INFO] [1678963215.556366]: counter clock wise
[INFO] [1678963220.547270]: 180
[INFO] [1678963220.556602]: clock wise
[INFO] [1678963225.558863]: 90
[INFO] [1678963225.564483]: counter clock wise
[INFO] [1678963230.561244]: 0
[INFO] [1678963230.570462]: clock wise

```

```

boo
[INFO] [1678963210.555039]: 90
[INFO] [1678963210.560472]: counter clock wise
boo
[INFO] [1678963215.551339]: 90
[INFO] [1678963215.560270]: counter clock wise
boo
[INFO] [1678963220.547246]: 180
[INFO] [1678963220.552760]: clock wise
boo
[INFO] [1678963225.557887]: 90
[INFO] [1678963225.562464]: counter clock wise
boo
[INFO] [1678963230.563393]: 0
[INFO] [1678963230.570193]: clock wise

```

sensor

controller

```

[INFO] [1678963205.544229]: up: 84
down: 68
left: 36
right: 66
[INFO] [1678963210.547801]: up: 62
down: 33
left: 36
right: 100
[INFO] [1678963215.544701]: up: 45
down: 54
left: 29
right: 17
[INFO] [1678963220.540139]: up: 181
down: 120
left: 29
right: 62
[INFO] [1678963225.551972]: up: 190
down: 97
left: 148
right: 184
[INFO] [1678963230.549953]: up: 48
down: 15
left: 126
right: 131

```

```

[INFO] [1678963205.553009]: up
[INFO] [1678963205.558993]: degree: 90
clockwise: False
[INFO] [1678963205.573317]: left
[INFO] [1678963205.578668]: -----
[INFO] [1678963210.553782]: left
[INFO] [1678963210.564415]: degree: 90
clockwise: False
[INFO] [1678963210.570093]: down
[INFO] [1678963210.573233]: -----
[INFO] [1678963215.550002]: down
[INFO] [1678963215.557235]: degree: 90
clockwise: False
[INFO] [1678963215.566633]: right
[INFO] [1678963215.574069]: -----
[INFO] [1678963220.545524]: right
[INFO] [1678963220.551727]: degree: 180
clockwise: True
[INFO] [1678963220.567177]: left
[INFO] [1678963220.570910]: -----
[INFO] [1678963225.557676]: left
[INFO] [1678963225.569585]: degree: 90
clockwise: False
[INFO] [1678963225.575848]: down
[INFO] [1678963225.580856]: -----

```

خروجی rqt_graph:

```

graph LR
    sensor("/sensor_4729_1678963932076") --> distance("/distance")
    distance --> controller("/controller_4606_1678963193191")
    controller --> motor1("/motor1")
    controller --> motor2("/motor2")
    motor1 --> topic1("/motor1_4569_1678963188283")
    motor2 --> topic2("/motor2_4581_1678963189370")

```

توضیح کدها:

Sensor: چهار عدد رندوم تولید کرده و در قالب پیامی از نوع proximity آن را در تاپیک distance پابلیش میکنیم:

```
#!/usr/bin/python3


import random
import rospy
from std_msgs.msg import String
from hw0.msg import proximity

def talker():
    pub = rospy.Publisher("distance", proximity, queue_size=10)
    rospy.init_node("sensor", anonymous=True)
    rate = rospy.Rate(0.2) # Hz (number per second)

    while not rospy.is_shutdown():
        msg = proximity()
        msg.up = random.randint(10, 200)
        msg.down = random.randint(10, 200)
        msg.left = random.randint(10, 200)
        msg.right = random.randint(10, 200)

        rospy.loginfo(msg)
        pub.publish(msg)
        rate.sleep()

if __name__ == "__main__":
    talker()
```



```
hw0 > msg > ☒ proximity.msg
1  int64 up
2  int64 down
3  int64 left
4  int64 right
```

:Controller

ابتدا جهت چرخش را با توجه به وضعیت فعلی پیدا میکنیم سپس دو پیام در دو تاپیک مربوط به موتور ۱ و ۲ ارسال میکنیم که حاوی جهت و زاویه چرخش ثر قالب twist می باشد.

```
from std_msgs.msg import String
from hw0.msg import proximity, twist

state = 0
def callback(data):
    global state
    degree = 0
    clockwise = True
    distances = [data.up, data.right, data.down, data.left]
    min_dis_dir = min(range(len(distances)), key=lambda x : distances[x])
    if (min_dis_dir != state):
        diff = min_dis_dir - state

        if (diff == 1 or state == -3): # turn right 90 degree
            degree = 90
            clockwise = True
        elif (diff == -1 or diff == 3): # turn left 90 degree
            degree = 90
            clockwise = False
        elif (diff == 2 or diff == -2): # turn 180 degree
            degree = 180
            clockwise = True

    msg = twist()
    msg.degree = degree
    msg.clockwise = clockwise

    pub1 = rospy.Publisher("motor1", twist, queue_size=10)
    pub2 = rospy.Publisher("motor2", twist, queue_size=10)

    pub1.publish(msg)
    pub2.publish(msg)

    rospy.loginfo(msg)
    state = min_dis_dir

def listener():|
    rospy.init_node('controller', anonymous=True)

    rospy.Subscriber("distance", proximity, callback)

    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()
```

```
hw0 > msg > ≡ twist.msg
1   int64 degree
2   bool  clockwise
```

Motor\:

پایم را از تاپیک مربوطه برداشته و چاپ میکنیم.

```
#!/usr/bin/python3

import rospy
from hw0.msg import twist

def callback(data):
    clockwise = 'clock wise' if data.clockwise else 'counter clock wise'
    rospy.loginfo(data.degree)
    rospy.loginfo(clockwise)

def listener():
    rospy.init_node('motor1', anonymous=True)
    rospy.Subscriber("motor1", twist, callback)

    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()

if __name__ == "__main__":
    listener()
```

موتور ۲ نیز مانند موتور ۱ است با این تفاوت که پیام را از تاپیک موتور ۲ برمیدارد.