



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
مکاترونیک

گزارش مینی پروژه ۴

نام و نام خانوادگی	کامیار رحمانی
شماره دانشجویی	۸۱۰۱۹۹۴۲۲
تاریخ ارسال گزارش	۱۴۰۲/۳/۲۶

در انجام این پروژه مراحل زیر را طی کردم: (مراحل به ترتیب نیست!)

- یکی از چالش های من در انجام پروژه این بود که وقتی turtlesim_node را اجرا می کردم turtle1 در جای مشخصی قرار داشت و باعث میشد جای مناسبی برای نوشتن حروف دیگر نباشد. به همین دلیل با استفاده از کد زیر جای turtle1 را تغییر دادم و برای اینکه رد حرکتش باقی نماند از سرویس pen_setter استفاده کردم و آرگومان off آن را 1 کردم و همچنین از TeleporterAbsolute استفاده کردم و موقعیت اولیه آن را در $x=6$ و $y=4.4$ قرار دادم:

```
rospy.wait_for_service('turtle1/set_pen')
pen_setter = rospy.ServiceProxy('turtle1/set_pen', SetPen)
pen_setter(0, 0, 0, 0, 1)
rospy.wait_for_service('turtle1/teleport_absolute')
teleporter = rospy.ServiceProxy('turtle1/teleport_absolute', TeleporterAbsolute)
teleporter(6, 4.4, 0.0)
```

- برای اینکه لاکپشت های دیگری را به turtlesim اضافه کنم از سرویس spawn استفاده کردم: برای مثال در شکل زیر turtle2 را اضافه کردم و در موقعیت $x=4$ و $y=4.4$ قرار دادم: (برای سایر لاکپشت ها هم همین را تکرار کردم فقط موقعیت اولیه تغییر کرد)

```
rospy.wait_for_service('spawn')
spawner = rospy.ServiceProxy('spawn', Spawn)
spawner(4.0, 4.4, 0.0, 'turtle2')
```

- در این پروژه برای حرکت لاکپشت هم از دستور velocity و هم از دستور TelePortAbsolute استفاده کردم:

دستور velocity را برای هر دو نوع حرکت خطی و حرکت دورانی (جهت دوران حول خودش) به صورت زیر پیاده سازی کردم:

از روی فاصله ای که قصد داشتم تا turtle حرکت کند و سرعت، زمان لازم برای حرکت را بدست آوردم و تا پایان زمان حرکت پیام Twist را از طریق cmd_vel به لاکپشت ارسال کردم:

```
duration = distance / speed
rate = rospy.Rate(10)
start_time = rospy.Time.now().to_sec()
while rospy.Time.now().to_sec() - start_time < duration:
    pub.publish(vel_msg)
    rate.sleep()
```

برای حرکت دورانی از همین روش استفاده کردم فقط به جای فاصله زاویه را قرار دادم:

```
angle = math.radians(angle)
duration = angle / abs(vel_msg.angular.z)
rate = rospy.Rate(10)
start_time = rospy.Time.now().to_sec()
while rospy.Time.now().to_sec() - start_time < duration:
    pub.publish(vel_msg)
    rate.sleep()
```

برای نوشتن حرف V از دستور velocity استفاده کردم ولی به دلیل اینکه خیلی دقیق نبود با آزمون و خطا توانستم حرف مورد نظر را رسم کنم.

به خاطر عدم دقت دستور velocity از برای نوشتن حروف دیگر از دستور TeleportRelative جهت دوران لاکپشت حول خودش استفاده کردم.

برای نوشتن حرف K از TeleportRelative استفاده کردم که ربات نسبت به موقعیت فعلی خود تغییر کند. آرگومان اول این دستور مربوط به حرکت خطی و آرگومان دوم آن مربوط به حرکت زاویه ای است:

```
teleporter = rospy.ServiceProxy('turtle3/teleport_relative', TeleportRelative)
teleporter(0.0, 90.0 * math.pi / 180)
```

برای نوشتن حرف I تماما از دستور velocity و آزمون و خطا کردن استفاده کردم.

برای مثال یکی از چالش هایی که داشتم این بود که وقتی لاکپشت به سمت جلو حرکت میکرد وقتی میخواستیم نصف همین مقدار را به عقب برگردد به صورت دقیق این کار را نمیکرد و کمی عقب تر یا جلوتر قرار میگرفت که باعث میشد حروف دقیق نوشته نشوند.

به طور کلی در انجام این پروژه برای اینکه حروف به صورت مناسب نوشته شوند آزمون و خطای زیادی انجام دادم تا به نتیجه برسم.

هم چنین برای اینکه رنگ هر عدد و حرف متفاوت باشد از دستور SetPen در سرویس استفاده کردم:

```
rospy.wait_for_service('turtle2/set_pen')
pen_setter = rospy.ServiceProxy('turtle2/set_pen', SetPen)
pen_setter(0, 255, 0, 4, 0)
```

فایل py. مربوط به هر لاکپشت را به صورت جداگانه نوشتم و در نهایت فایل launch را تشکیل دادم تا همه لاکپشت ها همزمان حروف را رسم کنند.

شکل نهایی:

