

## ROBOT OPERATING SYSTEM PROJECT

### HAZIRLAYANLAR

MEHMET AYTAÇ AKTAN - 2111504213

HALİL DAYI - 2111504015



---

Robot Operating System

## ÖZET

Bu projenin amacı: Robot Operating System 2'in işleyişi hakkında bilgi sahibi olmaktır, Turtlesim ortamı aracılığı ile node kavramının öğrenmektir. Böylece gerçek hayattaki yüzlerce node'a sahip olan projeleri anlayabilmemiz, bu node'lar arasındaki bağlantıyı anlayabilmemiz ve gerektiğinde bu node'ları bir amaç için kullanabileceğimiz bir başlangıç noktası olmaktadır.

Projede Python dili kullanarak Turtlesim ortamında çizdirmeye çalıştığımız harfler Y, 3, 5 ve B'dir. Bu proje Robot Operating System 2 gerektirdiği için ve Ubuntu'da bunu kurmak daha elverişli olduğu için ilk olarak Virtual Machine'i bilgisayarımıza kurduk. Virtual Machine ise bilgisayar kaynaklarının bir kısmını, seçilen işletim sistemine kullanımı için verir. Böylece Ubuntu işletim sistemi, bilgisayarımızdan tahsis edilen kaynakları kullanabildi ve Robot Operating System 2'i kurabildik. Robot Operating System 2 kurulumundan sonra ise gerekli olan paket ve uygulamaları indirerek Turtlesim ortamında Python dili aracılığıyla verilen harfleri çizmeyi tamamladık.

## GİRİŞ

Bu proje ile yazılım geliştirmeye uğraşanlar, ROS2 ve Python dili ile nasıl interaktif uygulamalar geliştirebileceklerini öğrenecektir.

Python dili kullanarak Turtlesim ortamındaki node'ların kontrolünü sağladık. Bu node'ların kontrolü için gerekli olan temel parçalar: Topic, Publisher ve Subscriber.

Topic: Veri alışverişinin gerçekleştiği temel iletişim yoludur. Topic veri yayınlayan bir Publisher ve veri alan bir veya birden fazla Subscriber arasında işbirliği yapar.

Publisher: Belirli bir Topic üzerinden veri yayınlanması veya gönderilmesi gereken bir bileşendir. Bu projedeki Publisher, Turtlesim ortamındaki node'ların hareketleri için gerekli olan verileri yayınlamıştır. Örneğin: kaplumbağa node'larını döndürmek veya ötelemek için Publisher'dan gelen verilerin yayınlanmasına ihtiyaç duyulur.

Subscriber: Belirli bir Topic üzerinden yayınlanan verileri almak veya okumak için kullanılan bir bileşendir. Bu projedeki Subscriber, Turtlesim ortamındaki node'ların hareketleri için gerekli olan verileri abone olduğu Topic üzerinden alır ve uygular.

Ayrıca yazılım geliştirmeye uğraşanlar, Turtlesim ortamında harf çizdirme işlemi yaparak temel ROS2 iletişimini ve hareket kontrolünü de öğrenecektir. Bu beceriler, robotik ve otonom sistemler geliştirme süreçlerinde büyük önem taşımaktadır.

## YÖNTEM

Bu projede ROS2 Humble sürümü, Python 3 dilini, Virtual Machine ve Ubuntu işletim sistemini kullandık. Proje için Turtlesim paketi kullanılarak Python kodları aracılığıyla Turtlesim ortamında Y, B, 3 ve 5 harflerini çizmek için hareket komutları gönderildi. Python'daki hareket komutları, Turtlesim ortamına gelerek bu harflerin çizimi elde edildi.

Kodlarımızda "rclpy" isimli Python kütüphanesini kullandık. Bu kütüphane, Python'da ROS2 node'larını oluşturmamızı ve ROS2 ile iletişime geçmemiz için Python arayüzü sağladı. Mesajların ve servislerin yayılması için ROS2'nin iletişim mekanizmasından yararlandık.

Kodumuzda "import threading" yazarak Thread yapısını kullanabildik. Böylece Turtlesim ortamındaki üç node aynı anda çalışırken geriye kalan bir node, bu üç node'un çalışması bittikten sonra çalışması sağlandı. Ayrıca "import time" yazarak time.sleep() fonksiyonunu kullanabildik. Böylece Turtlesim ortamında çalışan node'ların, dönme esnasında dönüşlerini düzgün tamamlayabildi.

## Y HARFİ

```
def draw_Y(node, publisher):
    msg = Twist()

    for _ in range(3):
        # Y sayısının alt temelini çizimi
        msg.linear.x = 2.0
        msg.angular.z = 0.0
        publisher.publish(msg)
        time.sleep(1)

        # Sonraki çizim için dönüş açılarının ayarı
        msg.linear.x = 0.0
        msg.angular.z = 0.7853981634 # Yaklaşık 45 derece
        publisher.publish(msg)
        time.sleep(1)

        # Geriye kalan üst kısımları loop ile çizimi
        msg.linear.x = 2.0
        msg.angular.z = 0.0
        publisher.publish(msg)
        time.sleep(1)

        # Sonraki çizim için dönüş açılarının ayarı
        msg.linear.x = 0.0
        msg.angular.z = 3.1415926536 # Yaklaşık 180 derece
        publisher.publish(msg)
        time.sleep(1)

    # Hareketin Sonu
    msg.linear.x = 0.0
    msg.angular.z = 0.0
    publisher.publish(msg)
```

Yukarıdaki görsel Y harfinin çizilmesi için gereken Python kodudur. İlk olarak kaplumbağayı 45 derece döndürüp öteliyoruz. Sonra 180 derece döndürdükten sonra tekrar öteliyoruz. Böylece Y harfinin sağ kısmı bitmiş oldu. Daha sonra kaplumbağayı 45 derece döndürüyoruz ve öteliyoruz. 180 derece döndürdükten sonra tekrar öteliyoruz. Böylece Y harfinin alt kısmı da bitmiş oldu. En sonunda kaplumbağayı yaklaşık 45 derece döndürüp öteliyoruz ve 180 derece döndürüyoruz böylece Y harfinin sol kısmını da tamamlamış oluyoruz.

## 5 SAYISI

```
def draw_five(node, publisher):
    msg = Twist()
    msg.linear.x = 2.0
    msg.angular.z = 1.8 # Yaklaşık 103 derece

    # İlk yarım çemberi çizmek için
    for _ in range(20):
        publisher.publish(msg)
        time.sleep(0.1)

    # Sonraki çizim için dönüş açıları ayarları
    msg.linear.x = 0.0
    msg.angular.z = -1.6 # -90 derece
    publisher.publish(msg)
    time.sleep(1.2)

    msg.linear.x = 1.0
    msg.angular.z = 0.0
    for _ in range(10):
        publisher.publish(msg)
        time.sleep(0.1)

    # Sonraki çizim için dönüş açıları ayarları
    msg.linear.x = 0.0
    msg.angular.z = -1.4 # Yaklaşık -90 derece
    publisher.publish(msg)
    time.sleep(1)

    msg.linear.x = 1.0
    msg.angular.z = 0.0
    for _ in range(5):
        publisher.publish(msg)
        time.sleep(0.1)
```

Yukarıdaki görsel 5 sayısının çizilmesi için gereken Python kodudur. İlk olarak kaplumbağayı 103 derece döndürüyoruz. Daha sonra yarım daire çizerek 5'in alt kısmını tamamlıyoruz. Daha sonra 90 derece döndürüp öteliyoruz. En son kaplumbağa'yı 90 derece döndürüp öteliyoruz ve böylece 5 sayısının çizimini tamamlamış oluyoruz.

## 3 SAYISI

```
def draw_three(node, publisher):  
    msg = Twist()  
    msg.linear.x = 1.9  
    msg.angular.z = 1.9 # 108 derece  
  
    for _ in range(20): # 3 sayısının alt kısmının çizimi  
        publisher.publish(msg)  
        time.sleep(0.1)  
  
    msg.linear.x = 0.0  
    msg.angular.z = -2.4 # -130 derece  
    publisher.publish(msg)  
  
    for _ in range(15): # 3 sayısının üst kısmı için gerekli dönüşün yapılması  
        publisher.publish(msg)  
        time.sleep(0.1)  
  
    msg.linear.x = 1.5  
    msg.angular.z = 1.9 # 108 derece  
  
    for _ in range(11): # Sayının üst kısmının tamamlanması  
        publisher.publish(msg)  
        time.sleep(0.1)
```

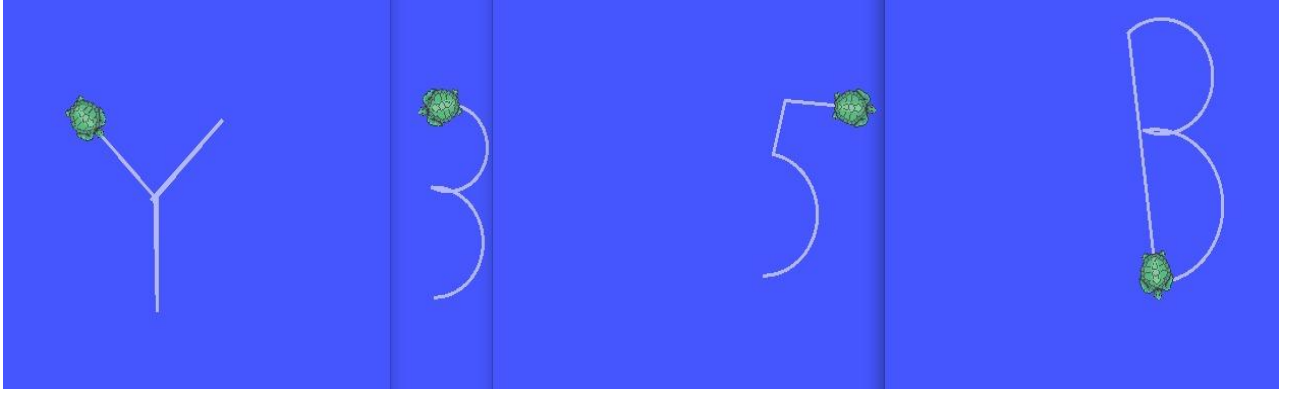
Yukarıdaki görsel 3 sayısının çizilmesi için gereken Python kodudur. İlk olarak 3 sayısının alt tarafındaki yarım daireyi çiziyoruz. Sonra kaplumbağayı 130 derece döndürüyoruz ve 3 sayısının üst tarafındaki yarım daireyi çiziyoruz. Böylece 3 sayısının çizimini tamamlamış oluyoruz.

## B HARFİ

```
def draw_B(node, publisher):  
    msg = Twist()  
    time.sleep(15)  
    msg.linear.x = 1.9  
    msg.angular.z = 1.8  
  
    for _ in range(20): # Çemberin tamamını dönmek için 30 adım yeterli olabilir  
        publisher.publish(msg)  
        time.sleep(0.1)  
  
    msg.linear.x = 0.0  
    msg.angular.z = -2.0 # -90 derece  
    publisher.publish(msg)  
    time.sleep(0.01)  
  
    for _ in range(15): # Yarı çemberin yarısını dönmek için 15 adım yeterli olabilir  
        publisher.publish(msg)  
        time.sleep(0.1)  
  
    for _ in range(15): # Çemberin tamamını dönmek için 30 adım yeterli olabilir  
        msg.linear.x = 2.2  
        msg.angular.z = 2.3  
        publisher.publish(msg)  
        time.sleep(0.1)  
  
    msg.linear.x = 0.0  
    msg.angular.z = 1.25  
    publisher.publish(msg)  
    time.sleep(1.1)  
  
    for _ in range(11): # Çemberin tamamını dönmek için 30 adım yeterli olabilir  
        msg.linear.x = 2.0  
        msg.angular.z = 0.0  
        publisher.publish(msg)  
        time.sleep(0.1)
```

Yukarıdaki görsel B harfinin çizilmesi için gereken Python kodudur. İlk olarak B harfinin alt kısmındaki yarım daireyi çizmeye çalışıyoruz. Daha sonra kaplumbağayı 180 derece döndürüp B harfinin üst kısmındaki yarım daireyi çizmeye çalışıyoruz. Sonra B harfini çizmenin üçüncü aşamasına geliyoruz. Kaplumbağayı 90 derece döndürdükten sonra loop bitene kadar yukarıdan aşağıya öteliyoruz. Böylece B harfinin son aşamasını da tamamlamış oluyoruz. B harfi diğer üç harf çizildikten sonra çizilecektir.

## SONUÇ



Yukarıdaki görselde şimdiye kadar gösterdiğimiz kodların Turtlesim ortamındaki halidir. Yukarıda dört farklı pencerede Python kodunu çalıştırdığımızda Turtlesim ortamındaki node'ların, Python kodlarından aldığı veriler ile çizdiği şekiller görülmektedir. Y sayısı, 3 ve 5 harfleri aynı anda çizilirken B harfi diğer üç node işini bitirdikten sonra çizilmiştir.

## KAYNAKÇA

- ROS2 HUMBLE – <https://docs.ros.org/en/humble>
- Stack Overflow
- Youtube – ROS2 Tutorials – ROS2 Humble For Beginners --  
<https://www.youtube.com/playlist?list=PLSegLrePWgJudpPUof4-nVFHGkB62lzy>
- Udemy - <https://www.udemy.com/course/ros2-for-beginners/>