

# ITU AUV Ödev Raporu

Ödev üzerinde çalışmaya Ubuntu'yu indirip dual boot olarak bilgisayarıma kurarak başladım. Son zamanlarda kullanmaya başladığım Zed adlı editörü bilgisayarımın görüntü driverlarının Ubuntu 20.04'ü desteklememesi sebebiyle çalıştıramadım. Forumlardaki farklı çözümleri denedim ama çalışsa bile çok kötü performans verdiği için Zed kullanmaktan vazgeçtim. Bu aşamalarda farklı bir problemle karşılaşmadım.

Sitesindeki anlatımları takip ederek ROS Noetic'i kurdum. Beginner seviyesi tutorialları takip ederek tamamladım. Karşılaştığım kavramları ve terminal komutlarını tekrar bakabilmek için not aldım ([link](#)). Node, topic, publisher, subscriber gibi kavramlar daha önce çalıştığım sistemlerden tanıdık geldiği için anlayıp uygulamakta sorun yaşamadım. `turtlesim` ile farklı deneyler yapıp kavramlara olabildiğince hakim olmaya çalıştım.

Dil syntaxlarını hatırlamak amacıyla topic-subscriber ödevini hem Python hem C++ ile uyguladım. İlk başta `SensorData` adıyla bir msg oluşturdum. Sensörü tanımlamak için `id`, değeri için `value` fieldları ekledim. `talker` isimli node 10Hz rateiyle `sensor_data` topic'i üzerinde `SensorData` tipinde veri publish ediyor, `listener` isimli node ise bu topic'e subscribe olmuş ve gelen verileri okuyarak, okuduğu toplam veri sayısını `message_count` isimli topic'e publish ediyor. Terminal üzerinden gözlemleyebilmemiz için iki node da publish ettikleri ve okudukları verileri `ROS_INFO` fonksiyonuyla logluyorlar. Kod üzerine de açıklayıcı commentler ekledim.

`ros::spin()` ve `ros::spinOnce()` fonksiyonlarının işlevini tutoriallar üzerinde tam olarak anlayamadım. Detaylı araştırınca anladığım kadarıyla; `spin` farklı looplar içermeyen ve sadece gelen mesajlara cevap veren nodelarda kullanılıyor, infinite bir loopa giriyor. `spinOnce` ise o andaki mesajlara cevap veriyor ve node'un içindeki loop kaldığı yerden devam ediyor, yani kodu bloklamamış oluyor.

Bunlar dışında `CMakeLists.txt` ve `package.xml` dosyalarında da mesajlaşma ve yazdığım scriptlerin çalışması için çeşitli değişiklikler yaptım. Bu kısımlarda tutorialı takip ettim, yaptıklarımın etkisini tam olarak anlayamadım. C++ altyapım eksik olduğu ve CMake hakkında çok fikrim olmadığı için anlayamadığımı düşünüyorum.

[Proje dosyalarına bu repodan \(talhabw/auv-homework\) ulaşabilirsiniz.](#)

**Talha Karasu**