**ETERATION CASE ÇALIŞMASI**

**RAPORU**

**HAZIRLAYAN**

**KADİR CEYHAN**

i

**ÖZET**

Case çalışması sürecinde iki cihazın haberleşmesi (bu genelde robot ile iletişimde kullanılıyor) üzerinde çalışmalar yapılmıştır. Geliştirme sürecinde Ubuntu 20.04 versiyonu ile geliştirme yapılmıştır. Buradan hareketle ROS Noetic Ninjemys (latest LTS) yazılımı tercih edilmiştir. Bir Publisher ve bir subscriber arasında iletişim gerçekleşmektedir. Publisher tarafından yayınlanan “ Merhaba, ben Kadir “ mesajına , subscribe olan bir cihaz mesaj yayınlandığı an mesaj alındı çıktısını verip ekrana bastırılmaktadır. Çıktılar terminal ekranı üzerinden görünmektedir. Kullanıcı çalışmayı durdurana kadar program çalışmaktadır. Çalıştırılan script dosyaları Python 3.8.10 ile geliştirilmiştir. Son olarak ise projenin tek bir yerden çalışmasını sağlamak için bir launch dosyası yazılmıştır.

ii

**İÇİNDEKİLER**

ÖZET ..........................................................................................................................................i

İÇİNDEKİLER............................................................................................................................ii

1. GİRİŞ......................................................................................................................................1

3. ROS NASIL ÇALIŞIR............................................................................................................2

4. MATERYAL VE YÖNTEM....................................................................................................3

4.1.MATERYAL……………………………………............…………………………….…….3

4.2.YÖNTEM……………………………………………............……………………….…….4

5. ARAŞTIRMA SONUÇLARI…...………………………………….……………….….....…5

6.KAYNAKÇA……………………………………………………………………………...….6

1

**1. GİRİŞ**

ROS (Robot Operating System, Robot İşletim Sistemi), 2007 yılında Willow Garage firması tarafından Stanford Üniversitesi Yapay Zekâ laboratuvarında yapılan çalışmaların sonucunda oluşturulan bir robotik yazılım platformudur. Açık kaynaklı bir meta işletim sistemidir. Günümüzde robotlarla iletişim kurmada sıkça kullanılmaktadır. Kullanım alanı olarak Çoklu robotlar, Küçük ve gömülü bilgisayar platformları, gerçek zamanlı sistemler, ideal olmayan ağlar da çokça kullanılmaktadır. Proje kapsamında ise ROS kullanılarak iletişim ve haberleşme sağlanmıştır.

2. KAYNAK ARAŞTIRMASI

Dokümantasyon inceledikten sonra kullanılacak araçlar liste halinde çıkarıldı. Bunlardan ilki linux işletim sistemidir. ROS Windows işletim sisteminde de çalışmasına karşın daha çok kullanıldığı alan linux işletim sistemidir. Bu doğrultuda bir sanal makine oluşturulup içerisine Ubuntu 20.04 versiyon işletim sistemi kuruldu. Daha sonra Python programlama dili tercih edileceği için ayarlamaları yapıldı. Python kodlar Ubuntu üzerinde gedit kullanılarak yazılacağı için ortam sağlandı. Daha sonra uygun ROS seçilerek kurulum sağlandı. Git ortamı da proje kapsamında olduğu için Ubuntu üzerinde indirildi ve kuruldu. Araştırmalar internet üzerinden yürütüldü.

2

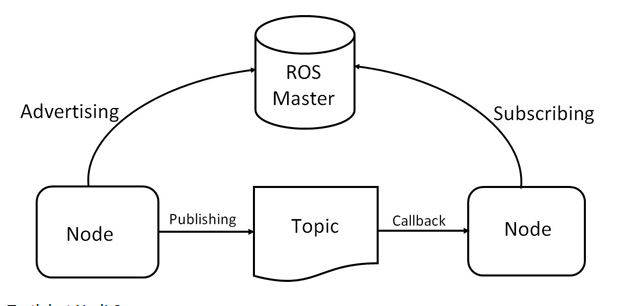
**3. ROS NASIL ÇALIŞIR**

ROS işlemlere düğüm (node) adı verilen ve her düğümün sadece bir işten sorumlu olacağı şeklinde tasarlanmıştır. Düğümler birbirleri arasındaki iletişimi topic adı verilen kanallar ile sağlamaktadır. ROS haberleşme sisteminde yayınla-abone ol (Publisher-subscribe) sistem modeli ile çalışmaktadır.

Örnek olarak bir Publisher’ın yayınladığı mesajı bir subscribe alabilmesi için aynı topic adına sahip olunması gerekmektedir. Proje kapsamında topic adı ‘mytopic’ olarak belirlenmiştir.

Konulara abone olan ve yayın yapan düğümler arasında veri alışverişinin sağlanabilmesi için ise sistem tasarımı gereği bir ana (master) düğüm sistem tarafından oluşturulmaktadır ve bu düğüm ROS üzerindeki düğümleri adlandırma ve kayıt hizmeti sağlamaktadır. İçerisinde TCP/IP kullanmaktadır.

ROS uzay çalışmalarında da kullanılmaktadır.



3

**4. MATERYAL VE YÖNTEM**

**4.1. Materyal**

Proje detaylı bir şekilde anlaşıldıktan sonra materyaller aşağıdaki gibi gereksinimler, materyaller listesi çıkartıldı.

Virtualbox: Virtualbox bir sanal makinedir. Halihazırda bulunan işletim sistemi üzerine farklı işletim sistemleri kurularak işlem yapmaya olanak sağlar. Oracle’ a ait bir ürün olup indirme işlemi kendi sitesi üzerinden gerçekleştirilmiştir.

Linux-Ubuntu (20.04) : Ubuntu ROS ‘un daha verimli bir şekilde kullanılabilmesi için internet üzerinden indirme işlemi gerçekleştirildi ve virtualbox içerisine kurulum gerçekleştirildi.

Python (3.8.10): Python bir bilgisayar programlama dilidir. ROS için c++ programlama dili de kullanılmaktadır fakat genelde Python kullanıldığı için ve dokümantasyonda önerilen dil olduğu için tercih edildi.

Gedit: Gedit Python kodlarının yazıldığı bir ekrandır. Bu sayfada kodlar yazıldıktan sonra terminal ekranından denemler debuglar ve çalıştırmalar yapıldı ve ekranda gösterildi.

Git: Git teknolojisi ekip çalışmalarında yoğun bir şekilde kullanılmaktadır. Bu sayede yapılan işler diğer ekip tarafından kolay bir şekilde takip edilebilir ve projenin son hali ile çalışmaya devam edilebilir. Proje kapsamında Ubuntu üzerinde git yüklendi ve hesap oluşturulup commit edildi.

Gihub: Github projenin saklandığı, proje sürecinde ekip tarafından son denemelerin görüldüğü haberleştiği ortamdır. Günümüzde sıkça kullanılmaktadır.

ROS: Son olarak Ubuntu işletim sistemine uygun ROS yüklenerek kurulum gerçekleştirildi. ROS ile kullanılacak kütüphaneler de indirildi ve Python kodları yazılırken kullanıldı.

4

**4.2. YÖNTEM**

Dokümantasyon detaylı bir şekilde inceledikten ve anlaşıldıktan sonra ilerleyiş için gereksinimler ve kısa notlar alındı. Yukarıda bahsedilen materyaller yüklendikten sonra kullanılamaya başlandı. Bu işlem öncesinde internet üzerinde çeşitli araştırmalar tanımlar ve projeler incelendi. Onlardan çıkarımlarda bulunarak küçük denemeler yapılarak başlandı. Örnek olarak önce sadece bir Publisher scripti yazıldı ve denendi. Ardından bu Publisher ile aynı topic adına sahip subscriber scripti yazıldı ve iletişim sağlanmaya çalışıldı. Bu kısımlarda çeşitli sorunlarla karşılaşıldı. Fakat denemeler ve araştırmalar ile sorunlar çözüldü. Talker ve listener scriptleri tek bir paket içerisine bir daha düzenli olması bakımından scripts adlı klasöre koyuldu. Kodlar yanına yorum satırı yazılarak açıklandı.

Son olarak ise projenin tek bir yerden çalıştırılması için bir launch dosyası yazıldı. Ardından terminal ekranından oluşturulan launch dosyası roslaunch komutu ile çalıştırıldı.

Komutlar sırasıyla;

source devel/setup.bash

roslaunch composiv\_tryouts composiv\_tryout.launch

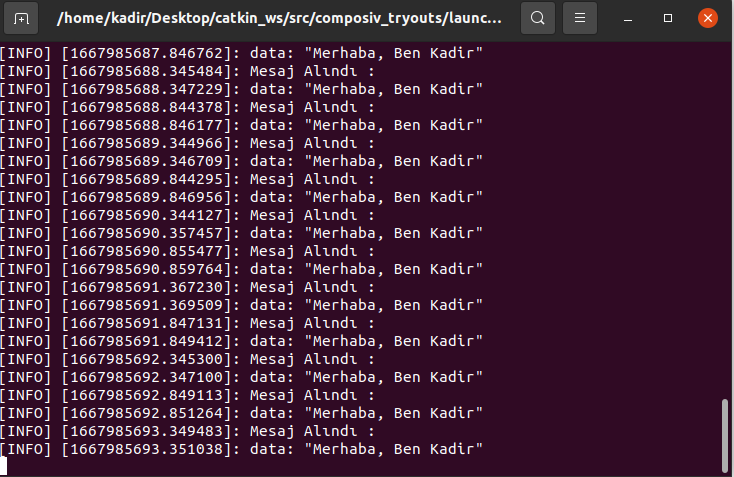
şeklindedir.

Proje kapsamında tüm isimlendirmeler dokümantasyona uygun yapılmıştır.

5

**5. ARAŞTIRMA SONUÇLARI**

Araştırmalar sonucunda proje başarılı bir şekilde çalıştırılmış ve isterler tamamlanmıştır. Projenin son terminal çıktısı aşağıdaki gibidir.



6

**6. KAYNAKÇA**

1. https://devnot.com/2020/ros-robot-operating-system-nedir/

2. https://www.aselsan.com.tr/tr/inovasyon/haber-detay/ros-nedir-gunumuzden-ornekler-ve-kullanim-alanlari-3423

3. https://www.ros.org/

4. http://wiki.ros.org/Documentation

5. https://en.wikipedia.org/wiki/Robot\_Operating\_System

6. https://www.virtualbox.org/

7. https://ubuntu.com/download

8. https://ubuntu.com/tutorials

9. https://www.youtube.com/watch?v=JprymETaUVQ

10. https://github.com/ros/ros

11. https://github.com/ros/ros\_tutorials/tree/noetic-devel/turtlesim/images

12. https://automaticaddison.com/how-to-create-and-execute-ros-launch-files/

13. https://media.geeksforgeeks.org/wpcontent/uploads/20220213134817/ROSpublisher.jpg

14. https://www.youtube.com/watch?v=g4ra4toM0cU&list=PLVVHpwCs2zSkIn1rBu1Ccm4JaC2bAewPe&index=8

15. https://www.youtube.com/watch?v=MkiWj4VwZjc