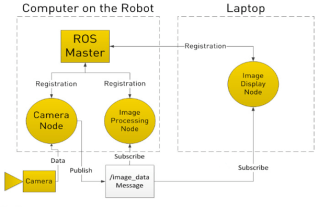
**ROS Nasıl Çalışır?**

**Bir ROS sistemi yayınlama/abone mesajlaşma modelini kullanarak diğer düğümler ile iletişim kuran düğümlerden oluşur. Bir ROS sistemi içerisinden birçok düğüm bulunabilir. Düğümler, hesaplama yapabilen işlemlerdir. Örneğin; bir düğüm kameradan görüntü alır, bir düğüm görüntüyü işler, bir düğüm görüntüyü görüntülenmesini sağlayabilir. Bu düğümlerin birbirlerinden haberdar olup birbirleri ile haberleşebilmeleri için ROS Master’a ihtiyaç vardır. ROS Master merkezi XML-RPC sunucusudur yani mesajları içeren ağ tabanlı bir yapıdır. Biraz daha ayrıntılı şekilde bu konuya açıklık getirecek olursak; ROS, ROS Master ile başlar ve ROS düğümlerinin birbirlerini bulmasını ve birbirleriyle konuşmasını sağlar. Düğümler haberleşmeyi konular yayınlayıp bu konulara abone olarak gerçekleştirir. Diyelim ki robot üzerinden bulunan bir kameradan alınan görüntüleri robot üzerinde ya da başka bir bilgisayarda görmek istiyoruz. Kamera ile iletişim kurmak için bir kamera düğümü, görüntüleri işlemek için bir görüntü işleme düğümü ve görüntü gösterme düğümü vardır. Bu düğümlerin hepsi ROS Master’a kayıtlıdır. Kamera düğümü, ROS Master’a kayıt olurken /image\_data adında bir konu yayınlayacağını belirtir. Diğer düğümler ise kameradan yayınlanan bu konuya abone olduklarını belirtir. Bu şekilde kamera düğümü kameradan alınan verileri /image\_data konusu üzerinden mesajları diğer düğümlere iletir. Not: Bir düğüm birden fazla konu yayınlayabilir ve birden fazla konuya abone olabilirler.**

****

**ROS’un Alternatifleri**

**ROS’a alternatif olabilecek başka robot yazılımları da mevcuttur. Bunlar Player, YARP, Orocos, Carmen, Orca, MOOS, Microsoft Robotic Studio.**

# http://yapbenzet.kocaeli.edu.tr/ros-ornek-uygulama/urulum ve Yapılandırma

ROS'un kurulumu için Ubuntu 14.04 ve ROS Indigo dağıtımlarını kullanacağız.

### 1.1 ROS Ubuntu Paket Deposu

ROS'un paketlerini çekebilmemiz için, depo adresini eklememiz lazım.

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu trusty main" > /etc/apt/sources.list.d/ros-latest.list'

Ardından GPG anahtarını yükleyeceğiz:

wget --no-check-certificate https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -

Son olarak depo önbelleğini güncelleyeceğiz:

sudo apt-get update

### 1.2 Kurulum

ROS, Rviz, simülatörler ve çeşitli kütüphaneler için şu komut yeterli:

sudo apt-get install ros-indigo-desktop-full

Gerekli olduğu takdirde **ros-indigo** ile başlayan paketler kurulabilir.

### 1.3 rosdep

Bağımlılıkların kurulumu için **rosdep** adlı bir araç gerekiyor. Kurulum ve yapılandırma için:

sudo rosdep init

rosdep update

### ROS Çalışma Alanı (Workspace)

ROS'la çalışırken bir çalışma alanı (workspace) kullanılabilir. Bunun için önce bir paketin kurulması lazım:

sudo apt-get install python-rosinstall

Ardından istenilen yerde, örneğin ev dizininde, bir çalışma alanı klasörü oluşturalım:

cd ~

mkdir ROS\_Workspace

rosws init ROS\_Workspace /opt/ros/indigo

**ROS\_Workspace** klasörü artık ROS için bir çalışma alanı haline geldi. Bunu aktifleştirmek için, kullanılan terminal programının ayar dosyasına bir satır eklenmesi lazım. Bash kullanılıyorsa ev dizinindeki **".bashrc"** dosyasına, Zsh kullanılıyorsa da yine aynı klasördeki **".zshrc"** dosyasına aşağıdaki satır eklenmeli.

Bash için:

source ~/ROS\_Workspace/setup.bash

Zsh için:

source ~/ROS\_Workspace/setup.zsh

Ardından terminal oturumu yeniden başlatılarak, ROS komutları aktifleştirilebilir.

Son olarak paketler için çalışma alanında özel bir klasör oluşturmamız gerekiyor. Aşağıdaki komutları sırayla verelim:

roscd

mkdir sandbox

rosws set sandbox

Artık **sandbox** klasöründeki tüm paketler, ROS tarafından hızlıca erişilebilir halde olacak.

**ROS - Robot Operating System Nedir?**

10 Mar 2015

Öncelikle ROS’u kısaca tanımlamaya çalışacağım;

* Robotlar için bir işletim sistemidir.
* Paketler için bir koleksiyondur ve yazılım derleme aracı olarak kullanılmaktadır.
* Dağıtılmış işlemler/makineler arası iletişim ve yapılandırma için bir mimaridir.
* Sistemin çalışma ve analiz aşamaları için geliştirme aracı olarak kullanılabilir.
* Dil bağımsız bir mimariye sahiptir. Aynı robot üzerinden birçok farklı dil kullanılabilir. (c++, python, lisp, java)

**ROS Ne Değildir?**

* Gerçek bir işletim sistemi değildir.
* Bir programlama dili değildir.
* Geliştirme ortamı/IDE değildir.

**ROS Dağıtımları**

* ROS Fuerte Turtle
* ROS Electric Emys
* ROS Diamondback
* ROS C Turtle
* ROS Box Turtle

**ROS ve Komponentleri**

**ROS Core**

ROS Master,

* Merkezi XML-RPC sunucusudur.
* Bağlantıların iletişimi sağlamaktadır.
* Sabit konfigürasyon parametreleri ya da diğer keyfi verileri kaydeder.
* Aslında, insanların anlayabileceği dilde mesajları barındırran ağ tabanlı bir yapıdır.

**ROS Stacks & Packages**

ROS kodu iki farklı seviye içerisinde gruplanmıştır;

* Packages
* Stacks

Packages, ROS’a çok küçük bir bağımlılıklar bağlanmış yazılım koleksiyonlarıdır. Stacks ise bu ufak parçaların bir araya gelerek oluşturduğu dağıtıma verilen isimdir.

**Packages**

* Kodlarınızı, build yapılandırmasınızı ve launch dosyalarınızı barındıran bir dizindir.
* Birden fazla node içerebilir.
* İçerisinde manifest.xml adında bağımlılıkların tanımlandığı bir konfigürasyon dosyası barındırır.
* Sadece paketin amacı ile ilgili kodları içerir.

**Nodes**

* Bazı fonksyionları yerine getiren yapılardır.
* Diğer Topic ve Servicesler ile haberleşebilir.
* Node isimleri benzersizdir.
* Node yapısı, iyi ayrılmış, skale edilebilir bir yapının oluşturulması için vardır.

**Build System**

* Kodların nasıl derleneceğinin belirtildiği CMakelist.txt dosyasına ihtiyaç duymaktadır.
* rosmake komutu paket ve bağımlılıkları derler.
* Eğer bağımlılıklar indirilmemişse, indirilir ve kurulur.
* Aynı anda birden fazla paket derlenebilir. ROS önce bağımlılıkları çözer.

**Komut Satırı**

ROS aşağıdaki bazı komutları içerisinde barındırır;

* roscd: ROS paketi dizinine geçer.
* rosls: Paket içeriğini listeler.
* rosmake: Paketi derler.
* roscreate-pkg: Yeni bir paket oluşturur.
* rosdep: Paket bağımlılıklarını listeler.
* roscp: Paketler arası kopyalama işlemi gerçekleştirir.
* rostest: Test dosyalarını çalıştırır.

**İletişim Yolları**

**Topic (Pub/Sub)**

* Asenkron iletişimler için kullanılır.
* İçerisinde güçlü mesaj tipleri barındırır.
* Callback fonksyionlara sahiptir, bu sayede mutli-thread çalışabilir.
* İstek/cevap etkileşimleri için uygun değildir.
* Çoktan-Çoka bağlantılar için uygundur.

**Service (Higher Priority)**

* Standart fonksiyonlar gibi, senkron olarak kullanılır.
* Güçlü mesaj tiplerini içerisinde barındırır.
* Bire-bir iletişim için uygundur.
* Birden fazla istemciler olabilir.
* Servis çağrıları boyunca Topic Callback fonksyionları herhangi bir sorun teşkil etmez.

**Actions**

* Topic yapısının üzerine inşaa edilmiştir.
* Uzun işlemler gerçekleştirilmesi için kullanılır.
* İptal edilebilir.

**Messages**

* Node iletişimi mesaj gönderimi üzerine kuruludur.
* Mesajlar, özel tipli alanlardan oluşan bir veri yapısından ibarettir.
* ROS içerisinde birçok hazır mesaj standartı vardır ve özel olarak buna yenileri eklenebilir.

[KodKarma](https://kodkarma.blogspot.com.tr/)

“Somewhere, something incredible is waiting to be known.” - Carl Sagan

**1 Ocak 2017 Pazar**

ROS (Robot Operating System ) nedir, nasıl kurulur ?

Robot İşletim Sistemi (ROS), robotu oluşturan parçaların birbirleriyle ve dış dünya ile etkileşimini basitleştirmeyi amaçlayan bir yazılım platformudur. Temel amacı kullanıcı, işletim sistemi (Ubuntu) ve ekipmanlar (sensör, kamera, joystick, vb) ile etkileşimi sağlamaktır. Bir işletim sisteminde donanım ile ilgili nasıl bir soyutlama var ise ROS içinde bu soyutlamadan bahsedebiliriz. Robotun tüm donanımsal detaylarına girmeden kontrol imkanı sunar. Örneğin bir robot kolunu hareket ettirmek için, robot kolunun donanımsal tasarımını yapanların phyton, c++ da hazırlamış olduğu kütüphaneler içindeki komutları kullanırsınız. ROS dağıtık mimari yapısını destekler. Robotun kamera görüntülerinden haritayi çıkartan ve gideceği yolu oluşturan yazılım ve donanım başka bir makinada iken, robotun tekerleğini hareket ettiren modülü başka bir makinada olabilir.  
  
ROS sürümleri (ROS paketlerinin versiyonlanması.) kar amacı gütmeyen Open Source Robotics Foundation (OSRF) kontrolünde yayınlanır. ROS hakkında detaylı bilgi edinebileceğiniz sitesi [organizasyon](http://www.ros.org/)linkinde sunulmuştur. Sürümlere ait detaylı bilgiyi [dağıtımlar](http://wiki.ros.org/Distributions) linkinden bulabilirsiniz.  
  
Ubuntu (Xenial) üzerine ilk kurulumu son sürüm olan (tavsiye edilen) Kinetic sürümü ile gerçekleştirdim. Lakin ROS un Indigo sürümünde kurulum paketi ile birlikte gelen Gazebo robot simulasyon programı desteği bu yazıyı yazarken Kinetic için yoktu ve bende öğrenme aşamasında olduğum için, iş gazebo simülatör kismina gelince kaldım. Bunun üzerine ROS Indigo sürümünü, Ubuntu 14.4 Trusty üzerine sil baştan kurmaya karar verdim.  
  
Microsoft ortamında yazılım geliştirmeye aşina biri olup linux konusunda temel seviyede bilgi sahibi olmama rağmen kurulumu [sitesindeki](http://wiki.ros.org/ROS/Installation)adımları izleyerek Kinetic için kolayca, Indigo için biraz uğraşarak gerçekleştirebildim. Sebebi ise  gazebo' ya ait modellere normal kurulumla erişemedim. Tüm modelleri ayrıca indirmek zorunda kaldım. Kinetic'den Indigo dağıtımına Gazebo simülatörünü kullanayım diye yola çıkmış ve böyle bir espri ile karşılaşmış olduk.  
  
Sitedeki linklerin ROS sürümlerine göre bir yapılanması mevcut.  
  
http://wiki.ros.org/**<sürüm adı>**/Installation/Ubuntu adresindeki **<sürüm adı>** değişkenine Kinetic yazarsaniz Kinetic sürümüne ait kurulum sayfalarına gidersiniz. Biz Indigo kurulumuna gideceğiz.  
  
Şimdi ROS [sitesinde](http://wiki.ros.org/indigo/Installation/Ubuntu)Indigo kurulum talimatlarını aşağıdaki gibi terminal üzerinde sırasıyla işletelim.  
  
ROS paketlerini alabilmek için Ubuntu program paketleri listesine (sources.list) packages.ros.org sitesini ekleyelim. Aşağıdaki tek bir komut satiridir.

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

listeyi güncellemek için ROS deposuna ait anahtar bilgilerini girelim.

sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116

ROS kurulumuna geçmeden paketlerimizi güncelleyelim

sudo apt-get update

Şimdi ROS  ve yanında çeşitli yardımcı araçlar (rviz, rq, gazebo 2, turtlesim simulator, robot kütüphaneleri) olan paket kurulumuna geçelim. Ek bilgi olarak, ROS sürümlerinde gördüğümüz kaplumbağa (Ilk sürüm olan ROS Box Turtle'in her sürümde evrimleşmiş hali) turtlesim simulator çalıştırıldığında karşınıza çıkar.

sudo apt-get install ros-indigo-desktop-full

ROS platformunu çalıştırmak için gerekli olan ROS harici işletim sistemi bağımlılıklarını yüklemek  için aşağıdaki komutlari çalıştırıyoruz.

sudo rosdep init

rosdep update

ROS platformuna ait komutların tanınması ve çalıştırılması için ROS'a ait dosyalarin yollarının bilinmesi  gerekir. Her bir terminal oturumunda (terminal ekrani açtığınızda) bunun otomatik yapılabilmesi için aşağıdaki komutları çalıştırıyoruz. (terminalde bash kabuğu kullanıyorsak.)

echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc

source ~/.bashrc

rosintall komutunu ros paketlerini indirmek için kullanırsınız. Bu komutu kullanabilmek için sisteme kurmaniz gerekmektedir. Bunuda aşağıdaki komutla yaparız.

sudo apt-get install python-rosinstall

kurulumu bu şekilde tamaladıktan sonra aşagıdaki komutla ROS sistem değişkenlerini kontrol edebilirsiniz.

export| grep ROS

declare -x ROSLISP\_PACKAGE\_DIRECTORIES=""

declare -x ROS\_DISTRO="indigo"

declare -x ROS\_ETC\_DIR="/opt/ros/indigo/etc/ros"

declare -x ROS\_MASTER\_URI="http://localhost:11311"

declare -x ROS\_PACKAGE\_PATH="/opt/ros/indigo/share:/opt/ros/indigo/stacks"

declare -x ROS\_ROOT="/opt/ros/indigo/share/ros"

Eğer değişkenleri yukarıdaki gibi göremedi iseniz setup.bash dosyasini tekrar source etmeyi denemelisiniz. Indigo sürümünü kurduğumuz için **ROS\_DISTRO="indigo"** olduğunu gözlemleyiniz.  
  
ROS Indigo kurulumumuzu böylelikle tamamladık. Şu aşamada terminal ekranını açıp ROS komutları girebilirsiniz.   
  
ROS üzerinde yazılım geliştirecekseniz (ROS paketleri oluşturmak veya var olan ROS paketleri değiştirip derlemek)  catkin çalışma ortamını kurmanız gerekmektedir.  
  
Açık kaynaklı yazılım geliştirme kalıpları, yeni problemleri çözmek için mevcut çözümleri tekrar kullanma kolaylığı nedeniyle güçlüdür. ROS'da bu mantığı kullanır. Birçok robotik uygulaması, diğer robot uygulamalarda karşılaşılan problemleri içerir veya çok benzer alt sistemlere sahiptir. Çözüm için diğer paketleri kullanmak beraberinde örgütsel zorlukları getirir. Her bir paketin kendine ait derleme şekli, bağımlı oldukları kütüphaneler bulunur.  catkin bu zorlukların üstesinden gelmek için oluşturulmuş bir dizin yapısıdır. Kurulumu için aşağıdaki komut satırlarını çalıştıralım.

$ mkdir -p ~/catkin\_ws/src

$ cd ~/catkin\_ws/src

$ catkin\_init\_workspace

catkin\_ws dizinine baktığınızda /src dizininin oluştuğunu ve içinde CMakelist.txt dosyası olduğunu göreceksiniz. catkin\_make komutunu aşağıdaki gibi çalıştırdığınızda

$ cd ~/catkin\_ws/

$ catkin\_make

/build  
/devel dizinlerinin ve //devel dizininin içinde ise setup.\*sh dosyalarını göreceksiniz. Bunlardan setup.bash dosyasini source komutu ile çalıştirarak ROS sistemine dahil ediyoruz.

$ source ~/catkin\_ws/devel/setup.bash

$ echo "source ~/catkin\_ws/devel/setup.bash" >> ~/.bashrc

İşlemi kontrol için aşağıdaki komutu çalıştırarak catkin dizininin en başa eklendiğini gözlemleyiniz.

$ echo $ROS\_PACKAGE\_PATH

Böylece ROS kurulumuna catkin kurulumunu da ekleyerek kurulum safhasini tamamlamış olduk.  
  
Terminal penceresini açarak ilk komutumuz olan roscore komutunu girelim. Bu komut ROS sisteminin çalışması için gereken modül ve programların yüklenmesini sağlar. Bunlar sırası ile ROS Master, ROS Parameter service, rosout loglama modülleridir.

$ roscore

ROS platformunda aynı veya farkli bilgisayarlarda çalişan her bir kod modülü Node (düğüm) olarak geçer. Her bir düğüm robotunuzla ilgili özelleşmiş bir iş yapar. Bir düğüm oluşturmanın iki yolu vardır; birincisi terminal üzerinden ikincisi ise phyton veya c++ kodu ile düğüme ilişkin kodlamayla olur. Düğümler birbirleriyle mesajlaşabilir. Mesajlaşma ancak düğümlerin kendilerini ROS platformuna kaydettirmeleri ile mümkündür. Düğümlerin kaydedildiği ve izlendiği modül ROS Master olarak geçer. Düğümler arası haberleşme TCP/IP veya TCPROS kullanılarak yapılır. Yukarıdaki roscore komutu ile ROS Master'i devreye almış oldunuz.  Terminal ekraninda komut cevabını incelediğinizde aşağıdaki URI, master'in lokasyonunu belirtir. <makina\_ismi\_veya\_ip> değişkeni bende osboxes gozukuyor. Bu sanal makinamin ismi. Sizde de kendi makinanizin ismi gozukecektir.

ROS\_MASTER\_URI=http://<makina\_ismi\_veya\_ip>:11311/

Biraz önce açtığınız ve roscore komutunu çalıştırdığınız terminali kapatırsanız ROS sistemide kapanır. Terminali kapatmadan ctrl + c ile de ros mater modulunu kapatabilirsiniz.  
  
Yukaridaki URI bilgisine bu haliyle müdahale etmez isek network ortamında birden fazla makinaya dağilmiş bir yapıda düğümlerin master'i bulması ve master'in düğümleri izlemesinde sıkıntı yaşayabilirsiniz. Matlab üzerinden (robotik paketi içerir ve ros platformu desteği verir.) Ubuntu kurulu makinaya erişirken  (ros platformu kurulu makinam) bu sıkıntıyı yaşadım. Çözüm olarak düğümlerin birbirlerini URI üzerinden çözmesini kolaylaştırmak için ROS\_MASTER\_URI ve ROS\_IP platform değişkenlerini uygun ip değerleri ile her bir makina da set etmeniz gerekir.  
  
ROS Master modülünü kapatarak aşağıdaki komutlarla önce değişkenlerin varsayılan değerlerine bakmak istiyorum.  
  
ROS\_MASTER\_URI : roscore çalıştırılan makinanin ip adresini barindirmali. Ros platformunda birden fazla düğüm olabilir ama tek bir master vardir. Bizim örneğimizde master ve düğümlerin tek bir makinada olduğuna dikkat edelim.

echo ROS\_MASTER\_URI

komut cevabı olarak  herhangi bir değer göremedim. Bu makinada yani  192.168.1.7 nolu makinada ros master modülünün çalışmasını istediğimden dolayı aşağıdaki gibi komutu çalıştırıyorum.

echo export ROS\_MASTER\_URI=http://192.168.1.7:11311

yukarıdaki işlemi terminali açıp roscore çalıştırmadan önce her seferinde yapmanız gerekir. Terminal her açıldığında bu bilgileri kendisi alsın istiyorsanız bunu bash kabuğuna yazmalısınız.

echo export ROS\_MASTER\_URI=http://192.168.1.7:11311 >> ~/.bashrc

ROS\_IP : Bulunduğunuz makinanın ip adresidir. Düğümleriniz bu bilgiyi kullanarak kendilerini master 'a kaydederler.

echo ROS\_IP

komut cevabı olarak  herhangi bir değer göremedim. Bu makinada yani 192.168.1.7 de olduğum için komutu aşağıdaki gibi yazarak çalıştırıyorum.

echo export ROS\_IP=192.168.1.7 >> ~/.bashrc

Bu aşamadan sonra roscore komutuyla ROS master modulunu tekrar çalıştırıp yeni master URI bilgisini gözlemleyiniz.  
  
Aşağıdaki komutla .bashrc dosyası içinde en altta export işlemlerimiz ve source işlemlerimizin nasıl dosyaya eklendiğini görebilir ve IP, URI  değişikliklerini artık buradan da yapabilirsiniz.

gedit ~/.bashrc

Eğer sanal makinada ubuntu kurmuşşsaniz sanal makinanizda network bağlantı yöntemini bridge olarak düzenlemeniz gerekir.  Ayrıca güvenlik duvarı kullanıyorsanız (firewall) ilgili portlara erişim izni vermeniz gerekmektedir.  
  
Böylelikle ROS nedir ve kurulumu nasıl yapılır sorularına cevap vermiş olduk. Bir sonraki [yazımızda](https://kodkarma.blogspot.com.tr/2017/01/matlab-robotik-paketi-ve-ros.html)temel ros komutlarını kullanarak iki makinalı bir sistemde matlab üzerinden ROS 'a erişmek ve haberleşmeyi göstermek adına bir örnek yapacağız.

Gönderen [murat pekcan](https://plus.google.com/109835580729700522850)zaman: [16:32](https://kodkarma.blogspot.com.tr/2017/01/ros-nedir-ve-nasl-kurulur.html)

[Bunu E-postayla Gönder](https://www.blogger.com/share-post.g?blogID=4877635516036198927&postID=3325514939041006105&target=email) [BlogThis!](https://www.blogger.com/share-post.g?blogID=4877635516036198927&postID=3325514939041006105&target=blog" \o "BlogThis!" \t "_blank) [Twitter'da Paylaş](https://www.blogger.com/share-post.g?blogID=4877635516036198927&postID=3325514939041006105&target=twitter" \o "Twitter'da Paylaş" \t "_blank) [Facebook'ta Paylaş](https://www.blogger.com/share-post.g?blogID=4877635516036198927&postID=3325514939041006105&target=facebook" \o "Facebook'ta Paylaş" \t "_blank) [Pinterest'te Paylaş](https://www.blogger.com/share-post.g?blogID=4877635516036198927&postID=3325514939041006105&target=pinterest" \o "Pinterest'te Paylaş" \t "_blank)

Etiketler: [Matlab](https://kodkarma.blogspot.com.tr/search/label/Matlab), [ROS](https://kodkarma.blogspot.com.tr/search/label/ROS)

1 yorum :

1. https://lh3.googleusercontent.com/zFdxGE77vvD2w5xHy6jkVuElKv-U9_9qLkRYK8OnbDeJPtjSZ82UPq5w6hJ-SA=s35