

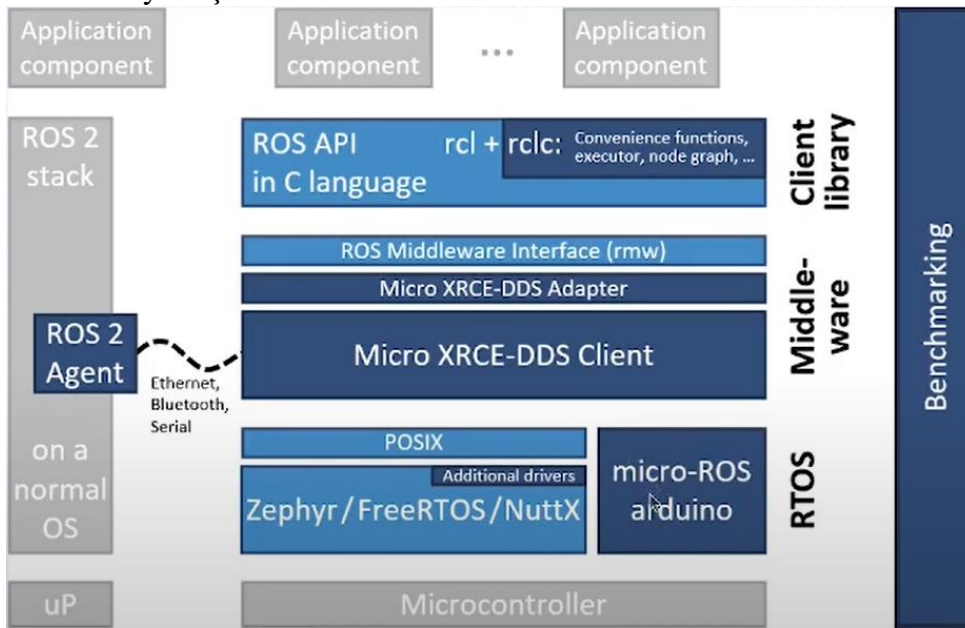
Robot Operating System (ROS2)

İnsan ile robot arasında iletişimi sağlayan açık kaynak kodlu bir yazılımdır. Gerçekten sağlam, genel amaçlı robot yazılımı oluşturmak zor. Robotun bakış açısından, insanlar için önemsiz görünen problemler genellikle görevler ve ortamlar arasında çılgınca değişir. Bu varyasyonlarla uğraşmak o kadar zordur ki, hiçbir birey, laboratuvar veya kurum bunu tek başına yapmaya çalışmaz.

Gazebo, Ros simülasyon ortamında kullanılabilen robot davranışlarını simüle eden yapıdır. Yani biz ROS da Gazebo yu kullanırız. Bu ortamlarda, kullanılacak robotun dış kısımları dahil olmak üzere tüm aksanlarını oluşturabiliriz.

Biz bir işlemci kullanacağımız için micro ROS kullanacağız. Bu işlemcimizin ROS ekosistemine bağlanmasını sağlayan yapıyı oluşturacaktır.

ROS 1 de bir serial haberleşme ile işlemcilerle iletişim kurabiliyordu fakat bağlantı problemleri başta olmak üzere birçok hata ortaya çıktı. ROS 2 de ise serial,bluetooth,ethernet bağlantılarıyla iletişim kurulabiliyor. Aynı zamanda ROS2 ile birlikte micro ROS ortaya çıktı ve bağlantı hızı paket kaybı gibi sorunlar düzeltildi. Yani Data Distribution Service protokolü ile endüstride kullanılmaya başlandı.



Orta kısım İşlemcimizi sol kısım işe bilgisayarımızı belirtir. İşlemciye göre uygun olan yapıyı(Zephyr,FreeRTOS,NuttX,POSIX,micro-ROS arduino) kurup bunu Middleware seviyesinde bilgisayarımız ile iletişime sokacağız. Bunun için Bilgisayarımıza bir ROS paketi kurmalıyız. Bilgisayarımıza kurduğumuz ROS a veri aktarımı yapabilmek için işlemcimiz(microROS) ile bilgisayarımız(ROS2) arasında bağlantı kurabilmek için aynı zamanda bilgisayarımıza 'ros agent' yapısını da kurmalıyız. Bu işlemlerin nasıl yapılacağı en detaylı şekilde verilmiştir. Bu dökümana gitmek için [tıklayınız...](#)

Verilen dökümanda tüm ROS versiyonları bulunmaktadır. Bunlardan size uygun olanı seçiniz ve installation kısmında detaylı olarak nasıl indirileceği belirtilmiştir.

İşlemcimize microROS, bilgisayarımız ROS ve ros agent i kurduk. Artık bilgisayarımız ile işlemcimiz arasında veri aktarımına gerçekleştirebiliriz. ROS ile işlemcimizin veri aktarımını rosserial veya micro-ROS gibi paketler kullanılabilir.

rosserial: UART üzerinden verileri ROS 2'ye aktarmanızı sağlar.

micro-ROS: STM32 gibi düşük seviyeli donanımlar için optimize edilmiş bir ROS sürümüdür ve daha gelişmiş veri iletişim imkanları sunar.

GAZEBO ve RVIZ

Gazebo bize bir robotun hareketlerinin bütünü ve kabiliyetlerini sunar, RVIZ işe bu robotun her bir ekleminin tek başına nasıl bir hareket kabiliyetine sahip olduğunu gösterir.

URDF, bir robotun yapısını ve fiziksel özelliklerini tanımlayan XML tabanlı bir format. Gazebo'ya bu URDF formatındaki robotları eklemek için ek gazebo etiketleri kullanılmalı ve bazı fiziksel parametreler (inertia, friction, kütle merkezi vb.) doğru biçimde tanımlanmalıdır. Ayrıca ek gazebo etiketlerinden (inertia, gazebo, link ve joint) bulunmaktadır. URDF'nin bazı kısıtlarını ve daha gelişmiş tanımlamalar için SDF (Simulation Description Format) kullanımını öneriliyor. Gazebo .urdf formatı kendisi sdf formatına çeviriyor. Çünkü .urdf sadece robot parçalarındaki bölümlerin kinematiklerini belirtir, fakat bir robotu dünyada çalıştıramaz ve .urdf paralel işlem yapamaz, evrensel bir tanımlama değildir.

Bu sebeple SDF formatı gazebo da kullanılır. Bu format bir robotun dünyadaki tüm hareketlerini içeren bütüncül bir format türüdür.

Simulasyon formatında <gazebo> eklersin ve buranın içinde her parça için fonksiyonun ve yapısını belirtirsin. Üç farklı <gazebo> elementi var. Bunlar <robot>, <link> ve <joint> dir. İleride bunların açıklaması yapılacaktır.

.urdf formatının gazebo'da kullanımı için detaylı bilgiyi [bulabilirsiniz...](#)

Gazebonun ROS integrasyonu ile ilgili detaylı bilgiyi [bulabilirsiniz...](#)

- joint yapısı RVIZ de olan ve manuel olarak eklemeleri değiştirebildiğimiz yapıyı ifade eder. (joint 1 , joint 2 , ...)

- Robotu gazebo'ya dönüştürürken , RVIZ ve ROS daki işlemleri kapatmamalıyız. Robotun hala çalıştığını görmek için Rviz de test etmek önemlidir.

- ‘ .xacro ‘ formatı, urdf formatının çok daha esnek formatıdır (fonksiyon gibi). Bir dört tekerlek için .urdf formatı ile 4 defa aynı kod parçacığını ve oluşturmalıyız fakat .xacro ile 4 defa tanım yaparız ve 4 tekerlek oluşur gibi bir düşünce düşünülebilir.

- .gazebo dosyası sadece gazebo'da görülür. Normalde iki tane xacro tanımlanır. Birisi Rviz diğeri Gazebo.

ROS2 – GAZEBO Formatının Entegrasyonu

Biz kullandığımız husky robotu için bu sitedeki veriyi kullandık...

Kullandığımız drone için bu sitedeki veriyi kullandık...

Hem dronun hem de husky nin aynı ortamda kullanılması (herhangi iki, üç veya çok daha fazla modelin bir arada kullanılması da aynı mantıkla ilerliyor) için her aracımızı ayrı ayrı build etmek veya tek bir klasörde ortak olarak build etmek (kullanıcının seçimine kalıyor) gerekli. Bununla birlikte her bir modelin en temelde bringup (kontrolcüler, simulasyon ortamı ve diğer parametreleri başlatan launch dosyaları), control klasörü ve description klasörü vardır. Bu klasörler drona ait olan tüm verileri içerir. Fakat gazebo'da simulasyon olarak görmeden önce Rviz de sensörlerin, eklemelerin, 3d modeli gibi fiziksel yapısını oluşturan yapının test edilmesi gerekir.

Her model başlatıldığı vakit, tanımlanan topic'leri üzerinden haberleşme sağlar. Bazı topic'ler publisher bazıları subscriber olarak oluşturulur. Gazebonun başlatılması ile tüm bu topic'ler oluşturulur ve yazılan script ile bu topic'lerin bağlantısı yapılıyor.