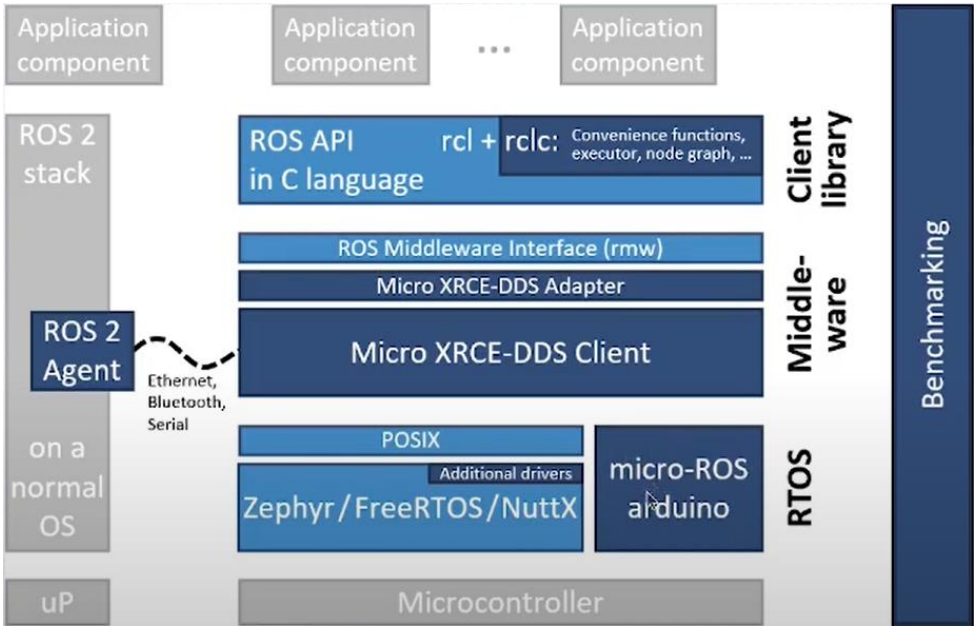
Robot Operating System (ROS)

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 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 işlemleri bir mikroişlemci(STM32) de yapacağımız için microROS 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 microROS ortaya çıktı ve bağlantı hızı paket kaybı gibi sorunlar düzeltildi. Yani Data Distribution Service protokolü ile endüstridede kullanılmaya başlandı.



Orta kısım İşlemcimizi sol kısım işe bilgisayarımızı belirtir. İşlemciye göre uygun olan yapıyı(Zephyr, FreeRTOS, NuttX, POSIX, micro-ROS ardunio) 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 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.

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 bütününü 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 kısımlarını ve daha gelişmiş tanımlamalar için SDF (Simulation Description Format) kullanımını öneriliyor.

Gazebo **.urdf** formatı kendisi **sdf** formatına çeviriyor. Çünki .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 gazeboda kullanımı için detaylı bilgiyi bulabilirsiniz...

Gazebonun ROS integrasyonu ile ilgili detaylı bilgiyi bulabilirsiniz…