

Formation : ROS 2, Robot Operating System

Développez des applications robotiques

Formation pratique - 3j - 21h00 - Réf. ROZ

Prix : 2470 CHF H.T.

NEW

ROS 2 (Robot Operating System) est aujourd'hui le framework le plus utilisé pour la création d'applications robotiques. Cette formation vous permettra de comprendre son architecture modulaire et d'acquérir les compétences nécessaires pour concevoir des applications Python ou C++ pour le pilotage de robots mobiles, de bras manipulateurs, et l'intégration d'algorithmes basés sur l'IA à un système robotisé.

Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Comprendre l'architecture modulaire des applications ROS 2
- ✓ Connaître les outils de débogage pour avancer sur son projet
- ✓ Mettre en place une navigation autonome sur un robot mobile
- ✓ Simuler des capteurs, des actionneurs et un environnement dans lequel évoluer
- ✓ Intégrer un traitement par IA dans un robot

Public concerné

Développeurs divers, ingénieurs en robotique, ingénieurs en systèmes embarqués, chefs de projet techniques.

Prérequis

Bonne connaissance en Python et C++.

Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

Programme de la formation

PARTICIPANTS

Développeurs divers, ingénieurs en robotique, ingénieurs en systèmes embarqués, chefs de projet techniques.

PRÉREQUIS

Bonne connaissance en Python et C++.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

1 Introduction à ROS 2

- Vue d'ensemble de l'écosystème ROS.
- Panorama de ses briques technologiques et de ses frontières.
- Le développement communautaire et les communautés du projet.

Travaux pratiques

Manipuler des paquets ROS existants avec le CLI pour comprendre leur architecture, inspecter le graphe d'exécution (rqt) et rejouer un enregistrement (ros2 bag).

2 Concepts techniques clés

- Structure des projets ROS, l'URDF, les systèmes de coordonnées et TF2.
- Le middleware RMW, ses implémentations, et le contrôle QoS.
- L'interface en ligne de commande (CLI).
- Les outils d'introspection et de débogage : visualisation du graphe et des données (rqt, RViz2), niveaux de logs.
- Enregistrement et rejou de scènes (ros2bag)

Travaux pratiques

Manipuler des paquets ROS existants avec le CLI pour comprendre leur architecture, inspecter le graphe d'exécution (rqt) et rejouer un enregistrement (ros2 bag).

3 Développer du code avec ROS

- Les bibliothèques clientes (RCL) pour Python et C++.
- Les types de nœuds Node, Lifecycle et Composable, et leurs exécuteurs.
- Les arbres de comportement BT.cpp et Groot2.

Travaux pratiques

Créer et compiler des nœuds simples Python et C++ consommant le RMW.

4 Intégrer un système ROS 2

- Les caractéristiques d'un système bien intégré à ROS 2.
- Les types de messages standards pour les capteurs fréquents (Lidars, Sonars, Force Torque...).
- La modularité des packages et des interfaces de messages.
- La simulation de capteurs, d'actionneurs et d'un environnement avec Gazebo.

Travaux pratiques

Observation, dans le simulateur Gazebo, de l'interface ROS 2 des capteurs et actionneurs d'un robot simulé.

5 La navigation autonome Nav2

- Les concepts de la navigation : cartographie, planificateur, suiveur.
- Localisation Monte-Carlo AMCL (Adaptive Monte Carlo Localization).
- Localisation et cartographie simultanées (SLAM, Simultaneous Localization and Mapping).

Travaux pratiques

Mise en œuvre d'une navigation autonome au Lidar commandée par un nœud Python et C++.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse psh-accueil@orsys.fr.

6 La manipulation par bras robotique MoveIt 2 et intégration de l'IA

- Architecture de MoveIt 2.
- Les solveurs de cinématique directe et inverse, et les contraintes optionnelles.
- Les solveurs de trajectoires avec évitement de collisions (OMPL, Open Motion Planning Library).
- Utilisation d'assistants de code pour la production de nœuds Python ou C++.
- Intégration d'un algorithme d'IA dans un nœud (DRL, TOD, DNN...).

Travaux pratiques

Mise en œuvre d'un bras commandé par un nœud Python et C++.