



## Conception robotique avec ROS

RT020

Durée: 3 jours

2 310 €

5 au 7 février  
5 au 7 mai

7 au 9 juillet  
24 au 26 novembre

### Public :

Chefs de projet robotique, développeurs et toute personne souhaitant utiliser le système ROS pour concevoir une application robotique.

### Objectifs :

Comprendre l'architecture de ROS, savoir choisir et mettre en oeuvre les composants adaptés à un projet robotique. Les travaux pratiques sont réalisés avec ROS 2

### Connaissances préalables nécessaires :

Connaissance générale des systèmes d'informations, d'un langage de développement comme Python ou C++.

### Programme :

#### Introduction

Présentation de ROS (Robot Operating System), historique, contraintes, fonctionnalités, licence.  
Objectifs des créateurs de ROS.  
Point sur les dernières distributions de ROS.  
Tour d'horizon des robots compatibles.  
Exemples d'utilisation de ROS.

#### Architecture

Principe de fonctionnement de ROS : modularité, légèreté, compatibilité  
Les différents éléments : les noeuds, le master, les topics, les messages, les services  
L'organisation des fichiers, l'installation : les packages, les stacks  
Présentation packages disponibles.

#### Mise en oeuvre

Installation sur Ubuntu.  
Configuration de l'environnement.  
Présentation du filesystem ROS et de roscd, eosis, rospack  
Premier pas : création d'un package.  
Gestion des noeuds : utilisation de roscore, rosnodet et rosrund.  
Création de topics et de services.

#### Développement



# — Phirio —

---

Programmation en python d'outils de publication et réception de messages (topics), et de services et clients simples.

## Exploitation

---

Utilisation de fichiers .bag pour stocker des messages et les réutiliser  
Gestion des dépendances entre packages  
Exécution de ROS sur une infrastructure de plusieurs serveurs.

## Extensions

---

Présentation des librairies principales