



# Phirio

## Cycle certifiant Développeur robotique

RT050

Durée: 14 jours

8 810 €

### Public :

Chefs de projets, ingénieurs et concepteurs robotiques

### Objectifs :

Savoir concevoir un système robotique de la conception avec ROS jusqu'à la vision et détection et reconnaissance d'objets avec OpenCV et des solutions d'IA.

### Connaissances préalables nécessaires :

Connaissance générale des systèmes d'informations, d'un langage de programmation comme python.

### Programme :

#### Objets connectés

Objets connectés

#### Les objets connectés

L'internet des objets. Les acteurs et produits du marché.  
Définitions, fonctionnement et architectures.

#### Identifier les usages et les technologies associées

Applications, services fournis par les objets connectés : domotique, santé, loisirs.  
Plateformes matérielles (Intel, Samsung) et logicielles. Les systèmes classiques android, IOS, systèmes embarqués : UI, Brillo, LiteOS, Galileo, Arduino  
Infrastructure de connectivité : wifi, bluetooth, BLE, zigbee, NFC, RFID, LoRa, 4G, 5G  
Protocoles : MQTT, AMQP, Coap, LoRaWan.

Atelier : mise en oeuvre d'une antenne Lora et connexion d'objets

#### Evaluer les opportunités économiques et d'innovations apportées par le domaine de l'IoT

Présentation des écosystème IoT. Impacts sur les métiers : concepteurs d'objets, fabricants de composants, opérateurs et équipements réseaux, acteurs de la sécurité et intégrateurs, fournisseurs de services et stockages, opérateurs de cloud

Atelier : démonstration d'une mise en oeuvre sur une habitation connectée



# Phirio

---

## Appréhender les normes et standards en vigueur dans le domaine de l'IoT

---

Protocoles: mqtt, bluetooth, wifi, 3G/4G, etc ...  
Avec un autre objet : M2M,  
Avec le réseau internet : vers un serveur, vers le cloud.  
Les plateformes IoT, définition de standards : OpenInterconnect Consortium.

Atelier : mise en place d'une passerelle LoRa/MQTT avec mosquitto

Atelier : interconnexion AWS/IoT et MQTT. APIs en C, java et python

Atelier : Création d'un compteur d'actions sur smartphone

---

## Savoir anticiper l'intégration de l'IoT dans une entreprise

---

Types de données collectées. Données locales, limites.  
Transfert et stockage sur une autre machine, dans le cloud ou sur internet. Analyse des données et fourniture de services associés.  
Techniques de protection des objets connectés. Authentification et autorisations. Protection des données, législation. Qualité de service coté client.  
BigData et IoT : de l'objet connecté à l'analyse de nuage d'objets.

Atelier : démonstration sur une chaîne de traitement  
LoRa/MQTT/Kafka/SparkStreaming/Hadoop

Atelier : mise en place d'une sécurisation SSL/TLS sur les flux IoT

---

## Conception

---

Conception

---

## Conception robotique avec ROS

---

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.



# Phirio

---

## 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, rosnode et rosrnun.  
Création de topics et de services.

---

## Développement

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

---

## Vision

Vision

---

## Computer Vision avec OpenCV

Présentation OpenCV  
Historique, fonctionnalités, versions, licence  
Site de référence, documentation disponible  
Principes de base de la vision par ordinateur  
Positionnement OpenCV par rapport aux autres solutions du marché

---

## Mise en pratique OpenCV

Installation d'OpenCV.

Atelier : Exemples simples : lecture, affichage, enregistrement d'images



# Phirio

---

## Manipulation d'images

bases de traitement d'images :  
Opération sur les matrices.  
les espaces de couleurs : CMYB, HSV, niveaux de gris;  
la segmentation d'images;

**Atelier : Exemple de seuillage avec OpenCV**

Conversion d'une image en binaire  
Histogramme d'une image.  
Les opérateurs binaires (NO, AND, OR, XOR):  
utilisation pour la fusion d'images

---

## Classificateurs et IA

Principe, différents classificateurs disponibles avec OpenCV  
Exemples : détection de formes, de contours, de visages  
Deep Learning avec PyTorch

**Atelier : Classification d'images sur machine Jetson**

**Atelier : Détection d'objets**

---

## Simulation

Simulation

---

## Simulation avec Gazebo

Présentation du projet Gazebo, licence, versions  
Fonctionnalités: simulation d'un robot physique avec interactions avec l'environnement  
Modèles de robots disponibles  
Compatibilité avec ROS et Player  
Exemples d'utilisation: compétitions DARPA, NASA, ...

---

## Mise en oeuvre

Prérequis techniques  
Installation sur Ubuntu  
Configuration des variables d'environnement  
Architecture de Gazebo : Master, bibliothèques de communication, physiques, visualisation 3D, etc ...  
Moteurs utilisés : ODE, Bullet, Simbody, DART  
Premiers pas : utilisation de l'interface graphique,  
choix de modèles, gestion de l'environnement

---

## Modèles, robots

Utilisation du Model Editor pour construire des modèles  
Construction de véhicules, ajout de capteurs, de plugins  
Utilisation des modèles, création de robots, traitement des données des capteurs, gestion de l'environnement physique



# Phirio

---

## Extensions

Intégration des packages ROS : gazebo\_ros\_pkgs pour simuler un robot avec gazebo en utilisant les messages, services de ROS.  
Simulation sur AWS avec CloudSim

---

## Analyse des données

Analyse des données

---

## Positionnement Python

Les valeurs d'observation, et les variables cibles.  
Ingénierie des variables.  
Analyses statistiques,  
Classification des données, rapprochements,  
Production de recommandations. Evolutions des outils statistiques classiques vers l'apprentissage automatique.

Atelier : exercices sur les outils statistiques de base

---

## Savoir utiliser les principaux outils de traitement et d'analyse de données pour Python

Besoins des data-scientists : calculs, analyse d'images, machine learning, interface avec les bases de données  
Apports de python : grande variété d'outils, expertise dans le domaine du calcul scientifique  
Présentation des outils d'apprentissage Python : scikit-learn, pybrain, TensorFlow/keras, mxnet, caffe

Atelier : mise en oeuvre de scikit-learn et génération de jeux de données.

---

## Être capable d'extraire des données d'un fichier

Pandas : manipulation de tables de données. Notion de dataframe.  
Manipulation de données relationnelles  
Tableaux avec Pandas: indexation, opérations, algèbre relationnelle  
Stockage dans des fichiers: CSV, JSON

Atelier : construction d'ETL de base entre json et csv

---

## Savoir appliquer les pratiques optimales en matière de nettoyage et de préparation des données avant l'analyse

Encodeurs  
Filtres et ETL  
Gouvernance des données. Qualité des données.  
Transformation de l'information en donnée. Qualification et enrichissement.  
Sécurisation et étanchéité des lacs de données.  
Flux de données et organisation dans l'entreprise. De la donnée maître à la donnée de travail. MDM.  
Mise en oeuvre pratique des différentes phases :  
nettoyage, enrichissement, organisation des données.

Atelier : construction d'un système de détection de contours



# Phirio

---

## Apprendre à mettre en place un modèle d'apprentissage simple

---

Les différentes méthodes : apprentissage supervisé, apprentissage automatique.  
Algorithmes : régression linéaire, k-voisins, classification naïve bayésienne, arbres de décision, ...

Atelier : classifieurs. scoring

APIs fournies en standard, modèles d'apprentissage  
Projet scikit-learn : classification, régression, validation de modèles prédictifs.  
Démonstrations avec les modèles fournis par scikit-learn  
Positionnement et comparaison avec Keras, mxnet, caffe

Atelier : codage d'une reconnaissance d'animaux avec une forêt aléatoire

---

## Choisir entre la régression et la classification en fonction du type de données

---

Présentation des types de données en entrées : données discrètes, données continues. Labelisation, mapping par fonction.  
Comprendre les algorithmes : régression linéaire, k-moyennes, k-voisins, classification naïve bayésienne, arbres de décision, forêts aléatoires, ...  
Critères de choix des algorithmes.

Atelier : construction d'un système décisionnel fondé sur des forêts aléatoires

---

## Évaluer les performances prédictives d'un algorithme

---

Les courbes d'apprentissage. Définitions : AUC, courbes ROC.  
Comprendre le principe des hyper-paramètres. Choix des hyper-paramètres.

Atelier : calcul et visualisation d'une matrice de confusion

Atelier : Visualisation de courbes d'apprentissage fonction des hyper-paramètres

Atelier : Visualisation d'une mise en sur-apprentissage

---

## Créer des sélections et des classements dans de grands volumes de données pour dégager des tendances

---

Présentation de pyspark  
Machine learning et deep learning  
TensorFlow: principe de fonctionnement, plateformes supportées, distribution,

---

## Analyse d'images avec TensorFlow et Keras

---

Introduction au traitement d'images et à l'apprentissage automatique, les apports de l'IA.  
Cas d'applications : analyse, tri d'images, détection d'objets, reconnaissance faciale, génération d'images, etc.  
Présentation de Keras, PyTorch et OpenCV : principes de fonctionnement, caractéristiques, points forts.



# Phirio

---

## Présentation des RN

---

- Principe des réseaux de neurones
- Différents types de couches: denses, convolutions, activations
- Fonctionnement des réseaux de neurones convolutifs (CNN).
- Descente de gradient
- Multi-Layer Perceptron

---

## Le projet Tensorflow et Keras

---

- Historique , fonctionnalités
- Architecture distribuée, plateformes supportées
- Principe des tenseurs, caractéristiques d'un tenseur: type de données, dimensions
- Définition de tenseurs simples,
- Gestion de variables et persistance,
- Représentation des calculs et des dépendances entre opérations par des graphes

---

## Mise en oeuvre avec Keras

---

- Conception d'un réseau de neurones
- Création et entraînement d'un modèle CNN simple avec Keras.
- Classification d'images avec Keras
- Notion de classification, cas d'usage
- Architectures des réseaux convolutifs, réseaux ImageNet
- RCNN et SSD
- Démonstrations

---

## Optimisation d'un modèle

---

- Visualisation avec Tensorboard
- Optimisation des couches de convolutions
- Choix des hyper-paramètres avec Keras et Keras Tuner
- Utilisation de checkpoints

---

## Segmentation d'Images avec PyTorch

---

- Comprendre la segmentation d'images.
- Création d'un modèle de segmentation convolutif avec PyTorch.
- Préparation des données d'entraînement pour la segmentation.
- Entraînement et évaluation des performances du modèle.

---

## Détection d'Objets avec OpenCV et IA

---

- Principes de la détection d'objets.
- Les différents types de modèles de détection d'objets (classificateurs en cascade, YOLO, SSD, Faster R-CNN, etc.).
- Mise en oeuvre d'OpenCV pour la détection d'objets.
- Introduction aux classificateurs en cascade d'OpenCV pour la détection d'objets.
- Présentation des modèles IA pré-entraînés pour la détection d'objets.
- Comparaison des différents modèles disponibles (YOLO, SSD, Faster R-CNN, etc.).
- Choix du modèle en fonction des besoins de l'application.



# — Phirio —

---

## Génération d'Images avec les GAN

---

Introduction aux réseaux génératifs adverses (GAN).  
Création d'un modèle GAN simple avec PyTorch.