



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



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, rosnodet et rosrund.
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



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



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

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
Tour d'horizon des outils:
pandas, pyarrow, agate, bokeh, scikit-learn, pybrain, tensorflow, keras, mxnet, caffe, Pytorch

Calculs et graphiques

NumPy : Base du calcul sur des tableaux
SciPy : Scientific Tools for Python, couche scientifique
Manipulation de tableaux, fonctions mathématiques.
Représentation graphique avec basemap et matplotlib.

Atelier : Mise en oeuvre de SciPy/NumPy : manipulation d'images, détection de contours

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
Comparaison et performances Pandas / pyarrow / NumPy

Atelier : construction d'ETL de base entre json et csvkaggl.com,

Comprendre les mécanismes d'interconnexion aux bases de données

Définitions : pilotes, connexions, curseurs, CRUD, transactions
Les pilotes : postgresql, mysql, mariadb, ... Présentation de sql-alchemy
Opérations : gestion du curseur, chargement de données, insertion et modification d'enregistrements

Atelier : mise en oeuvre avec postgresql. Construction d'ETL SQL/json

Comprendre les principaux outils de traitement et d'analyse de données pour Python

Présentation des outils d'apprentissage Python : scikit-learn, pybrain, TensorFlow/keras, mxnet, caffe



Atelier : mise en oeuvre de scikit-learn

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

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

Sites de références data-sciences

Ressources d'apprentissage, datasets, modèles de données pré-entraînés, etc ..
Présentation de : kaggle.com, data-puzzles.com, huggingface.co

Optimisation des développements

Tour d'horizon des outils actuels et futurs:
Jupyter notebook, Aide à la vérification de code, respect des recommandations PEP8 :
exemples avec pycodestyle, Pylint, Black
Analyse et production de code informatique avec une IA.
Génération de code avec OpenAI : démonstrations ChatGPT, apports, bonnes pratiques.

Atelier : utilisation de la génération de code et de snippets Python avec
ChatGPT

Analyse d'images avec TensorFlow et Keras

Historique du projet TensorFlow
Fonctionnalités
Architecture distribuée, plateformes supportées

Premiers pas avec TensorFlow

Installation de TensorFlow,
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

Optimisation des calculs

Calculs distribués : différents types de stratégies (synchrone ou asynchrone),
avec stockage centralisé des données ou dupliqué sur différents cpu
Distribution sur des GPUs
Utilisation de TPUs
Travaux pratiques sur une plateforme multi-GPU (RIG)



Phirio

Présentation des RN

Principe des réseaux de neurones
Descente de gradient
Multi-Layer Perceptron

Présentation de Keras

Conception d'un réseau de neurones
Différents types de couches: denses, convolutions, activations
Construction d'un modèle complexe
Prédictions et validation d'un modèle

Classification d'images avec Keras

Notion de classification, cas d'usage
Architectures des réseaux convolutifs, réseaux ImageNet
RCNN et SSD
Démonstrations sur les convolutions

Optimisation d'un modèle

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