

IHM ANNEXE 3

Formation développement d'IHM avec PyQt5 et communication Série en Python

Cette offre de formation permettrait aux participants d'apprendre à créer des interfaces graphiques interactives avec PyQt5 et maîtriser la communication série en Python pour interagir efficacement avec des systèmes embarqués.

Je souhaite aborder les aspects techniques suivants : environnement virtuel, communication série, traitement et affichage de données sur un graphique en temps réel tout en offrant une approche progressive et pratique.

Offre de Formation : Développer une Interface Homme-Machine avec PyQt5 et mettre en œuvre une communication série avec Python

Durée : 3 jours
Lieu : En présentiel, au CEMES, CNRS, Toulouse
Public cible : Électroniciens, ingénieurs et techniciens travaillant sur des systèmes embarqués, des capteurs ou des interfaces série pour la collecte et l'analyse de données.
Prérequis :

- Notions de base en Python, en programmation orientée objet et communication série.
- Amener son ordinateur portable sous Windows ou Linux requis.

Objectifs de la formation :

À la fin de la formation, les participants seront capables de :

1. **Installer et configurer un environnement virtuel Python** et gérer plusieurs versions de Python pour différents projets.
2. **Maîtriser l'utilisation de l'IDE PyCharm** pour développer, déboguer et gérer des projets Python de manière efficace.
3. **Utiliser les bibliothèques Python pyserial (ou serial)** pour établir une communication série avec un système externe, tel qu'un microcontrôleur ou un capteur.
4. **Afficher les données en temps réel** sur un graphique interactif avec **PyQt5** et **PyQtGraph**.
5. **Manipuler et traiter les données** à l'aide de **NumPy** pour effectuer des calculs et sauvegarder les résultats dans un fichier texte pour un traitement ultérieur.
6. **Utiliser des composants interactifs de PyQt5**, tels que **DockArea** et **ParameterTree**, pour ajuster dynamiquement les paramètres d'acquisition et les afficher en temps réel sur l'interface.
7. **Personnaliser et optimiser les interfaces graphiques** pour des applications scientifiques spécifiques.

Programme de la formation :

Demi-journée n°1 : Installation et configuration de l'environnement virtuel

- Pourquoi et comment utiliser des environnements virtuels Python ?
- Installation et gestion de plusieurs versions de Python avec **pyenv**.
- Création et gestion d'un environnement virtuel isolé pour un projet Python.
- Installation des outils nécessaires au développement : **PyCharm**, **pyserial**, **pyqt5**, **pyqtgraph**, **numpy**, etc.

Demi-journée n°2 : Gestion de la communication série

- Présentation de la bibliothèque **pyserial**.
- Développement d'une classe Python pour structurer la gestion de la communication série.
- Écriture, Lecture et envoi de données via la communication série.

- Mise en place d'une liaison série avec un système externe (e.g. un microcontrôleur ou un capteur).

Demi-journée n°3 : IHM interactives

- Utilisation du **DockArea** pour organiser de manière flexible les éléments de l'interface graphique.
- Introduction au **ParameterTree** pour créer des interfaces permettant de modifier dynamiquement les paramètres d'acquisition en temps réel.
- Exemples pratiques : modification et gestion des paramètres en temps réel.

Demi-journée n°4 : Affichage et traitement des données

- Introduction à **PyQt5** pour créer des interfaces graphiques en Python.
- Affichage des valeurs analogiques sous forme de graphique interactif avec **PyQtGraph**.
- Mise en œuvre d'un graphique dynamique pour visualiser les données en temps réel.
- Personnalisation de l'interface utilisateur : création et gestion des fenêtres, boutons, graphiques interactifs.
- Introduction à **NumPy** pour le traitement et la manipulation des données numériques.
- Opérations de base sur les tableaux NumPy.
- Sauvegarde des données traitées dans un fichier texte pour une analyse ultérieure.

Journée n°5 : Finaliser et Déployer son Application

- Création d'un fichier exécutable du programme afin de pouvoir le déployer sans dépendance à un environnement Python.
- Implémentation d'un mécanisme de logs et d'erreurs pour améliorer la fiabilité de l'application.
- Récapitulation des compétences acquises durant la formation.
- Discussion des applications possibles dans les projets de recherche, notamment pour la gestion des acquisitions, l'analyse des données et la création d'interfaces sur mesure pour les recherches scientifiques.
- Ouverture sur les possibilités d'extension des projets, telles que l'intégration d'autres types de capteurs, l'optimisation des processus de traitement de données.