



ID de Contribution: 26

Type: Non spécifié

Python en contrôle-commande et apports des LLM

mercredi 19 novembre 2025 16:00 (20 minutes)

Présentation : Python en contrôle-commande et apports des LLM

1. Introduction (2 min)

- Contexte : contrôle-commande → automatisation, supervision, communication avec des systèmes complexes
- Objectif : montrer l'usage de Python et l'apport des LLM dans le développement, en s'appuyant sur **des pratiques et outils mis en place dans mes projets**

Slide : Titre + image symbolisant contrôle-commande (automate ↔ PC)

2. Python en contrôle-commande (4 min)

- Points forts : simplicité, écosystème riche (PyEpics, PyQt, asyncio...), rapidité de prototypage
- Exemples d'utilisation :
 - scripts pour piloter des équipements
 - acquisition et traitement de données
 - interfaces homme-machine (IHM)
 - intégration avec systèmes existants
- Complément : **Python comble ses propres faiblesses** (performance, accès bas niveau) en s'appuyant sur d'autres langages, par ex. **C++ pour la communication hardware** ou bibliothèques optimisées en C

Slides :

- Schéma d'architecture (capteurs → Python → visualisation/commande)
- Exemple de code Python minimaliste (5-6 lignes lisibles)
- Encadré illustrant l'intégration Python ↔ C++ (wrapper, bindings)

3. LLM dans le développement (4 min)

- Pourquoi ? Gain de productivité, génération de code, documentation, conception d'architectures
- Cas d'usage concret :
 - génération rapide de scripts de test
 - assistance à la documentation utilisateur
 - proposition de solutions d'architecture
 - **validation / qualification automatisée de code** (tests unitaires, conformité aux standards)

- Limites : nécessite validation humaine, ne remplace pas l'expertise

Slides :

- Schéma : développeur ↔ LLM ↔ code Python
 - Exemple d'interaction : "propose des tests unitaires pour un module Python de communication"
 - Pipeline de tests automatisés avec une brique "LLM"
-

4. Approche de développement (4 min)

- Mise en place d'une **approche structurée** pour combiner Python, contrôle-commande et intelligence artificielle
- Objectifs :
 - clarifier l'architecture des projets
 - faciliter l'intégration des modules Python
 - expérimenter l'usage des LLM dans un cadre concret
- Illustration : exemple d'un mini-projet où Python pilote un équipement et où un LLM aide à valider/qualifier le code

Slides :

- Schéma d'architecture projet (couches/modules)
 - Exemple concret d'intégration Python + LLM dans un workflow de développement
-

5. Conclusion & ouverture (1 min)

- Bilan :
 - Python = outil clé en contrôle-commande
 - LLM = accélérateur et validateur
 - Approche structurée = amélioration de la qualité et de la productivité
- Perspectives :
 - renforcer la fiabilité
 - démocratiser l'usage des LLM dans l'industriel

Slide : 3 bullet points de conclusion + image symbolique (futur de l'automatisation)

Auteur: GEMOND, Thomas (CNRS-LPSC)

Orateur: GEMOND, Thomas (CNRS-LPSC)

Classification de Session: Bloc sessions unique

Classification de thématique: Langages et Ingénierie de Conception