



Phase : Conception préliminaire

Plan type pour le dossier de conception préliminaire

Un dossier de conception bien structuré permet de définir précisément les choix techniques et organisationnels pour le développement du système informatique.

1. Introduction

Faire une synthèse brève du dossier d'analyse et indiquer les objectifs du dossier de conception préliminaire.

2. Choix technologiques

- **Étude des besoins techniques** : gestion des données, communication, interface utilisateur, etc.
- **Évaluation des contraintes** : performances, compatibilité matérielle, portabilité, et coûts.
- **Comparaison des outils disponibles** : frameworks, bibliothèques spécifiques, protocoles.

Les décisions prises sont justifiées par leur pertinence vis-à-vis des objectifs et leur capacité à répondre aux exigences du système, tout en anticipant sa fiabilité, son évolutivité, et sa maintenabilité.

3. Représentation globale du système

3.1. Synoptique du système :

- Diagramme de déploiement UML réalisé avec **Modelio**, montrant les relations entre les sous-systèmes (capteurs, concentrateurs, interfaces utilisateur).
- Identification des flux principaux (données, énergie).

3.2. Architecture matérielle et logicielle :

- **Équipements matériels** : Description des capteurs, concentrateurs et actionneurs, avec un schéma des connexions (bloc ou câblage si nécessaire).
- **Logiciels associés** : Frameworks, bibliothèques, services.
- **Contraintes techniques** : Performances attendues, contraintes, fiabilité énergétique.
- Inclure un **diagramme bloc matériel** pour la vue d'ensemble physique du système.

4. Structure des données et protocoles d'échange

4.1. Organisation des données manipulées :

- **Modélisation des données :**
 - Modèle relationnel réalisé avec **DBSchema** pour les bases de données.
 - Exemple d'organisation de fichiers JSON, INI ou XML pour les données structurées non relationnelles.
- **Dictionnaire des données** : Tableau décrivant chaque champ de donnée (nom, type, contrainte).
- **Sécurité et intégrité** : Plan de sauvegarde, chiffrement et gestion des accès.

4.2. Définition des protocoles :

- **Protocole de communication** : Choix des technologies (MQTT, HTTP, WebSocket).
- **Spécifications des trames** : Exemples détaillés (JSON, binaire).
- **Gestion des erreurs** : Stratégies de reprise et validation des données.
- Utiliser un **diagramme d'activité UML** pour illustrer les échanges entre sous-systèmes via ces protocoles.

5. Adaptation des diagrammes de séquence et de classes

5.1. Diagrammes de séquence :

- **Intégration des choix technologiques** : Ajout des interactions concrètes entre les modules logiciels (ex. QTcpSocket, QSerialPort).
- **Illustration des échanges sous forme temporelle** : Diagrammes réalisés avec **Modelio** pour refléter les flux de données.

5.2. Diagramme de classes :

- **Mise à jour des classes** : Ajout des classes issues des frameworks et bibliothèques utilisés.
- **Relations entre les classes** : Associations, agrégations, héritages, avec les cardinalités pertinentes.
- **Validation de la cohérence** : Comparaison avec les cas d'utilisation pour garantir l'alignement.