

---

---

<b>Projet de fin de parcours</b>
----------------------------------

<i>Durée de réalisation : 4 jours / Travail individuel</i>
--

---

---

### Contexte

Au sein de ce projet de fin de parcours dédié à la formation en développement système sous Linux, l'accent est mis sur l'application pratique des compétences acquises tout au long du cursus.

Dans ce contexte, l'objectif est de mettre en pratique les connaissances en développement système, en programmation noyau et en gestion des périphériques, en créant un système fonctionnel basé sur une carte Raspberry Pi 3 (RPi). Cette carte sera intégrée à un réseau et interagira avec divers capteurs, actionneurs et périphériques, offrant ainsi aux participants l'occasion de démontrer leur maîtrise des concepts essentiels du développement système.

### Objectifs pédagogiques

- ☐ Développer des modules noyau
- ☐ Programmation système des applications parallèles et concurrentiels
- ☐ Communiquer les processus avec IPC
- ☐ Communiquer les processus sur le réseau

### Matériel

Ordinateur sous Linux – Carte Raspberry Pi 3 – Périphériques d'entrées/sorties

### Modalités d'évaluation

Les modalités d'évaluation pour ce projet incluent une soutenance technique d'une durée de 15 à 20 minutes, au cours de laquelle le candidat présentera sa réalisation. Cette présentation sera accompagnée d'un support visuel (présentation PowerPoint). L'objectif de cette présentation est d'expliquer en détail l'architecture de la solution mise en œuvre pour le projet, ainsi que les choix techniques effectués tout au long du processus.

Lors de la soutenance, le candidat devra démontrer sa compréhension approfondie du projet, en mettant en évidence les éléments clés de la conception, de l'implémentation et de la gestion des composants matériels et logiciels. Il devra également expliquer les raisons qui ont motivé ses choix techniques prises à chaque étape du développement.

## **Description**

Ce projet de fin de parcours vise à concevoir un système dédié à la surveillance et à la gestion d'une carte Raspberry Pi 3 (RPI) connectée à divers capteurs et actionneurs. Ce système sera accessible via une machine du réseau fonctionnant sous Linux.

La RPi sera équipée d'un écran LCD 16x2 connecté via le bus I2C, ainsi que d'une diode LED, d'un bouton poussoir et d'un capteur de température DHT11. L'interaction avec ces périphériques sera gérée par des modules noyau.

Les principales fonctionnalités du système comprennent :

- Affichage sur l'écran LCD d'un message d'accueil personnalisable par l'utilisateur, ainsi que de la température capturée par le capteur DHT11. Les données de température seront également enregistrées dans des fichiers journaux.
- Possibilité pour l'utilisateur d'activer ou de désactiver le clignotement de la diode LED via l'application ou en utilisant le bouton poussoir connecté à la RPi.
- Le clignotement de la diode LED est activé automatiquement lorsque la température atteint un seuil défini par l'utilisateur
- Acquisition de la température ambiante grâce au capteur de température.
- Mise en place d'un programme exécuté sur la Raspberry Pi pour assurer la communication entre l'application de supervision et la RPi.

L'application de contrôle et de supervision permettra de :

- Modifier la fréquence de capture des données de température.
- Consulter la dernière valeur de température capturée.
- Accéder à l'historique des valeurs de température.
- Personnaliser le message d'accueil affiché sur l'écran LCD.

**NB. : La mise en places de fonctionnalités supplémentaires serait fortement appréciée**

## **I-Préparation de la solution**

- ☐ Proposer les montages de la réalisation
- ☐ Proposer l'architecture de la solution
- ☐ Installations et configurations : Raspberry OS, chaine de compilation croisée, bibliothèques de développement, etc.

## **II- Programmation de modules noyau**

Développer les modules noyau permettant de contrôler les périphériques connectés à la carte Raspberry Pi

- ☐ Pilote pour mesurer la température
- ☐ Contrôler des périphériques d'entrée/sortie
- ☐ I<sup>2</sup>C : contrôler un écran LCD 16x2

## **III- Programmation applicative C sur la raspberry**

- ☐ Développement d'applications en C pour l'interaction avec les modules noyau.
- ☐ Mise en œuvre de la communication entre processus à l'aide de mécanismes IPC.
- ☐ Mise en place de mécanismes de communication entre les applications et les modules noyau.

## **IV- Communication et Gestion Distante**

- ☐ Intégration de fonctionnalités de communication réseau pour permettre la gestion à distance du système.
- ☐ Développement d'une application de contrôle à distance pour surveiller et configurer le système.

## **V- Intégration et tests**

- ☐ Intégration de tous les composants et modules développés en une solution fonctionnelle.
- ☐ Réalisation de tests approfondis pour assurer la réactivité des alertes et la stabilité du système.
- ☐ Vérification de la cohérence des données affichées sur l'écran LCD avec les mesures réelles des capteurs.
- ☐ Validation de la gestion à distance en s'assurant que les commandes sont correctement transmises et exécutées.
- ☐ Implémentation d'autres fonctionnalités utiles, telles que l'enregistrement des données dans des fichiers journaux.
- ☐ Optimisation du système pour minimiser l'utilisation des ressources et garantir une performance fluide