Rapport de GPE – Partie Numérique

**Livraison finale**

Casier sécurisé pour clé

**[GPE]\_NOM1\_Prenom1\_NOM2\_Prénom2.pdf**

Ce rapport doit être synthétique et s’adresse à un public averti quant aux termes techniques utilisés.

Dans cette partie nous allons nous intéresser au développement logiciel du système et les quelques modifications hardware effectuées. L’objectif est de réaliser une interface entre le système et l’utilisateur notamment à l’aide d’un écran tactile afin de renseigner un code PIN et d’un capteur éthylotest prouvant la sobriété pour déverrouiller un casier.

# Cahier des charges

Le casier doit pouvoir contenir des clés de voiture et les rendre facilement accessible à la main. L’objectif étant de réaliser un coffre sécurisé à destination d’un utilisateur lambda, l’interface doit être simple et intuitive. Que ce soit lors du renseignement/personnalisation du code PIN ou lors du test d’alcoolémie.

Le système contient donc quatre états du point de vu utilisateur, qui ont été repensé depuis le livrable intermédiaire :

- 1. déverrouillé (UNLOCKED)

- 2. création du PIN (SETPIN)

- 3. verrouillé (LOCKED)

- 4. renseignement du PIN (ENTERPIN)

L’objectif de ce projet est de réaliser un coffre sécurisé du côté software plus que physique , ainsi la solidité de la structure importe peu, seul le système électronique ne doit pas montrer de failles pour l’utilisateur.

# Manuel d'utilisation

Vous devez présenter ce manuel du point de vu utilisateur final (les outils de développement n’ont pas leur place ici !).

Vous devez décrire ici les modalités d'utilisation de votre réalisation de manière à ce qu'il puisse être mis en œuvre et essayé rapidement.

Pour un jeu d'échec par exemple, il faudra expliquer comment le joueur doit saisir les coordonnées de déplacement. Ou encore, si l'affichage n'est pas explicite, il faudra l'expliquer ici.

Au besoin, vous pouvez vous appuyer sur une représentation 3D ou une photo.

Précisez quelles sont les contraintes d’alimentation à respecter pour utiliser votre réalisation.

Votre manuel d’utilisation est complet si l’utilisateur qui ne connaissait pas votre produit n’a aucun doute sur ce qu’il peut faire et comment il peut le faire. N’hésitez pas à faire relire cette partie par un autre groupe !

# Description d'un algorithme du programme

Vous devez représenter ici au moins l’une des tâches de votre programme, en respectant le formalisme de représentation d’un algorithme. Elle peut se faire sous forme schématique (algorigramme, diagramme d'activité UML, …) ou sous forme textuelle (pseudo code, langage C, …). Une machine à états est la bienvenue.

# Structure du programme

Cette partie doit lister et décrire l'ensemble des fichiers du projet que vous avez produits, ainsi que leurs principales fonctions.

Un exemple est donné ci-après.

## Fichier « uart.c »

Ce module permet de manipuler l'UART pour envoyer et recevoir des caractères sur la liaison série RS232 de la carte.

|  |  |  |
| --- | --- | --- |
| **Fonction** | **Nom du développeur** | **Description** |
| UART\_testerReception | M. Dupond | Permet de savoir si un caractère a été reçu |
| UART\_recevoirOctet | M. Dupond | Permet d'obtenir un caractère (bloquant jusqu'à réception de caractère) |
| UART\_envoyerOctet | Mme. Dupont | Permet d'envoyer un caractère |

## Fichier « uart.c »

Ce module permet de manipuler l'UART pour envoyer et recevoir des caractères sur la liaison série RS232 de la carte.

|  |  |  |
| --- | --- | --- |
| **Fonction** | **Nom du développeur** | **Description** |
| UART\_testerReception | M. Dupond | Permet de savoir si un caractère a été reçu |
| UART\_recevoirOctet | M. Dupond | Permet d'obtenir un caractère (bloquant jusqu'à réception de caractère) |
| UART\_envoyerOctet | Mme. Dupont | Permet d'envoyer un caractère |

## Fichier « uart.c »

Ce module permet de manipuler l'UART pour envoyer et recevoir des caractères sur la liaison série RS232 de la carte.

|  |  |  |
| --- | --- | --- |
| **Fonction** | **Nom du développeur** | **Description** |
| UART\_testerReception | M. Dupond | Permet de savoir si un caractère a été reçu |
| UART\_recevoirOctet | M. Dupond | Permet d'obtenir un caractère (bloquant jusqu'à réception de caractère) |
| UART\_envoyerOctet | Mme. Dupont | Permet d'envoyer un caractère |

## Fichier « uart.c »

Ce module permet de manipuler l'UART pour envoyer et recevoir des caractères sur la liaison série RS232 de la carte.

|  |  |  |
| --- | --- | --- |
| **Fonction** | **Nom du développeur** | **Description** |
| UART\_testerReception | M. Dupond | Permet de savoir si un caractère a été reçu |
| UART\_recevoirOctet | M. Dupond | Permet d'obtenir un caractère (bloquant jusqu'à réception de caractère) |
| UART\_envoyerOctet | Mme. Dupont | Permet d'envoyer un caractère |

# Tests

Dans cette partie vous décrivez les différents tests que vous avez effectués avec le résultat du test.

Ce rapport de test permet de savoir à quel stade du développement se trouve votre réalisation.

|  |  |  |
| --- | --- | --- |
| **Intitulé du test** | **Description de ce qu’il faut faire pour jouer le test et de ce qu’on doit observer** | **Observation obtenue et conclusion** |
| Test d’alimentation du microcontrôleur : | Mesure au voltmètre ; les entrées d’alimentation doivent être à 5V | OK |
| Test de la liaison Tx -> PC | Test de la liaison électrique entre la broche utilisée pour une communication série en sortie de l’UART (TX) et la broche de réception de l’ordinateur | OK |
| Test d’envoi unidirectionnel entre le microcontrôleur et l’ordinateur d’un caractère | un caractère doit être envoyé sur la broche d’émission du microcontrôleur, il doit être reçu sur le PC et doit apparaître sur un terminal. | NON  Caractère non reçu |
| Test de la led « validation » | Lorsque l’utilisateur presse le bouton « validation », la led correspondante doit s’allumer pendant 500ms. Cette vérification peut être réalisée à l’oscilloscope. | KO  La led s’allume seulement pendant 400ms. |

# Cahier de suivi

Pour chaque date (chaque séance de travail, pendant ou en dehors des créneaux prévus à l’agenda), notez dans ce cahier de suivi les tâches réalisées, les réalisateurs, les difficultés rencontrées, l’état d’avancement de la réalisation…

L’objectif de cette démarche est de vous inciter à prendre du recul sur le déroulement de votre projet. Cette habitude est essentielle pour l’ingénieur qui doit savoir quantifier son temps de travail et apprendre à anticiper les durées des tâches.

Reprenez et complétez le cahier de suivi du livrable intermédiaire.

|  |  |  |
| --- | --- | --- |
| Date | Tâches, réalisateurs, difficultés rencontrés. | A faire la prochaine fois |
| 03/01 | Lucky Luke : routage de la carte Saloon 2.0  Jolly Jumper : développement du module logiciel Abreuvoir.c/h.  Difficulté rencontrée : le bug dans le fichier Dalton.c nous a fait perdre du temps… Nous l’avons résolu en lisant la documentation (plus vite que notre ombre). | Lucky Luke : DRC, validation du routage, envoi en fabrication.  Jolly Jumper : Test du module logiciel Abreuvoir.c/h |
|  |  |  |

# État d'avancement et analyse du projet réalisé

Dans cette partie, vous rendez compte de l'avancement de votre projet.

Le cas échéant, nous vous demandons de lister les points à terminer (telle partie à écrire, telle fonctionnalité à ajouter, tel sous-programme à tester, …).

Vous devez ANALYSER ce que vous avez produit :

Exemples :

* Si vous deviez refaire le projet, que feriez-vous autrement ?
* Quel moyen de débogage aurait-il fallu utiliser plus tôt ou différemment ?
* Quelles limites gênantes et pour quels composants ?

# Conclusion

Cette partie est disponible pour une brève conclusion.

Cette partie peut aussi servir à mettre en valeur un apprentissage, par exemple une difficulté que vous avez surmontée.