Le but de ce projet SAE est de créer un produit/instrument qui doit servir de démonstrateur technique pour les journées portes ouvertes de l’IUT mais il doit aussi pouvoir être assez réaliste pour pouvoir être utiliser par un vrai instrumentiste. Notre instrument va imiter un clavier maitre avec certaines fonctionnalités qui sont clonées/déportées sur une application PC ou Android.

Analyse descriptive/pourquoi

Qui : Intervenant journée portes ouverte (Car l’option ESE manque de démonstrateur « fun »)

Quoi : Démonstration, entrainement, concert, amusement. (Sert à montrer que l’option est intéressante (Création de PCB, IOT, info embarquée)

Où : Hall IUT Neuville, salle d’entrainement, concert (N’importe où, environnement Sec) (Car n’est pas imperméable)

Quand : N’importe Quand

Comment : En entrainement ou en par un instrumentiste ou en démonstration par les Intervenant de l’IUT.

Voici les différents interacteurs :

* Application
* Utilisateurs (instrumentiste et intervenants journée porte ouverte)
* Environnement extérieur
* Environnement d’utilisation
* Normes et règlementation
* Source(s) d’énergie(s) (Batterie prise secteur)
* Chef d’orchestre (MIDI)

|  |  |  |
| --- | --- | --- |
| Fonction de services | Fonctions Techniques | Solutions technique |
| FP1 : Générer du son lors d’un appui sur clavier | FT1 : Détecter les touches du clavier | ST1 : µcontrôleur |
|  | FT1.1 : Générer un son différent pour chaque touche | ST1.1 : µcontrôleur |
|  | FT1.2 : Gestion du son intégré | ST1. 2 : Utilisation amplificateur SAE S3 |
|  | FT1.3 : Changer les sonorités | ST2 : Bouton sur clavier et sur Application |
|  | FT1.4 : Enregistrement des sons joués |  |
|  | FT1.5 : Changer l’intensité du son de sortie |  |
|  | FT1.6 : Synchronisation sur chef d’orchestre |  |
| FS1 Application |  |  |
|  |  |  |
|  |  |  |
| FS2 : Source d’énergie | FT2 Visualisation niveau de batterie | ST 2 : Témoin lumineux |
|  | FT2.1 Visualisation de charge | ST2.1 : Témoins lumineux |
|  | FT2.2 Fonctionnement sur batterie et secteur | ST2.2 Ajout batterie |
|  | FT2.3 : Respecter les normes | ST2.2.1 Ajout système auto de charge décharge batterie |

Voici les différentes fonctionnalités.

Fonctionnalités

* Clavier 2 octaves (25touches)
* Différentes sonorités : différentes formes du signal
* Enregistrement des sons jouées
* Identification des notes jouées
* Système intégré pour la gestion du son (ou prise jack si pas possible)
* Taille standard
* Identifier un son (nom de la note correspondante)

Solutions techniques

* Utilisation ESP32 pour contrôler le système
* Témoin lumineux pour identifier la touche jouée
* Utilisation d’un bouton ou application pour contrôler la sonorité (forme du signal)
* Utiliser un potentiomètre pour contrôler le son
* Utilisation ampli SAE S3 pour Haut-parleur
* Potentiomètre pour curseur de son

Fonctionnalités Application

* Utilisation clavier
* Contrôle de la sonorité
* Curseur de son
* Animation Noepixel en dehors du mode identification des touches
* Reconnaissance son (nom de la note)

Contraintes

* Budget (voir avec prof)

A acheter

* ESP32-S3 DEVKIT : [ESP32-S3-DevKitC-1 Carte de développement APKLVSR WROOM-1-N16R8 ESP32-S3-DevKitC-1 avec WiFi, Bluetooth 5.0 compatible avec Arduino : Amazon.fr: Informatique](https://www.amazon.fr/ESP32-S3-DevKitC-1-d%C3%A9veloppement-APKLVSR-WROOM-1-N16R8-compatible/dp/B0CQNB74R9/ref=asc_df_B0CQNB74R9?tag=bingshoppin0f-21&linkCode=df0&hvadid=80539415941440&hvnetw=o&hvqmt=e&hvbmt=be&hvdev=c&hvlocint=&hvlocphy=&hvtargid=pla-4584138886240346&th=1)
* Contact touche clavier : à chercher ou fabriquer
* Bande de Led Noepixel