ECOLE SUPERIEURE
D'INGENIEURS DE RENNES

Université de Rennes

Réalisation d’un Boîtier de Commande (Stream deck) connecté en Bluetooth

BROU Justin
EL NAGGAR Sofiane
HANY Matéo
KOBON Charles

2023/2024

SOMMAIRE

I. Présentation du Projet

A. Concept et Inspiration

Avec l’explosion des nouvelles technologies, on ressent de plus en plus le besoin de mieux interagir avec nos appareils électroniques. Même si on a toujours utilisé des claviers et des souris pour ça, aujourd’hui, on cherche des moyens plus simples et adaptés à nos besoins. C’est dans cette optique qu’est apparue l’idée du Stream Deck, conçu au départ pour les personnes qui font du streaming, mais qui peut servir pour bien d’autres choses.

Notre inspiration principale provient des défis quotidiens que rencontrent les utilisateurs d'ordinateurs et de dispositifs mobiles pour effectuer des tâches courantes rapidement. Avec un Raspberry pi pico W 2022 et six (6) boutons poussoirs, l'idée est de créer un boitier externe personnalisable qui interagit sans fil avec un ordinateur ou un autre dispositif, permettant à l'utilisateur de déclencher des macros prédéfinies, comme augmenter le son ou lancer une application.

B. Description Technique

Au cœur de notre dispositif se trouve le Raspberry pi pico W 2022, un microcontrôleur avec des capacités Bluetooth intégrées. C’est cette puce qui gère l’entrée des six (6) boutons poussoirs et envoie les données correspondantes via Bluetooth à l'application cible.

L’interface utilisateur de ce boîtier est gérée par une application web. Le front-end de cette application est conçu en HTML et CSS, offrant une interface visuellement attrayante et facile à utiliser. Le back-end, quant à lui, est développé en python, JavaScript et Node.js. Il est séparé en deux parties, un serveur et un client. Son rôle principal est de récupérer et d'interpréter les données envoyées par le Raspberry Pi, de gérer les macros associées à chaque bouton et de communiquer avec les applications installées sur l’ordinateur de l'utilisateur.

La communication entre le boîtier et l’application se fait via BLE (Bluetooth Low Energy), une technologie connue pour sa faible consommation d'énergie et sa fiabilité dans la transmission de données.

C. Portée et objectifs

La vision de ce projet est de créer une interface utilisateur externe, à la fois portable et personnalisable. Les objectifs spécifiques sont :

- Concevoir un boîtier fonctionnel et esthétiquement agréable intégrant le Raspberry pi pico W 2022 et les 6 boutons poussoirs.
- Développer une application web intuitive pour la gestion des macros et la communication avec le boîtier.
- Assurer une communication stable et efficace via BLE entre le boîtier et l'application.
- Permettre à l'utilisateur de personnaliser facilement les actions associées à chaque bouton.

À travers ce projet, nous aspirons à offrir une solution innovante qui enrichit l'expérience utilisateur en matière d'interactions avec leurs dispositifs numériques.

II. Cahier des Charges

A. Exigences Fonctionnelles

- **Interface de l'application :** Une interface utilisateur, web, claire et intuitive, permettant de paramétrer et de contrôler simplement les boutons du Stream Deck.
- **Configuration des Macros :** La possibilité pour l'utilisateur de choisir et de paramétrer une macro spécifique pour chaque bouton poussoir.
- **Gestion des Applications :** L'application devra être capable de détecter et de lister les applications installées sur l'ordinateur. De cette manière, l'utilisateur pourra définir des actions liées à ces applications, comme les ouvrir directement via le Stream Deck.
- **Fonctionnalités Spécifiques du Stream Deck :**
    - ❖ Gérer la luminosité de l'ordinateur (uniquement sur les ordinateurs portables).
    - ❖ Gérer le volume sonore de l'ordinateur.
    - ❖ Faire une capture d'écran.
    - ❖ Ouvrir une page web spécifique.

- ❖ Ouvrir une application précédemment sélectionnée.
- ❖ Ouvrir un fichier donné.

B. Exigences Techniques

- **Technologies Web** : Utilisation de HTML et CSS pour le développement du front-end de l'application.
- **Programmation Back-End** : Le back-end de l'application sera développé en utilisant Python, JavaScript et Node.js.
- **Communication avec Raspberry Pi** : L'application récupèrera en temps réel les valeurs envoyées par le Raspberry Pi via la connexion Bluetooth.
- **Connexion sans Fil** : Mise en place d'une connexion fiable par BLE (Bluetooth Low Energy) entre le boîtier Stream Deck et l'application.

C. Contraintes et Limitations

- **Portée de la connexion :** La portée effective de la connexion Bluetooth peut varier en fonction de l'environnement et des obstacles présents.
- **Compatibilité :** La gestion de la luminosité sera disponible seulement pour les ordinateurs portables.
- **Navigateur :** Certaines fonctions utilisées dans le front-end sont dépendantes du navigateur.
- **Alimentation :** Le boîtier Stream Deck nécessitera une source d'alimentation adéquate, que ce soit via batterie ou par connexion filaire.
- **Limitations matérielles :** La performance et la réactivité pourraient être limitées par les capacités matérielles du Raspberry Pi pico W 2022.

III. Choix et Justification Technologiques

A. Langages et Frameworks

- **HTML/CSS** : Le choix d'utiliser HTML et CSS pour le développement front-end s'est imposé naturellement, car ces langages sont des standards universels pour le développement web. Ils offrent la flexibilité et la compatibilité nécessaires pour créer une interface utilisateur intuitive et réactive.
- **Node.js :** Le choix de Node.js pour le back-end a été motivé par plusieurs raisons. D'une part, il s'agit d'un environnement d'exécution JavaScript côté serveur qui est performant et adapté aux applications en temps réel. De plus, c'était une occasion pour l'équipe de travailler sur cette technologie moderne et de développer de nouvelles compétences.

B. Communication sans Fil

- **Bluetooth Low Energy (BLE)** : BLE a été choisi comme protocole de communication entre le boîtier Stream Deck et l'application. Il offre une consommation d'énergie faible, ce qui est idéal pour des dispositifs portables, et une portée suffisante pour une utilisation domestique ou de bureau.

C. Plateforme Matérielle : Raspberry Pi

- **Raspberry Pi Pico W 2022** : Cette plateforme a été choisie pour sa flexibilité, sa communauté active et son coût relativement bas. Pour la programmation de cette carte, nous avons opté pour le langage Python en utilisant l'IDE Thonny, qui est spécialement conçu pour la programmation sur les cartes Raspberry Pi.

IV. Organisation du Projet

A. Répartition du Travail

Pour garantir une mise en œuvre efficace de ce projet, nous avons élaboré une répartition des différentes tâches de travail entre les différentes membres du groupe. On s'est basé sur trois axes principaux : le Front-End, le Back-End et la programmation du Raspberry pi.

a. Front-End

Le front-end a été réparti sur plusieurs personnes à travers le projet. Pour commencer, Charles s'est occupé du prototypage du front avec l'idée globale de l'interface et de son développement. Par la suite, Sofiane a récupéré la gestion du front avec l'implémentation d'une interface plus esthétique et donc un travail plus poussé sur le CSS. Charles l'a enfin amélioré pour ajouter des menus pour sélectionner les différentes applications.

b. Back-End

Le back-end a été séparé en de nombreuses parties basées sur le niveau de complexité estimé pour chacune d'elle. Nous avons mis d'un côté la récupération des applications et l'ouverture de ces dernières. De l'autre, toutes les autres fonctionnalités telles que la gestion du volume, de la luminosité etc.

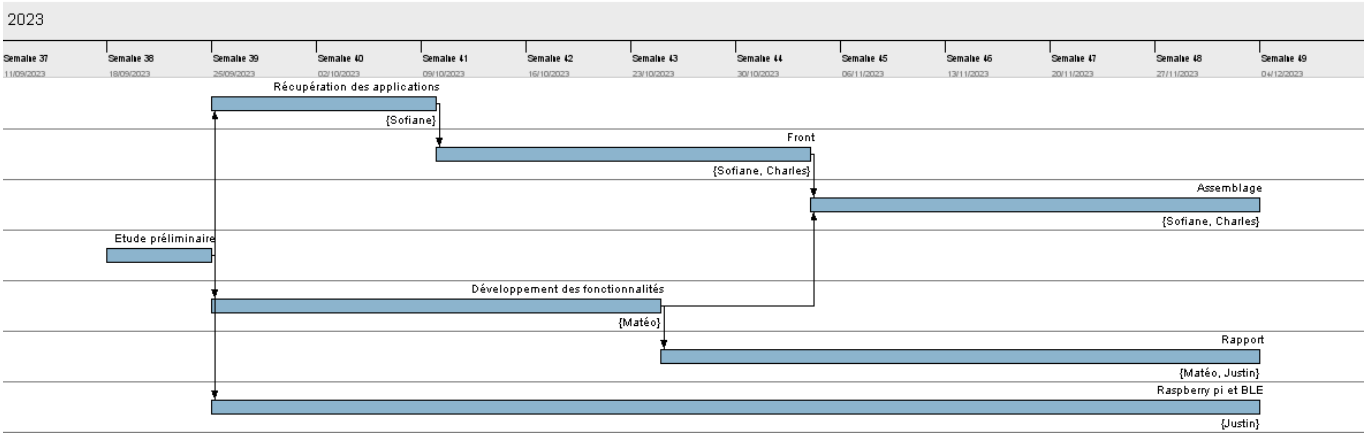
Sofiane et Charles se sont donc occupé de la gestion des applications et Matéo de tout le reste. La gestion des applications s'est avérée plus simple que prévu et a permis à Sofiane et Charles de se concentrer sur le front.

c. Raspberry pi

Justin est responsable de la partie matérielle avec le Raspberry Pi. Il va programmer le Raspberry Pi pour qu'il puisse se connecter et communiquer via Bluetooth. Il écrira aussi le code pour que le Raspberry Pi reconnaisse quand les boutons sont pressés. Après avoir connecté les boutons au Raspberry Pi, il vérifiera que tout fonctionne comme prévu. Il réalisera des tests pour s'assurer que les boutons répondent correctement aux commandes. Justin travaillera avec le reste de l'équipe pour que le Raspberry Pi interagisse bien avec l'application. Enfin, il rédigera des instructions sur tout ce qu'il a fait pour que tout le monde dans l'équipe puisse comprendre et continuer à travailler avec le Raspberry Pi.

d. Assemblage

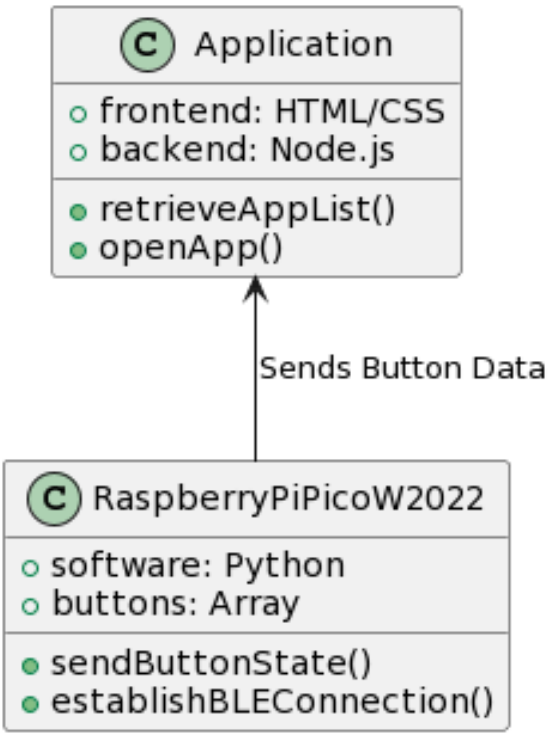
L'assemblage, c'est-à-dire la fusion entre le front et l'ensemble des fonctionnalités du back a été fait premièrement par Sofiane qui a tenté plusieurs méthodes qui se sont avérées infructueuses. Charles de son côté à mis en place le serveur, indispensable à l'interaction entre le client, la page web et le Raspberry. Pour finir, Charles a apporté un prototype d'assemblage fonctionnel.



B. Diagramme de Gantt

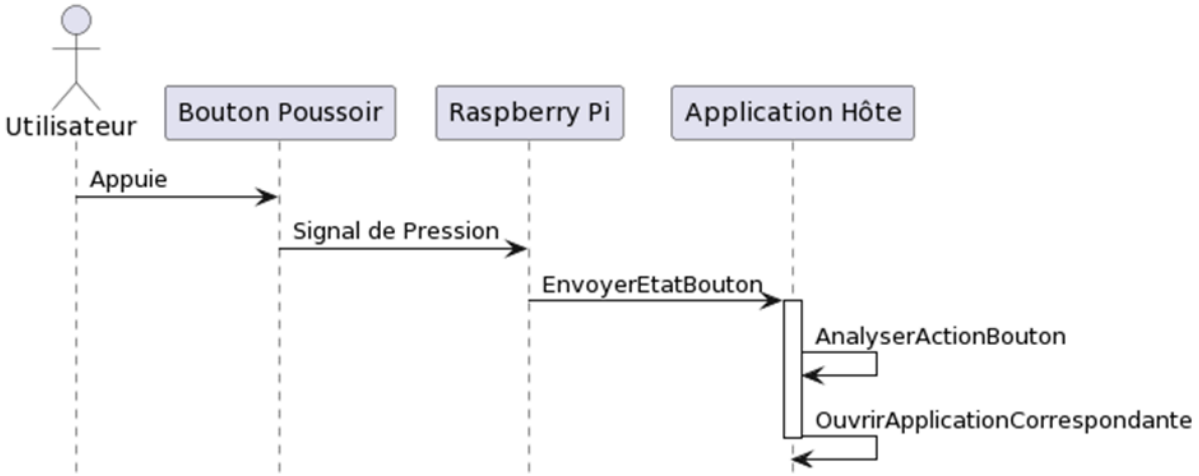
C. Utilisation d'UML pour la Conception

- **Diagramme de Classes** : Utilisé pour représenter la structure et la relation entre les classes du système.



- **Diagramme de Séquence** : Utilisé pour illustrer comment les objets interagissent dans une séquence temporelle spécifique.

➢ **Pour l'action "Ouvrir une application”**

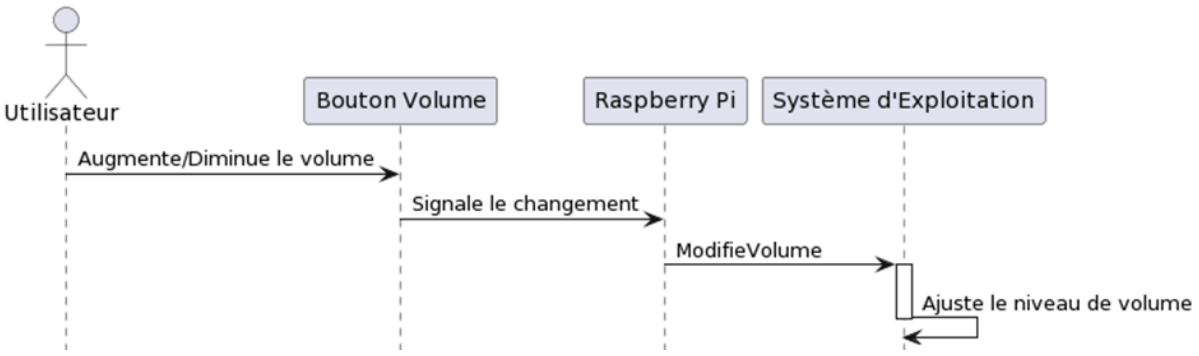


Dans ce diagramme :

- **Utilisateur** : La personne qui interagit avec le bouton poussoir.
- **Bouton Poussoir** : Le dispositif physique que l'utilisateur presse.
- **Raspberry Pi** : L'appareil qui reçoit le signal du bouton poussoir.
- **Application Hôte** : Le logiciel sur l’ordinateur qui réagit à l'entrée du bouton.

Lorsque l'Utilisateur appuie sur le Bouton Poussoir, le Raspberry Pi détecte cette pression et envoie l'état du bouton à l'Application hôte. L'application hôte analyse alors cette information et effectue l'action correspondante, qui est d'ouvrir une application spécifique.

➢ **pour Gérer le Son**



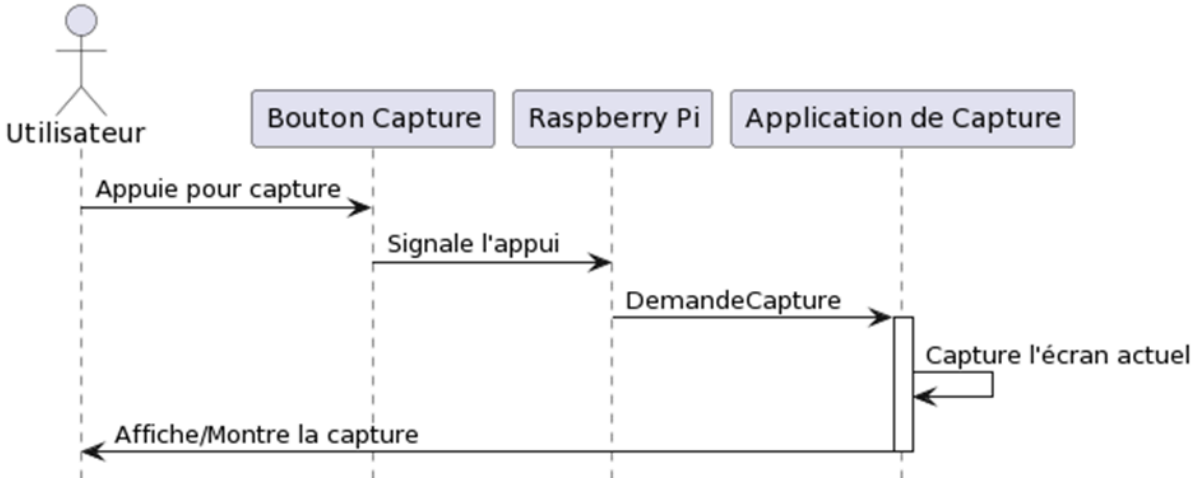
Dans ce diagramme :

- **Utilisateur** : La personne qui souhaite ajuster le volume.
- **Bouton Volume** : Le bouton spécifique assigné pour augmenter ou diminuer le volume.
- **Raspberry Pi** : L'appareil qui reçoit le signal de changement de volume.

- **Système d'Exploitation** : Le logiciel qui contrôle le niveau de volume de l'ordinateur.

L'utilisateur interagit avec le bouton Volume pour augmenter ou diminuer le son. Le Raspberry Pi capte ce signal et informe le Système d'Exploitation du changement souhaité. Le Système d'Exploitation ajuste ensuite le volume selon la demande.

➢ **Pour Faire une Capture d'Écran**

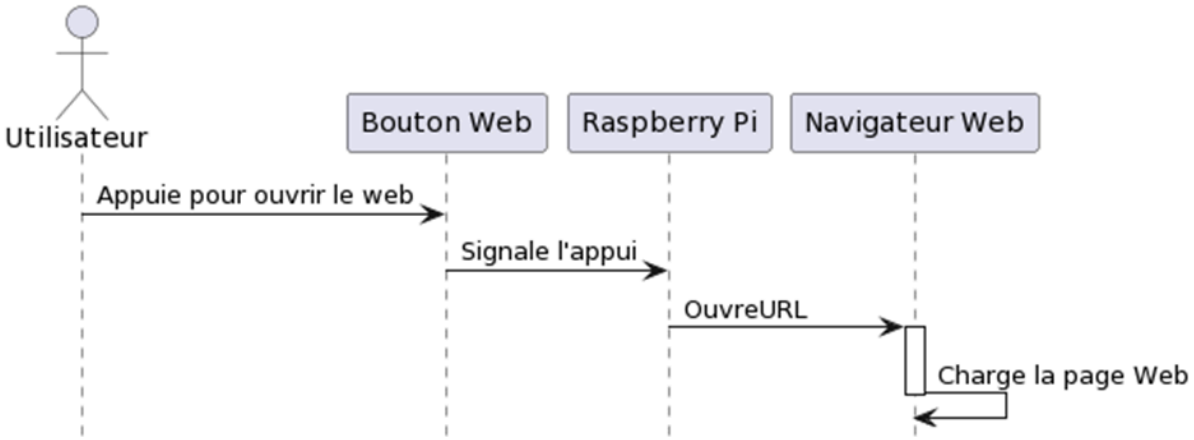


Dans ce diagramme :

- **Utilisateur** : La personne qui souhaite capturer l'écran actuel.
- **Bouton Capture** : Le bouton dédié à la fonction de capture d'écran.
- **Raspberry Pi** : L'appareil qui détecte l'appui du bouton et envoie la demande de capture.
- **Application de Capture** : Le logiciel qui effectue la capture d'écran.

Quand l'Utilisateur presse le Bouton Capture, le Raspberry Pi envoie la demande à l'Application de Capture qui réalise la capture d'écran. L'image capturée est ensuite affichée ou montrée à l'Utilisateur par l'Application de Capture.

➢ **Pour Ouvrir une Page Web**

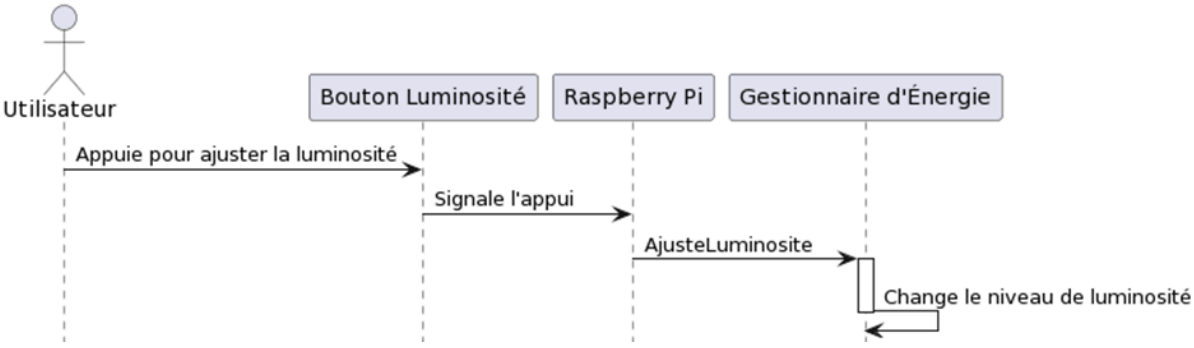


Dans ce diagramme :

- **Utilisateur** : La personne qui veut ouvrir une page Web spécifique.
- **Bouton Web** : Le bouton qui, une fois activé, déclenche l'ouverture d'une page Web.
- **Raspberry Pi** : L'appareil qui transmet la demande d'ouverture de page Web.
- **Navigateur Web** : L'application utilisée pour naviguer sur Internet.

L'utilisateur appuie sur le Bouton Web, le Raspberry Pi reconnaît cette action et communique avec le navigateur Web pour ouvrir l'URL désirée. Le navigateur Web charge et affiche la page demandée.

➢ **Pour Gérer la Luminosité (sur un ordinateur portable)**



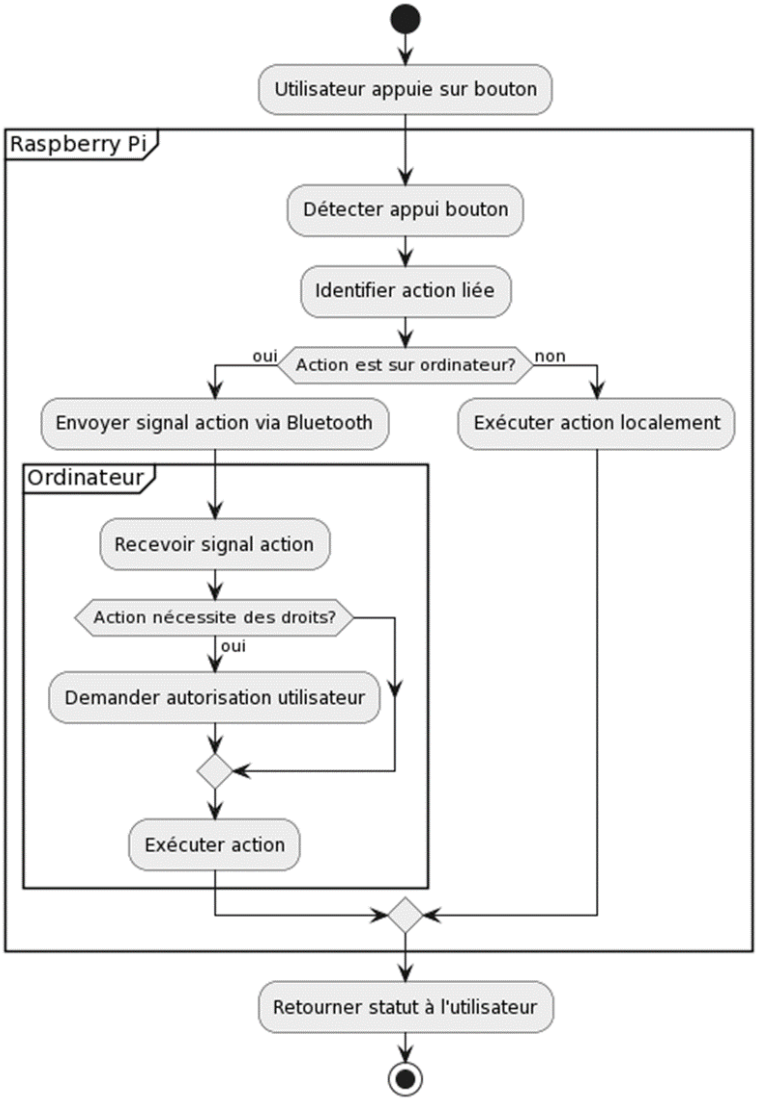
Dans ce diagramme :

- **Utilisateur :** La personne qui souhaite ajuster la luminosité de son écran d'ordinateur portable.
- **Bouton Luminosité** : Le bouton utilisé pour augmenter ou diminuer la luminosité de l'écran.
- **Raspberry Pi** : L'appareil qui reçoit le signal du bouton et qui envoie la commande pour ajuster la luminosité.

- **Gestionnaire d'Énergie** : Le composant du système qui gère les paramètres d'énergie, y compris la luminosité de l'écran.

Lorsque l'Utilisateur appuie sur le Bouton Luminosité, le Raspberry Pi envoie une commande au Gestionnaire d'Énergie pour changer le niveau de luminosité de l'écran de l'ordinateur portable selon la demande de l'utilisateur.

- **Diagramme d'Activité :** Représente le flux de contrôle ou le flux d'objet entre les activités et soutient la vue dynamique du système.



Dans ce diagramme d'activité UML :

- L'**Utilisateur** déclenche le processus en appuyant sur un bouton du Stream Deck.
- Le **Raspberry Pi** détecte l'appui sur le bouton et identifie l'action qui y est associée.
- S'il s'agit d'une action qui doit être effectuée sur l'ordinateur, le Raspberry Pi envoie un signal correspondant via Bluetooth.
- L'**Ordinateur** reçoit ce signal et, si l'action requiert des droits supplémentaires, demande une autorisation à l'utilisateur.
- L'ordinateur exécute ensuite l'action demandée.
- Un retour est donné à l'utilisateur, indiquant que l'action a été effectuée ou non.