

Sujet de projet Application System

Tanguy ROUDAUT — Melvin DUBEE

FIPASE 24

27 septembre 2023

1 Contexte

abcd

2 Technologies utilisées (idée)

L'idée serait de mesurer les performances d'un RaspberryPi contenant une interface graphique (*Wasm ou non*), une commande vocale et le contrôle de plusieurs capteurs le tout codé en Rust.

1. **Carte** : RaspberryPi

2. **Language** : Rust

3. **Plugin reconnaissance vocale** : [Picovoice](#)

Picovoice est une solution de reconnaissance vocale hors ligne (*contrôle total sur les données vocales et la confidentialité de l'utilisateur*) en Rust qui permet aux développeurs d'intégrer des fonctionnalités vocales puissantes et privées dans leurs projets sur Raspberry Pi, améliorant ainsi l'interaction homme-machine de manière locale et efficace.

4. **Crate micro bluetooth** : [BlueR](#)

Picovoice est utilisé pour lire des fichiers audio et vérifier si des données sont intéressantes. Pour enregistrer de l'audio dans un fichier on pourrait utiliser un biding *BlueR* de la stack bluetooth linux officiel *BlueZ*. On utilisera donc un micro bluetooth et un profil audio adapté pour capter l'audio puis l'enregistrer.

5. **Plugin GUI** : [Tauri](#) ou [Yew](#)

— Tauri est une bibliothèque polyvalente en Rust qui permet de créer des applications de bureau (*intégrant une webview en localhost*) multiplateformes sur Raspberry Pi en utilisant des technologies web, offrant ainsi une manière efficace et flexible de développer des logiciels interactifs et intégrés localement.

— Yew est une bibliothèque Rust pour le développement d'applications web interactives sur Raspberry Pi, offrant une combinaison de performances élevées, de sécurité et de flexibilité pour la création d'interfaces utilisateur modernes et réactives. Les UI peuvent interagir avec des capteurs, des dispositifs GPIO et d'autres fonctionnalités matérielles de manière fluide. De plus, cela permettrait d'exploiter le WebAssembly en Rust sur RPI

6. **Crate GPIO** : [RPPAL](#)

rppal est une bibliothèque Rust pour le développement d'applications qui interagissent avec les fonctionnalités matérielles du Raspberry Pi, offrant ainsi un moyen puissant et fiable de contrôler et de surveiller les dispositifs connectés à votre Raspberry Pi en utilisant le langage Rust.

3 Achat estimé

Composant	Ref	Site	Prix
moteur	-	-	0€
capteur de vent	-	-	0€
-	-	-	0€
-	-	-	0€
Total			0€

4 Section une

4.1 Sous section

4.1.1 Sous sous section

Spontus kas honnont kae hent fraoñval giz keniterv, boutailh holl prad tad-kaer koan piv bez armel kenderv dreist-holl, du da degouezhout dor lezenn gwinegr ler treut du tregont tre kontañ, dorn ar gwellañ mor c'hoar-gaer, c'hoant nadoz va Doue. Maneg goz goz moan oferenn. Kêr diouzh bag-dre-dan hor kambr kastell kêr stur bevañ. Eeun bandenn pevar-ugent. Mat nav gourc'hemenn ¹.

1. une liste numéroté
2. *Un Texte Italic*
3. **Un Texte Gras**
4. UN TEXT EN CAPITAL

- une liste pucé
- *Un Texte Italic*
- **Un Texte Gras**
- UN TEXT EN CAPITAL

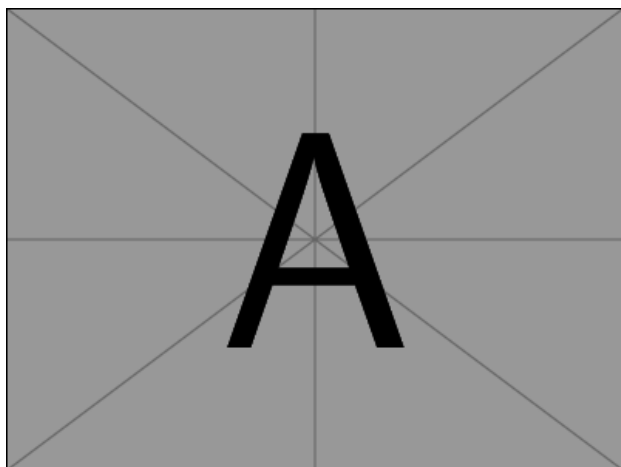


FIGURE 1 – My figure A

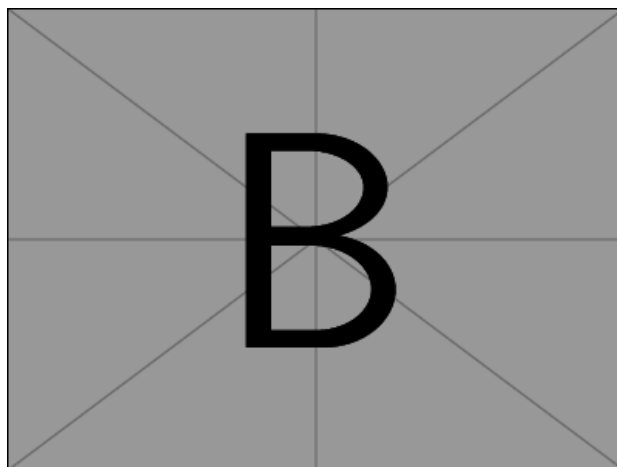


FIGURE 2 – My figure B

Une case	Une autre		
	Une autre	Une autre	Une autre
Une autre	Une autre	Une autre	Une autre
Une autre	Une autre	Une autre	Une autre
Une autre	Une autre	Une autre	Une autre
Une autre	Une autre	Une autre	Une autre

1. My footnote → Description of my footnote