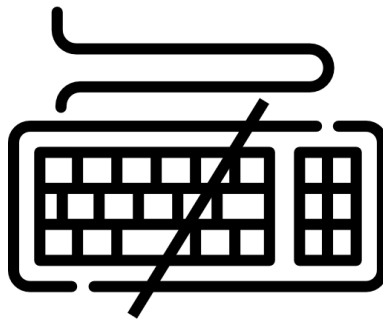


MyLab2 Split Keyboard

Rapport de projet

Micro-contrôleurs et périphérique



Michael Divià (N° 22649552)

jeudi 20 juin 2024

Informatique et Systèmes de Communication

Table des matières

1	Rappel du cahier des charges	3
1.1	Descriptif détaillé du projet	3
1.2	Schéma d'illustration	3
1.3	Sources d'interruptions	3
1.4	Objectifs impératifs	3
2	Travail effectué	4
2.1	Objectifs impératifs	4
2.2	Illustration de l'affichage	4
2.3	Vidéo de démonstration	4
2.4	Éléments importants	5
2.4.1	Dalle tactile	5
2.4.2	Carte Slave	5
2.4.3	Communication CAN	5
2.4.4	Carte Master	5
2.4.5	Communication UART	5
2.4.6	Script Python	6

Table des figures

1	Branchements et communications	3
2	Disposition du clavier	4
3	Claviers MyLab2	4
4	trame UART	5

1 Rappel du cahier des charges

1.1 Descriptif détaillé du projet

Le but est de créer un split keyboard à l'aide de 2 cartes MyLab2. Chaque carte affichera la moitié d'un clavier sur son écran tactile. Les 2 cartes communiqueront entre elles via le bus CAN. La carte Slave transmettra les inputs de l'utilisateur via CAN à la carte Master. La carte Master sera aussi connectée à un PC, via UART, qui convertira à l'aide d'un script Python les inputs reçus de la part des cartes en input clavier.

1.2 Schéma d'illustration

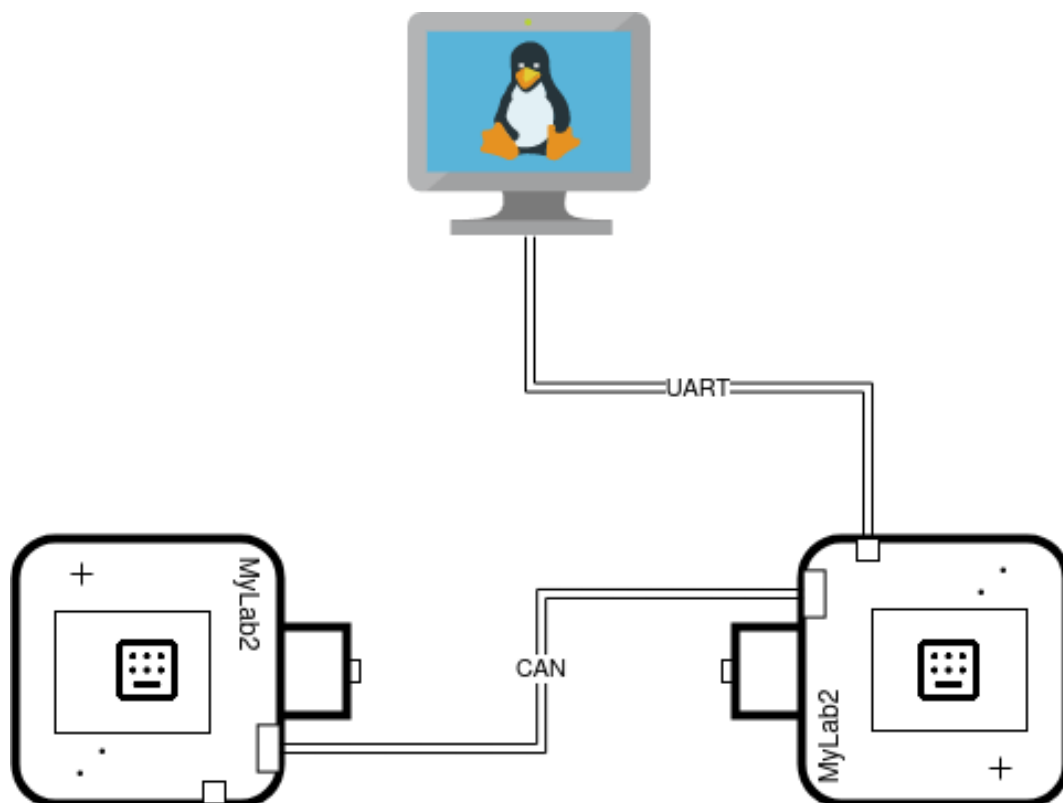


FIGURE 1 – Branchements et communications

1.3 Sources d'interruptions

- CAN
- Écran tactile

1.4 Objectifs impératifs

- Capture des inputs de l'écran tactile
- Transmission des données via CAN
- Transmission des données via UART
- Script Python d'interprétation des trames UART

2 Travail effectué

2.1 Objectifs impératifs

Tout les objectifs impératifs ont été accomplis et sont fonctionnels.

2.2 Illustration de l'affichage

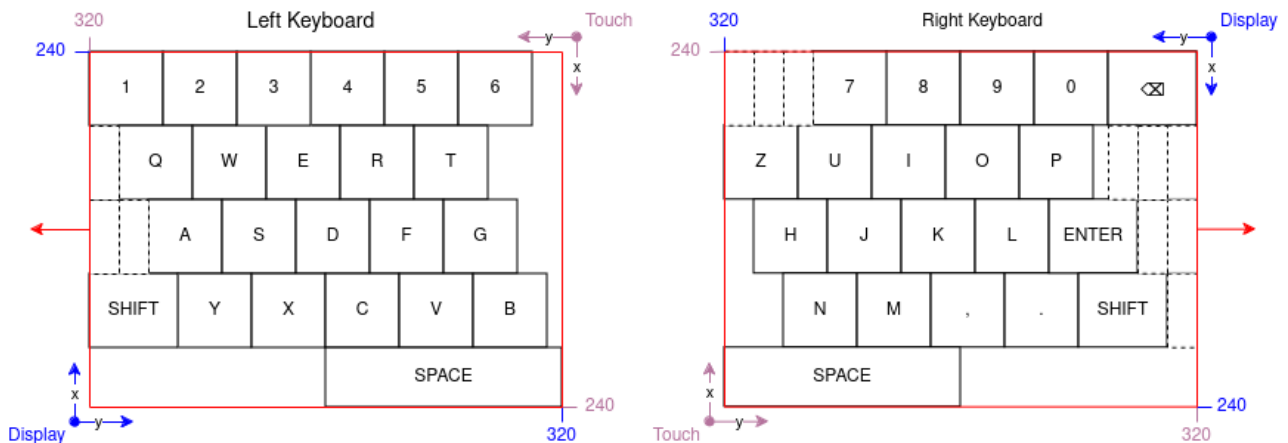


FIGURE 2 – Disposition du clavier

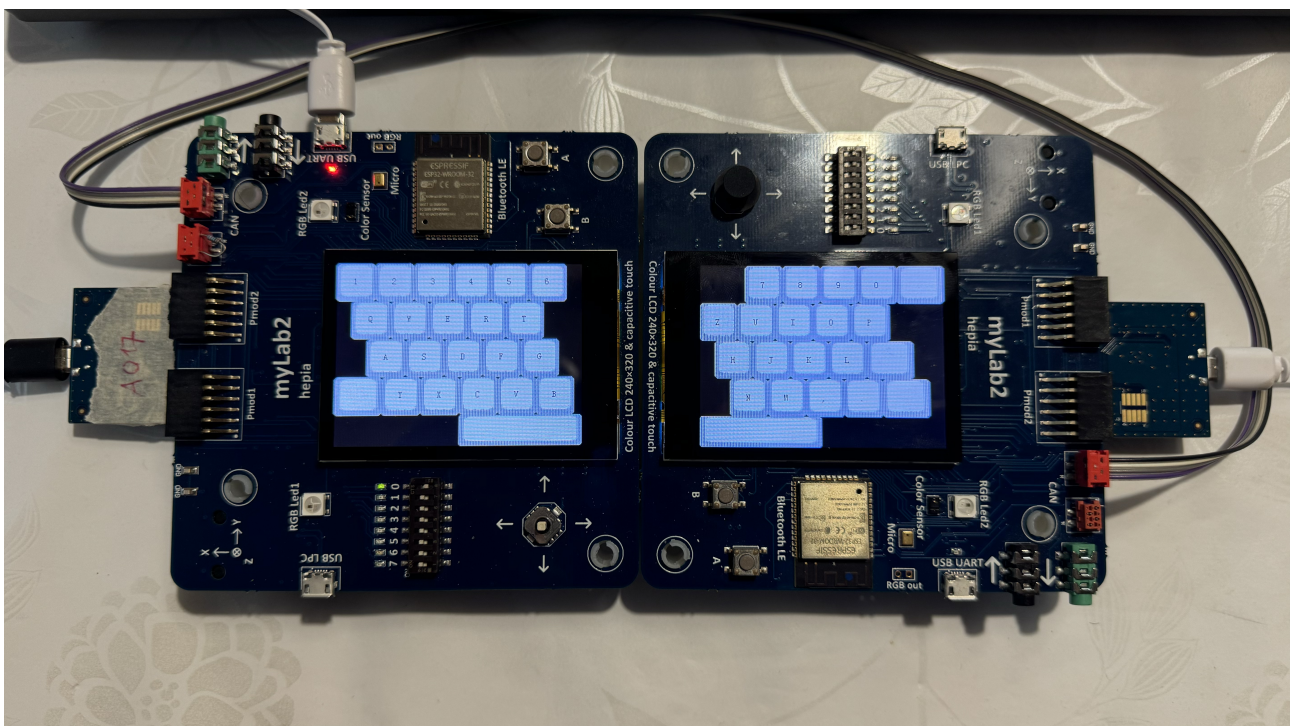


FIGURE 3 – Claviers MyLab2

2.3 Vidéo de démonstration

Une vidéo de démonstration peut être téléchargée via [ce lien](#).

2.4 Éléments importants

2.4.1 Dalle tactile

Nous communiquons via I2C avec la dalle tactile. Cette dernière crée une interruption et set un flag à 1 quand nous la touchons et à 0 quand nous la relâchons.

Nous pouvons ensuite aller récupérer (via I2C) les coordonnées X et Y de l'emplacement que nous touchons sur la dalle.

Nous pouvons ensuite parcourir un tableau de référence (keyboard) afin de connaître la touche du clavier que nous avons actionné.

2.4.2 Carte Slave

La carte Slave va récupérer les inputs de sa dalle tactile, les interpréter (*cf.* point 2.4.1) et envoyer, via CAN (*cf.* point 2.4.3), l'index de la touche à la carte Master.

2.4.3 Communication CAN

La communication CAN est des plus simple. Le message ne comporte d'une seule donnée : l'ID du tableau keyboard de la touche appuyée ou relâchée.

2.4.4 Carte Master

La carte Master va récupérer les inputs de sa dalle tactile, les interpréter (*cf.* point 2.4.1) et envoyer, via UART (*cf.* point 2.4.5) au PC. Elle effectue la même chose pour toutes informations reçues par CAN de la part de la carte Slave.

2.4.5 Communication UART

Voici une trame UART envoyée par la carte Master :

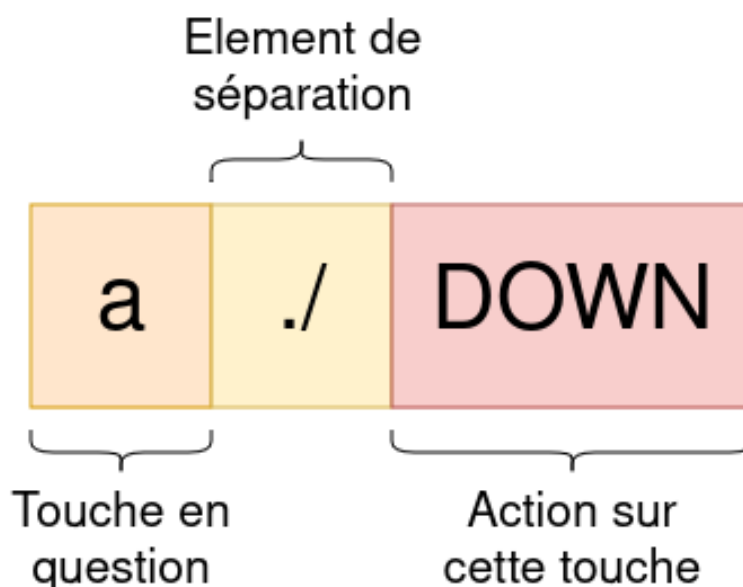


FIGURE 4 – trame UART

Elle est donc composée de la touche que le script Python devra actionner (a minuscule), d'un élément statique de séparation (./), ainsi que l'action à entreprendre sur la touche en question (DOWN ou UP).

2.4.6 Script Python

Le script Python récupère les trames UART (*cf.* point 2.4.5) de la part de la carte Master et les transforme en input clavier à l'aide de [pyautogui](#). Pour cela, il lui suffit de `split` la ligne UART reçue grâce à l'élément statique de séparation et d'effectuer l'action demandée sur la touche en question.