

Programmation temps-réel

Émetteur et récepteur lumineux

Claudio Sousa, David Gonzalez - Groupe 3

17/05/2018

1 État du projet

Le projet est en état de marche et toutes les fonctionnalités ont pu être implémentées.

L'émetteur fonctionne sans soucis.

Les 4 tâches du récepteur fonctionnent également bien, y compris sous forte charge.

2 Anomalies ou bugs

Les programmes fonctionnent sans anomalie.

3 Traces sous forte charge

3.1 Déroulement

Une trace complète a été prise à chaque étape de charge de l'émetteur afin de pouvoir comparer les effets. Voici donc ci-dessous 4 fois la trace avec les 4 sections (étapes) mises en évidence :

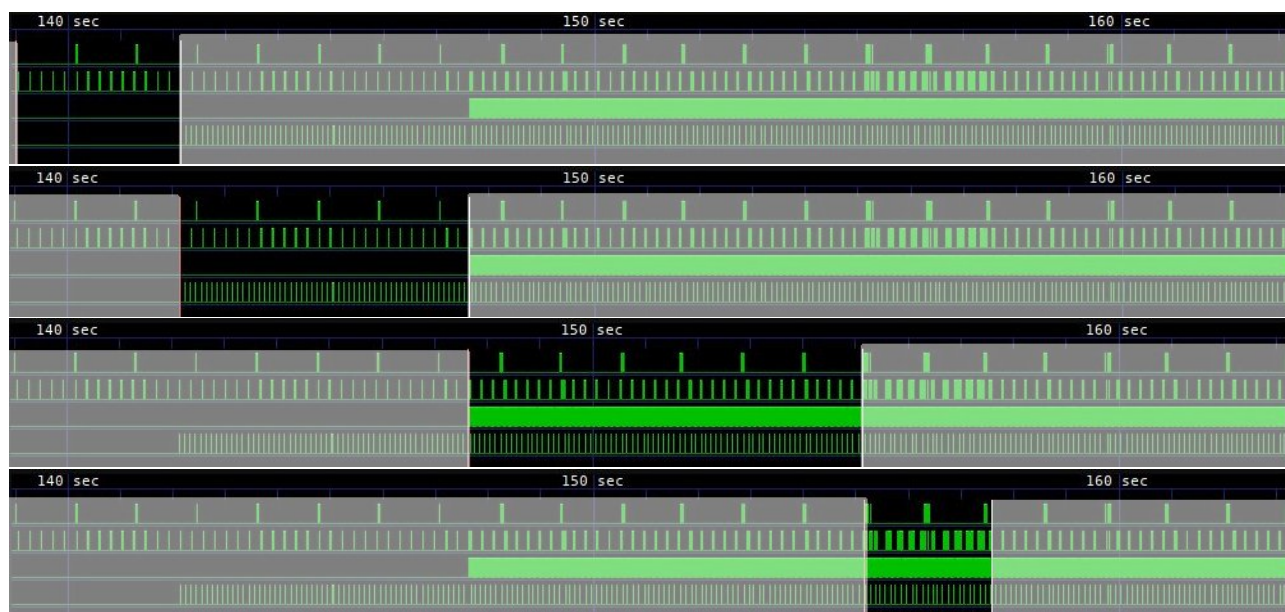


FIGURE 1 – Déroulement et traces globales sectionnées en 4 parties

Ces sections décrivent le déroulement de l'opération. Tout d'abord, le récepteur a été mis en marche avec les paramètres suivants :

- scroll fast (220 ms);
- load 0;
- leds off.

La première section est celle avec les paramètres décrits ci-dessus.

La deuxième section est lorsque la tâche des LEDs a été mise en marche (*leds on*).

La troisième section est lorsque la tâche de charge (load) a été mise en marche (*load 15*).

La quatrième et dernière section est celle où une commande *send* a été envoyée (*send test.txt*).

3.2 Section 1

Voici un zoom sur la section 1 avec les timings de chacune des tâches en exécution :

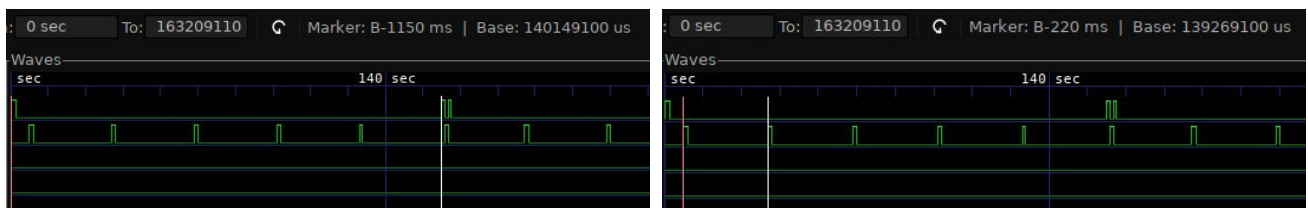


FIGURE 2 – Zoom sur la section 1

Seules deux tâches sont en exécution, puisque à ce stade, la tâche des LEDs et la tâche de charge ne sont pas actives.

Ici, le processeur ne fait pas grand chose et a largement le temps d'exécuter toutes les tâches en cours et de respecter les timings de la tâche 2 (commande à 220 ms). La tâche 1 a un interval de 1150ms, ce qui correspond à la valeur attendue ($4 \text{ instants/char} * 18 \text{ chars} * 8 \text{ ms} * 2 \text{ frames} = 1152 \text{ ms}$).

3.3 Section 2

Voici un zoom sur la section 2 avec les timings de chacune des tâches en exécution :

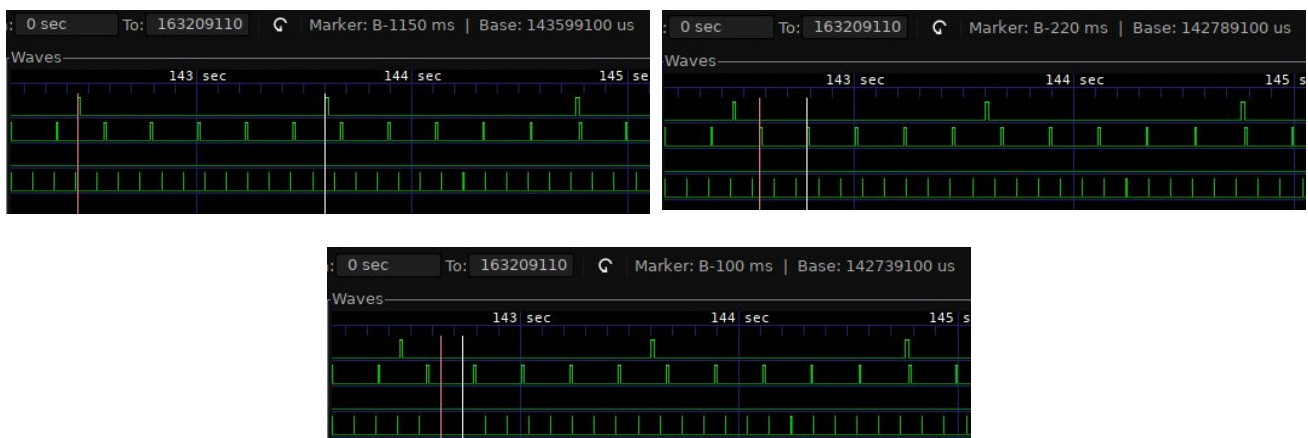


FIGURE 3 – Zoom sur la section 2

Pas grand chose à dire de plus par rapport à la première section. En effet, la tâche 4 (LEDs) a été activée, mais son temps d'exécution étant négligeable, elle n'affecte en rien le système.

3.4 Section 3

Voici un zoom sur la section 3 avec les timings de chacune des tâches en exécution :

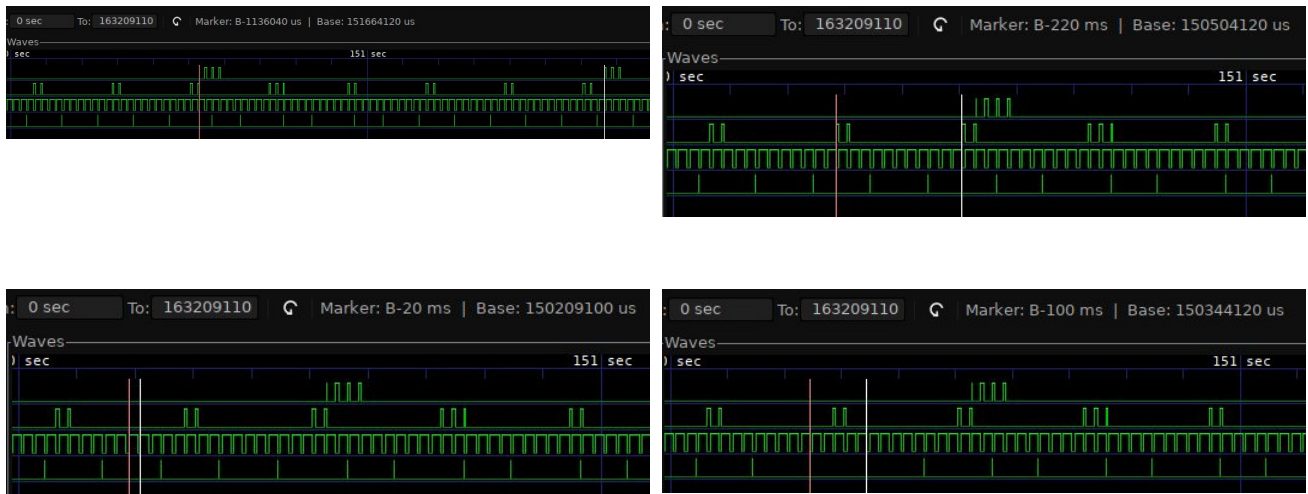


FIGURE 4 – Zoom sur la section 3

Ici, la tâche 3 (load) a été activée à son maximum (15 ms). On remarque tout de suite que les tâches ne peuvent plus être exécutées sans interruption du à la tâche de charge. Par conséquent, le temps de réponse se trouve augmenté, mais respecte toujours les contraintes de temps.

3.5 Section 4

La section 4 est le plus chargée de toutes avec la commande *send* envoyée.

Selon le trace ci-dessous, la tâche 1 a largement le temps de s'exécuter. On remarque une petite variation du temps d'exécution de la tâche, ce qui augmente la gigue temporel. La cadence étant donnée par le DMA du décodeur, la charge du CPU ne fait que retarder le décodage du message et n'affecte en rien la réception.

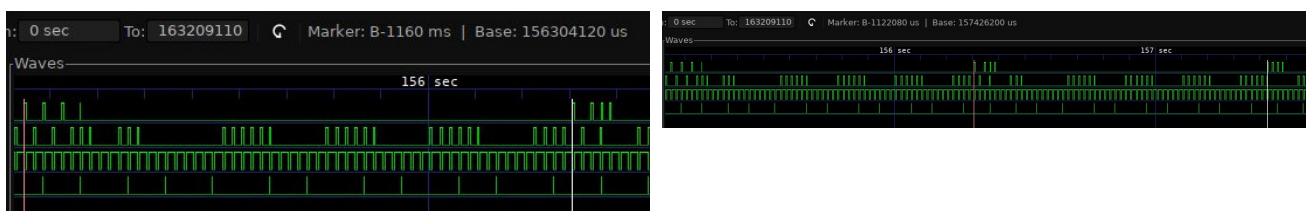


FIGURE 5 – Zoom sur la section 4, tâche 1

Concernant la tâche 2, la trace ci-dessous nous montre que son temps de réponse est d'environ 110 ms. Aucune gigue n'a pas été observée.



FIGURE 6 – Zoom sur la section 4, tâche 2

Concernant la tâche 3, son timing est parfait, puisque c'est la tâche avec la plus haute priorité.

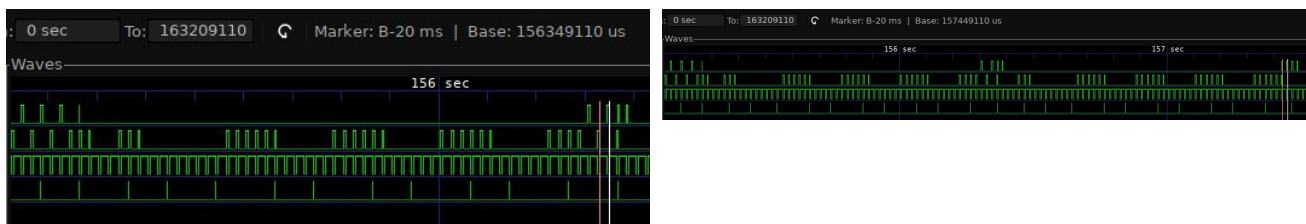


FIGURE 7 – Zoom sur la section 4, tâche 3

Finalement, pour la tâche 4, on observe que l'intervalle entre deux exécutions varie entre 80ms et 120ms. Cependant, ce n'est qu'une gigue qui ne donne pas lieu à une dérive, comme observé sur la trace ci-dessous :

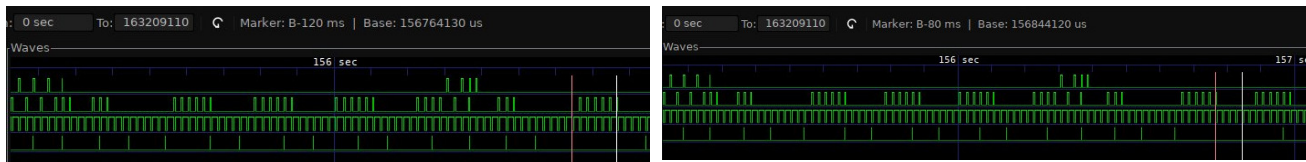


FIGURE 8 – Zoom sur la section 4, tâche 4

Lorsque la forte charge est passée, la période de la tâche est stabilisée :

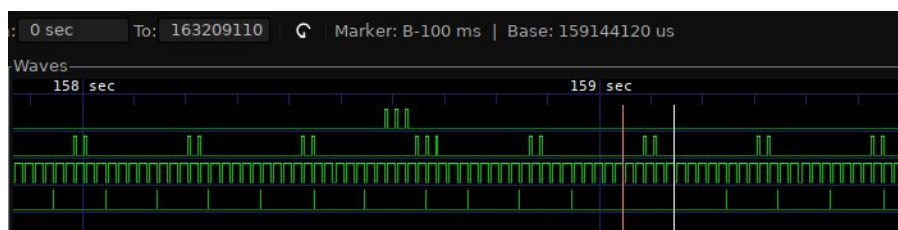


FIGURE 9 – Récupération du timing de la tâche 4