

## TP 10 DMA

Dans ce TP, nous allons observer les bénéfices qu'apportent le DMA en terme de copie de la RAM vers un périphérique. Codons une application simple, qui émet un message en boucle sur un UART, d'abord sans DMA et émettant donc une interruption pour chaque octet envoyé, puis avec DMA, en émettant une interruption uniquement à la fin de chaque message ; enfin, observons le gain en terme de temps processeur utilisé.

1. En reprenant le code du TP 6, écrivez un programme à base d'interruption qui envoie le message "mississippi\n" en boucle sur l'USART 1 (connecté à l'ordinateur via le port COM virtuel) à la vitesse de 115 200 bits par secondes. Vérifiez sur l'ordinateur que le message est bien reçu.
2. Modifiez le code de façon à ce que le message change à chaque nouvelle émission : on échangera par exemple deux lettres contigues aux positions 0 et 1, puis 1 et 2, puis 2 et 3 etc. (dans une fonction callback `HAL_UART_TxCpltCallback()`) item Configurez la broche de votre choix en sortie ; allumez cette broche quand vous entrez dans le *handler* de l'interruption de l'USART1, et éteignez-la quand vous en sortez. À l'aide de l'oscilloscope, mesurez :
  - (a) le rapport du temps passé à l'intérieur de l'interruption sur le temps passé en dehors (donc au repos si vous avez appelé `__WFI()` dans la boucle principale).<sup>1</sup>
  - (b) la fréquence de cette interruption. Comment l'expliquez vous ?
3. Modifiez maintenant votre programme de façon à utiliser un DMA pour copier automatiquement chaque octet du message quand l'UART est prêt à en accepter un nouveau :
  - (a) consultez le manuel de référence<sup>2</sup>, pour savoir sur quel périphérique (DMA1/2) et sur quel canal (0-8) est branché la ligne DMA de USART1.
  - (b) allumez et configurez le DMA de façon à ce qu'il copie le message en continu, octet par octet, de la mémoire vers le périphérique (fonction `HAL_DMA_Init()`).
  - (c) liez le *handle* de l'UART à celui du DMA (macro `__HAL_LINKDMA()`).
  - (d) armez l'UART (`HAL_UART_Transmit_DMA()`) pour qu'il informe le DMA qu'il est prêt à envoyer ;
  - (e) initiez le transfert DMA (`HAL_DMA_Start()`) en indiquant l'adresse source (l'adresse du buffer contenant le message), l'adresse cible (le registre de transmission USART1->TDR) et le nombre d'octet à copier ;<sup>3</sup>
4. À ce stade, le DMA va copier le message de la mémoire vers l'UART en continu, sans jamais le changer ; vérifiez que c'est bien le cas. Maintenant, nous allons demander au DMA de nous prévenir au moyen d'une interruption quand celui-ci a fini un transfert, pour que nous puissions changer le message.

---

1. Sur l'oscilloscope, cette mesure s'appelle le "Duty cycle" (Duty=xxx).

2. [http://www.st.com/content/ccc/resource/technical/document/reference\\_manual/4a/19/6e/18/9d/92/43/32/DM00043574.pdf/files/DM00043574.pdf/jcr:content/translations/en.DM00043574.pdf](http://www.st.com/content/ccc/resource/technical/document/reference_manual/4a/19/6e/18/9d/92/43/32/DM00043574.pdf/files/DM00043574.pdf/jcr:content/translations/en.DM00043574.pdf)

3. Attention, il y a un bug dans HAL : on doit forcément appeler `HAL_DMA_Start()` après avoir appelé `HAL_UART_Transmit_DMA()`, et pas l'inverse.

- (a) dans le NVIC, activez la ligne d'interruption correspondant au canal du DMA que vous utilisez (`HAL_NVIC_EnableIRQ()`). Définissez un *handler* pour cette interruption, qui appelle la fonction HAL de *dispatch* `HAL_DMA_IRQHandler()` ;
  - (b) (re)définir la fonction de *callback* `HAL_UART_TxCpltCallback(UART_HandleTypeDef *huart)` exécutée après chaque transfert (appelée cette fois-ci par `HAL_DMA_IRQHandler()`) et échangeant les lettres du message.
5. Grâce à la broche de sortie configurée plus haut, mesurez à l'oscilloscope :
- (a) le rapport de temps passé dans l'interruption du DMA sur celui passé à l'extérieur. Quel pourcentage d'occupation du CPU avez-vous économisé ?
  - (b) la fréquence de cette interruption. Comment l'expliquez-vous ?
- Répétez ces mesures pour un message plus long (disons, 100 octets)
6. Analysez le message reçu sur l'ordinateur. Vous devriez voir certaines anomalies dans les messages successifs : certaines paire de lettres ne sont pas échangées correctement. Pourquoi ?
7. Pour y remédier, implémentez le double buffering comme vu en cours.