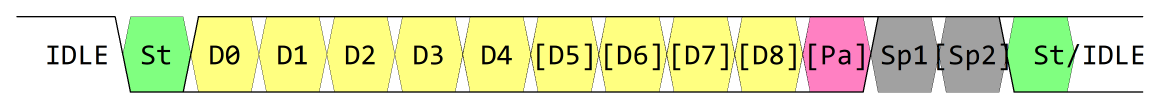
* + 1. La liaison série RS232 :

Le but est d’identifier quelques fonctions essentielles pour les différentes cartes ou SBC qui intègrent toutes par défaut un ou plusieurs ports UART. L’utilisation du port série RS232 de la carte Raspberry Pi dépend de sa déclaration et de son paramétrage et les bibliothèques associées sont capables de gérer tous les flux UART avec une carte à microcontrôleur. Les informations ci-dessous définissent une solution logicielle pour gérer une communication série avec les UARTs des microcontrôleurs AVR.

La ressource UART contient deux registres de 8 bits, un pour l’émission et un pour la réception et chaque UART est configurable en choisissant 3 paramètres essentiels dans une communication série :

* La parité paire, impaire ou aucune.
* La longueur du mot à transmettre de 5 à 9 bits.
* Le nombre de bit de STOP de 1 à 2.



* + - 1. *UART par interruption :*

Pour les microcontrôleurs AVR, il faut éviter d’utiliser les bibliothèques qui proposent un UART logiciel dont le principe est essentiellement basé sur une scrutation du port série. Une solution recommandée est d’utiliser le port de communication par interruption qui garantit ainsi aucune perte de donnée.

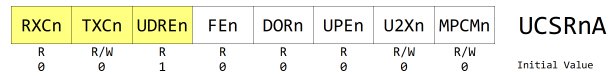
Tous les UARTs disposent de deux registres UDRn, un pour l’émission et un pour la réception mais identifiés par le même nom. La distinction se fait uniquement en fonction du type d’instruction assembleur ou de la commande utilisée en langage C (write/read).

Les 3 types d’interruption possibles pour chaque UART sont activés par un bit d’autorisation dans le registre UCSRnB. Ces interruptions sont actives uniquement si le bit d’autorisation globale I est activé dans le registre SREG.

Les évènements liés à ces ressources produisent des interruptions, donc un déroutage du programme principal vers des vecteurs d’interruption spécifiques (adresses matérielles) qui sont liés à l’activation de drapeaux du registre UCSRnA. Il faut privilégier les drapeaux qui se réinitialisent automatiquement une fois que le saut au vecteur d’interruption est appliqué.

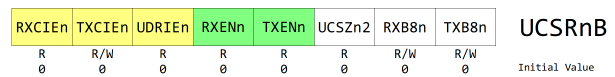
Il faut identifier les 3 types d’interruptions pour chaque UART :

* Une interruption à la réception d’un octet complet dans le registre de réception UDRn. Le drapeau RXCn sera activé quand un octet sera disponible à la lecture et une interruption sera possible si le bit d’autorisation RXCIEn du registre UCSRnB est activé. Le drapeau de l’interruption RXCn est désactivé automatiquement dès qu’on vient lire l’octet dans le buffer de réception UDRn et ce processus est vraiment pratique pour une lecture par interruption d’une trame de données.
* Une interruption à l’émission activée par la non présence d’un octet dans le registre d’émission UDRn. Ce drapeau UDREn devient par défaut actif dès l’initialisation du microcontrôleur même sans autoriser d’interruption particulière ou globale. Ce drapeau est associé à l’autorisation UDRIEn qui doit être activée pour obtenir un saut d’interruption. Ce drapeau revient à 0 quand on écrit une donnée dans le buffer d’émission et donc il n’y a pas de gestion automatique de ce drapeau par les actions matérielles du microcontrôleur qui dépendent exclusivement des actions du code programmé. C’est pourquoi on déconseille l’utilisation de cette interruption et on lui préfère l’interruption qui active le drapeau TXCn à l’émission.
* Une autre interruption à l’émission active le drapeau TXCn du registre UCSRnA lorsqu’un caractère est entièrement sérialisé (shifted out) hors du registre d’émission UDRn. L’avantage par rapport au drapeau précédent UDREn est que le drapeau TXCn est automatiquement remis à 0 quand il y a un saut au vecteur de l’interruption. Cette particularité nous affranchit de scruter le drapeau TXCn et permet de se focaliser sur l’action du programme principal en fonction de critères induits par les interruptions. Pour émettre une trame, on utilise une boucle qui écrit un caractère dans le buffer UDRn et qui attend que le drapeau logiciel défini par le programmeur soit actif pour écrire un deuxième caractère. Ensuite en tenant compte d’autres critères come l’identification d’un caractère de fin de trame et le drapeau logiciel activé suite à un saut à l’adresse de l’interruption, on arrête la boucle de scrutation.



* + - 1. *L’autorisation des interruptions :*

Les autorisations de toutes les interruptions sont appliquées dans le registre UCSRnB ainsi que l’activation de l’émission et de la réception de l’UART.



* + - 1. *Code C associé et recommandations :*

Pour éviter deux interruptions équivalentes en termes d'action attendue, si on active l'interruption UDRIEn, il faut désactiver l'interruption TXCIEn et réciproquement.

Les fonctions d’interruption associées à chaque drapeau du registre UCSRnA doivent contenir un minimum de code qui correspond à l’activation d’un drapeau défini par l’utilisateur. Les deux drapeaux pour la transmission TxRx\_Interrupt et UDR\_Interrupt étant complémentaires, il est préférable de favoriser l’interruption liée à TXCn.

L’interruption liée à UDREn est activée par défaut au démarrage quand le buffer d’émission UDRn est vide. Le drapeau UDREn est à 0 seulement quand UDR est plein ce qui signifie que l'interruption sera continuellement appelée sans process de désactivation automatique induit par le saut du programme principal vers un vecteur d’interruption.

Le drapeau TXCn sera activé quand le caractère du buffer UDRn est entièrement sérialisé, c’est-à-dire que le dernier bit a été transmis. Sachant que ce drapeau est remis à 0 automatiquement dès que le programme principal est dérouté vers le vecteur d’interruption correspondant, l’activation d’un drapeau défini par le programmeur permettra de ne pas retourner dans la fonction d’interruption. Avec un buffer vide l’interruption ne pourra pas être réactivée indéfiniment car il faut un process de sérialisation complet pour régénérer une interruption et c’est ce principe qui est préférable à l’utilisation du drapeau UDREn du registre UCSRnA.

Modification ajoutée