

Bus de communication UART

Prof : KAMAL BOUDJELABA

29 octobre 2022

Table des matières

1	Communication UART	1
1.1	Exemple	3

Liste des figures

1	Topologie de la communication UART	1
2	Trame UART	2
3	Exemple de trame UART	2
4	Données envoyées par l'UART	3
5	Données envoyées par l'UART avec un bit de parité paire	3

1. Communication UART

UART (Universal Asynchronous Receiver/Transmitter : réception et transmission asynchrones universelles) est un protocole de communication série qui facilite la communication hôte avec les périphériques auxiliaires. Principalement, il permet la transmission de données en série. Les périphériques UART sont également faciles à utiliser, car aucune horloge n'est envoyée entre eux, tout est asynchrone et la transmission est bidirectionnelle.

L'UART est considéré comme "universel" car les paramètres, y compris la vitesse de transfert et la vitesse des données, sont configurables par le développeur. Comme l'UART est également asynchrone, il n'utilise pas de signal d'horloge pour synchroniser les bits de sortie de l'UART émetteur avec les bits d'échantillonnage de l'UART récepteur. Sans horloge, l'UART de réception et de transmission doit être sur le même débit en bauds (débit binaire). Cela permet au système de savoir où et quand les bits ont été cadencés.

Essentiellement, UART gère la communication des signaux entre deux appareils via les moyens suivants :

- Simplex : transmission de données unidirectionnelle
- Semi-duplex (half duplex) : communication non simultanée entre deux appareils dans les deux sens
- Duplex intégral (full duplex) : transfert de données simultané dans les deux sens

Dans la communication UART, deux UART communiquent directement entre eux ; l'UART sur l'appareil émetteur (UART émetteur) reçoit des données parallèles de la CPU et les convertit en données série. Ces données série sont transmises à l'UART sur l'appareil récepteur (UART récepteur). L'UART récepteur reconvertit les données série reçues en données parallèles et les envoie au CPU. Pour que l'UART convertisse les données série-parallèle et parallèle-série, des registres à décalage sur l'UART émetteur et récepteur sont utilisés.

Dans la communication UART, seuls deux fils sont nécessaires pour la communication : les données circulent de la broche Tx de l'UART émetteur (émetteur Tx) à la broche Rx de l'UART récepteur (récepteur Rx).

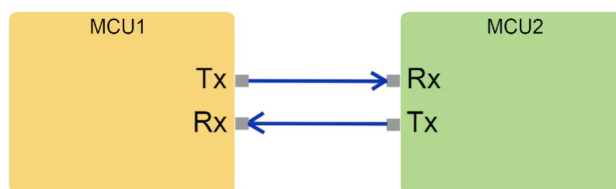


Figure 1. Topologie de la communication UART

Les données UART sont envoyées sur le bus sous la forme d'un paquet. Un paquet se compose d'un bit de départ, d'une trame de données, d'un bit de parité et de bits d'arrêt. Le bit de parité est utilisé comme mécanisme de contrôle d'erreur pour garantir l'intégrité des données.

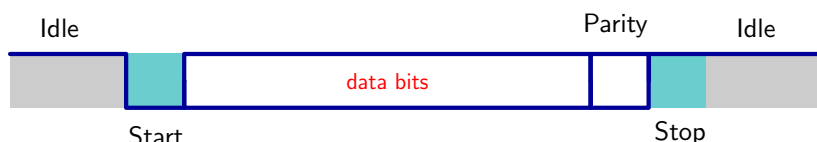


Figure 2. Trame UART

Les trames UART contiennent des bits de départ (Start) et d'arrêt (Stop), des bits de données (Data bits) et un bit de parité (Parity) optionnel

Un niveau de tension "haut" est utilisé pour indiquer un "1" logique et un niveau de tension "bas" est utilisé pour indiquer un "0" logique. Comme le protocole UART ne définit pas des tensions ou des gammes de tensions spécifiques pour ces niveaux, le niveau haut est également appelé "marque" et le niveau bas est appelé "espace".

- L'état d'inactivité (état idle : où aucune donnée n'est émise), la ligne est maintenue au niveau haut. Cela permet la détection facile d'une ligne ou d'un émetteur endommagé.
- L'émetteur doit signaler que les bits de données arrivent car le protocole UART est asynchrone. Cela est réalisé en utilisant le bit de départ. Le bit de départ est une transition de l'état d'inactivité (idle) à un état bas, et immédiatement suivi par les bits de données utilisateur.
- Les bits de données sont des données utilisateur et viennent immédiatement après le bit de départ. Il peut y avoir de 5 à 9 bits de données utilisateur, bien que 7 ou 8 bits soient les plus courants. Ces bits de données sont généralement émis avec le bit le moins significatif en premier.
- Bit de parité (optionnel) est inséré entre la fin des bits de données et le bit d'arrêt. La valeur du bit de parité dépend du type de parité utilisé (paire ou impaire)
 - Dans la parité paire, ce bit est réglé de manière que le nombre total de 1 dans la trame sera pair.
 - Dans la parité impaire, ce bit est réglé de manière que le nombre total de 1 dans la trame sera impair.
- Après que les bits de données (et éventuellement de parité) soient terminés, le bit d'arrêt indique la fin des données utilisateur. Le bit d'arrêt est une transition de retour à l'état haut (à l'état d'inactivité) ou restant à l'état haut pour une durée de bit supplémentaire. Un second bit d'arrêt (optionnel) peut être configuré, généralement pour donner le temps au récepteur d'être prêt pour la prochaine trame.

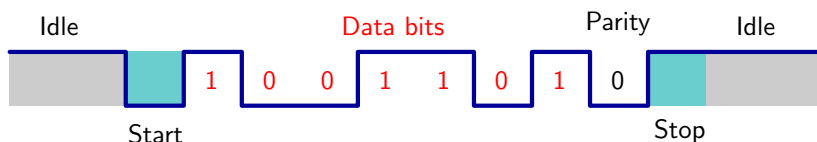


Figure 3. Exemple de trame UART

1.1 Exemple

On souhaite transmettre la lettre capitale "S" en 7 bits ASCII. Pour cela, on doit inverser l'ordre des bits car le bit le moins significatif est émis en premier.

Le code ASCII de S = 83_{dec} , 53_{hex} , 1010011_{bin}
 \Rightarrow Ordre LSB : 1100101

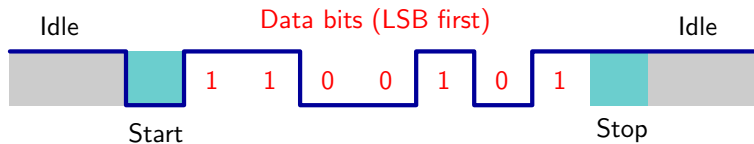


Figure 4. Données envoyées par l'UART

La lettre capitale "S" (1010011) contient un total de trois "0" et quatre "1". Si on utilise la parité paire, le bit de parité est "0" car il y a déjà un nombre pair de "1". Si on utilise la parité impaire, alors le bit de parité doit être "1" afin de permettre à la trame d'avoir un nombre impair de "1".

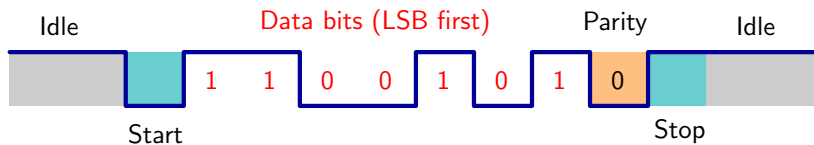


Figure 5. Données envoyées par l'UART avec un bit de parité paire