

9. Opérations périphériques - contrôleurs spécifiques (série et parallèle), interfaces - caractéristiques et standards

(Suite)

9.3. Opérations périphériques - contrôleur spécifique série. Interfaces - caractéristiques et standards /71/

Comme on a montré dans le cours précédant, sous le nom de l'*interface d'utilisateur* on va comprendre le milieu, à l'aide duquel l'utilisateur profite de la fonctionnalité du programme concret ou du dispositif d'un ordinateur donné. Cette *interface* donne la possibilité de l'utilisateur de poser les données d'entrée et de recevoir les résultats du dispositif concret de l'ordinateur, vers lequel (dispositif) l'utilisateur s'était tourné.

On va se permettre ici de donner une seconde définition qu'on use dans la science *informatique*, où sous la notion d'*interface d'utilisateur* on comprend: « L'ensemble des clés, des menus et/ou la vue générale des fenêtres (*Windows*) d'une concrète application (programme).

9.3.1. Contrôleur spécifique série

Ce circuit intégré transforme les données parallèles, traitées inférieurement (dans l'ordinateur) en données série et l'inverse. Cette transformation est exécutée dans le bloc «*Interface*» - *UART I 8250A* (Intel), montré sur la **Figure 6** - (*UART – Universal Asynchronous Receiver Transmitter*).

Dans les nouveaux ordinateurs on utilise le circuit intégré *UART 16450* ou le plus puissant *UART 16550*, les deux permettant de plus haute rapidité d'échange.

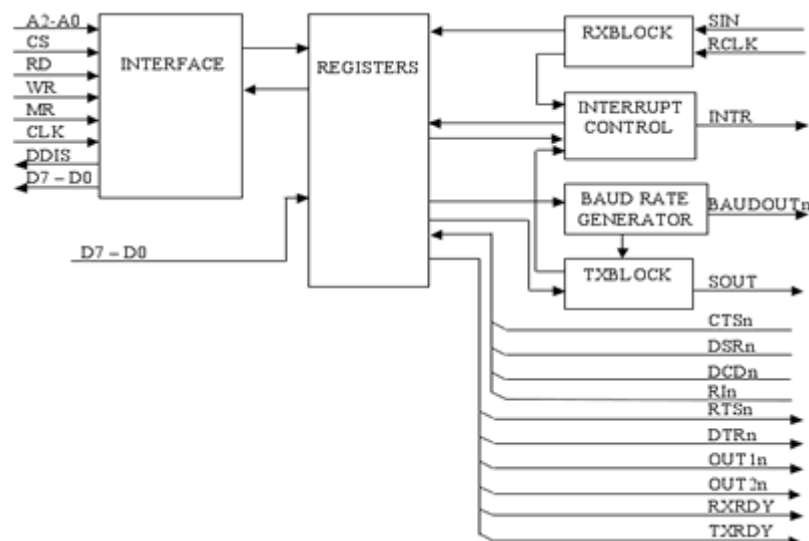


Figure 6. Bloc-diagramme du circuit intégré I 8250A

Le bloc des registres – «**REGISTERS**» - du circuit intégré *UART I 8250A* (Intel) est composé des types suivants de registres:

- **Divisor Latch Register - DLAR – Registre de Serrure de Division:** c'est le registre de la vitesse d'échange des données, mesurée en *Bauds*.
- **Receiver / Transmitter Register – RxTxR – Registre de Recevoir / Transmettre:** c'est le registre de recevoir/transmettre, réalisé avec un tampon (buffer) doublé et traite les données reçues ou transmises.
- **Interrupt Enable Register – INER – Registre Interruptif et Accessible:** c'est un registre de permettre les interruptions, parce qu'en même temps puissent être reçues plusieurs interruptions.
- **Interrupt Identification Register – INIR – Registre Interruptif et Identifiant:** c'est un registre d'identification de l'interruption donnée, qui est seulement lu et donne l'information de l'état actuel du traitement des interruptions.
- **FIFO Control Register – FIFO CR – Registre de Gestion de la mémoire de type FIFO:** il est réalisé seulement dans les *UART 16550*. Ce registre est dédié exclusivement pour l'enregistrement et se trouve à la même adresse du registre d'identification de l'interruption donnée, lequel peut être seulement lu, comme on a remarqué plus haut.
- **Line Control Register – LCR – Registre de Gestion de la Ligne:** c'est un registre de gestion des lignes, qui définit les paramètres de transfert et peut être aussi bien lu que enregistrer des données dans lui.
- **Modem Control Register – MCR – Registre de Gestion du Modem:** c'est un registre de gestion du modem, activé lorsqu'il existe une ligne de modem, lequel registre et peut être aussi bien lu que enregistrer des données dans lui.
- **Line Status Register – LSR – Registre de Gestion de la Ligne:** c'est un registre de l'état des lignes, qui sert d'établir l'état de fonctionnement actuel du circuit intégré UART utilisé.
- **Modem Status Register – MSR – Registre de l'Etat du Modem:** c'est un registre de l'état du modem, qui informe pour la situation des signaux d'entrée.

9.3.2. Interfaces - caractéristiques et standards

Tout ordinateur dispose au moins d'une **interface série** pour brancher la souris. Cette interface peut être utilisée aussi bien pour brancher un traceur, ou des modems, ou des imprimantes, tous avec interface série. La notation répandue américaine est **RS 232**, tandis que le standard européen est notée comme **V24/V28**. En différence avec l'interface parallèle où le transfert des données avec la périphérie est de huit bits et dans des cas rares il est bidirectionnel – **IEEE1248**, pour l'interface série en principe on utilise le transfert bidirectionnel. Ce transfert s'effectue successivement sur une ligne et avec le protocole activé entre l'ordinateur et la périphérie. Dans l'interface série le niveau haut (1) est avec une valeur de moins 3Volts à moins 15Volts et le niveau bas est de plus 3Volts à plus 15Volts. Pratiquement les 15Volts sont limités à 12Volts par le bloc de tension de l'ordinateur.

9.3.2.1. Caractéristiques principales et paramètres dans l'échange série des données – standard série RS 232:

La transmission des données pour l'interface **RS 232** peut se réaliser en *régime/transmission/échange synchrone* et en *régime/transmission/échange asynchrone*. Pour le régime / transmission synchrone l'émetteur et le récepteur peuvent fonctionner aux différentes fréquences d'horloge tout en signalant la validité des données sur les lignes séparées. Avec les données l'émetteur transmet sur une ligne séparée le signal d'horloge. Les données transmises se complètent en un bloc, qui est formé par des signes de gestion définis. Tous les symboles sont codés en **ASCII**. A l'aide de ces signes le générateur d'horloge du récepteur (de l'ordinateur qui reçoit) se synchronise par les données arrivantes:

SYNC	Bloc des données 1	ETB	SYNC	Bloc des données 2	EOT
-------------	---------------------------	------------	-------------	---------------------------	--------------	------------

Figure 28. Transfert des données en blocs

Le début du bloc est marqué par **SYNC** et la fin du premier bloc - **Bloc des données 1** par les symboles **ETB**- *End of Transmission Block - Fin du bloc transmis*. Sauf cela avec **ETB** est signalé que pratiquement la transmission est finie et que un nouveau bloc suit - **Bloc des données 2** et etc. La transmission finit avec l'émission du signe **EOT** - *End of Transmission – Fin de la transmission*

Dans la transmission série la plus utilisée - *asynchrone*, le récepteur et l'émetteur fonctionnent en une même fréquence d'horloge et la synchronisation s'effectue par un *bit de START*, un *bit de STOP* et éventuellement par un *bit de contrôle de parité*. Dans ce régime on n'utilise pas des lignes de transmettre du signal d'horloge et l'information de la synchronisation accompagne tout symbole unique grâce à quoi l'efficacité du régime synchrone est beaucoup plus basse que celle du régime asynchrone. Les paramètres de transmission de l'interface **RS 232** doivent être toujours identiques pour le récepteur et l'émetteur et sont les suivants: **Vitesse de transmission**, mesurée en **Bauds**, **Bit de START**, **Bit de STOP**, **Nombre des bits de données**, **Contrôle par parité**:

- **Vitesse de transmission**, mesurée en **Bauds**: elle montre le nombre des changes de l'état du signal, qui sont transmises pour une unité de temps et s'appelle aussi la fréquence de pas qui se définit avec l'unité de mesure **Baud**. Des valeurs typiques de cette vitesse sont: 50, 75, 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (Bauds). La vitesse en «Bauds» ne doit pas être confondue avec la vitesse de transmission, définie en **Bits pour seconde – bps**, le plus souvent elles ne sont pas les mêmes, parce qu'on utilise des différentes méthodes de compression de données. Alors la vitesse de transmission en **bps** puisse être beaucoup plus haute que la fréquence de pas en **Bauds**, qui est typique en utilisant des modems. C'est pourquoi pour l'interface série asynchrone se définit et la vitesse de modulation. La sûreté de transmission est déterminée de la longueur des câbles, du niveau des signaux et de la vitesse de transmission. Si le transfert/échange est accompagné de brouillages, la vitesse de transmission doit être diminuée.

- **Bit de START**: par ce bit est défini le début des données, qui sont émises;
- **Bit de STOP**: ce bit définit la fin des données émises;
- **Nombre des bits de données**: ce nombre est le plus souvent sept (7) ou huit (8). Des erreurs dans le transfert peuvent être reconnues par le **bit de contrôle par parité**.

Il existe les trois suivantes possibilités pour ce but:

1. *No parity check - sans contrôle par parité*: le contrôle par parité ne se réalise pas;

2. *Even parity check – contrôle par parité*: l'émetteur compte tous les bits à valeur 1 (unité) et si ce nombre est impair établit le bit de parité en état unité(1). Si ce nombre est pair – ce bit de parité s'établit en état zéro(0). Le récepteur additionne le nombre des bits de données, qui sont en unité(1), avec le bit de contrôle pair. La somme doit être un nombre pair. Si la somme est impair il y a une erreur dans le transfert (l'échange);

3. *Odd parity check – contrôle par imparité*: la méthode correspond à celle du contrôle par parité, mais ici la somme des bits, établis à unité(1) et le bit de contrôle par parité est toujours impair.

Le format d'émission des données doit être toujours le même pour les deux dispositifs. Dans les programmes de transmission les paramètres sont donnés à l'aide du menu.

Pour le cas le plus simple l'échange par **RS 232** se réalise par un bus de trois lignes:

RxD – ligne de réception;

TxD – ligne d'émission;

GND – terre.

La vitesse de transfert de **RS 232** est 20 kbps

9.3.2.2. *Standards des interfaces série*:

Sauf le standard **RS 232**, traité ci-dessus, il existe et sont utilisées pour de différentes applications de l'ordinateur les interfaces série suivantes:

➤ **RS 422**:

C'est une interface symétrique, réalisant une défense améliorée contre les brouillages. Elle fonctionne à tensions différentielles. La vitesse maximale de transfert est 10 Mbps au lieu de 20 kbps du **RS 232**, tout en réalisant des systèmes avec au maximum de dix récepteurs – **Simplex**. Une application typique est la transmission de données d'un ordinateur sur plusieurs unités périphériques comme dans les systèmes distribuifs de mesure.

➤ **RS 423**:

C'est une réalisation plus simple du **RS 422**, mais exécutée comme **RS 232** asymétriquement, qui puisse fonctionner aux vitesses d'échange jusqu'à 100 kbps au lieu de 20 kbps du **RS 232**.

➤ **RS 485**:

Jusqu'ici les interfaces série présentées permettent seulement un émetteur pour le système total d'interface. Pour le **RS 485**, qui représente un élargissement du **RS 422**, sont définies les caractéristiques électriques d'une interface bidirectionnelle de bus. Par cette interface on peut brancher jusqu'à trente-deux (32) unités, fonctionnant comme récepteurs et comme émetteurs – **Demi Duplex**. Une application typique est la réalisation des mini-réseaux, par exemple entre le processeur central et

les différents systèmes de caisse dans un super market. Une autre application est dans l'automatisation industrielle pour la réalisation des bus de systèmes de champ – **Fieldbus**.

➤ **TTY (TeleTYpe):**

Premièrement ce standard était prévu pour la gestion des unités lointaines d'écriture. Il est marqué comme **Interface de courant** ou **Interface 20 mA**. Il est utilisé pour séparer les potentiels des unités connectées – dans la technique de médecine, comme dans les gestions / contrôleurs programmables, gérés à l'aide de l'ordinateur. Cette interface fonctionne généralement avec deux cercles de courant à l'aide de quatre (4) fils. Le passage du courant de 20 mA (aussi bien de 40 mA ou de 60 mA) est unité(1), tandis que la manque du courant est zéro(0). Les vitesses typiques sont 110, 300, 1200 et 2400 bps à une distance entre les deux unités jusqu'à 1 km. Les cercles de courant dans les unités actuelles gèrent directement, au lieu de relais électriques, des optocoupleurs (photo coupleurs).

➤ **IEEE1394 :**

Ce type de standard d'interface série apparaît en 1995. **FireWire** est le nom commercial donné par Apple à une interface série multiplexée, aussi connue sous la norme **IEEE 1394** et également connue sous le nom d'interface **i.LINK**, nom commercial utilisé par Sony. Il s'agit d'un bus informatique véhiculant à la fois des données et des signaux de commandes des différents appareils qu'il relie. Connu encore comme **FireWire** – «**Branchement à chaud**» il s'est transformé en standard industriel, dédié au traitement multimédia. Au début, développé pour la compagnie *Apple*, maintenant cette interface série multiplexée de haute vitesse jouit d'une accroissante popularité. Il est utilisé pour tous les unités de «branchement à chaud» comme caméras digitales, unités de disques durs, imprimantes, scanners, CD-RW, DVD. Cela le fait préférable et parfois, comme le cas des caméras digitales, le seul choix raisonnable. Le câble le plus souvent est composé de six(6) fils, avec une longueur maximale pour tous les protocoles **FireWire** de 4,5 m. Il existe également une transmission par fibre optique, très coûteuse mais permettant d'atteindre 100 m. Le câble comprend des fils d'alimentation qui puissent assurer un courant jusqu'à 1,5 A pour une tension d'alimentation entre 8 et 30-40 V. C'est d'ici que vient le nom **FireWire**. Comme on réclame des exigences plus hautes pour les câbles d'**IEEE 1394**, ils coutent quelques fois de plus que les câbles **USB**.

Les caractéristiques d'**IEEE 1394** sont:

- «**Branchement à chaud**»: l'interface permet aux unités d'être branchées et déclenchées à / de l'ordinateur fonctionnant;
- Un câble d'interface mince et commode;
- Une configuration facile;
- Une haute vitesse d'échange de données;
- Une possibilité assurée d'élargissement;
- Un équipement de peu de valeur;
- Une possibilité d'alimenter les unités avec le câble d'interface.

La version courante du standard permet la transmission des données à l'aide de l'interface **FireWire** avec des vitesses de: 100, 200 et 400 Mbps. Dans les spécifications suivantes les vitesses maximales atteignent 800, 1600 Mbps et même plus hautes.

Le standard permet le fonctionnement des unités, branchées à un bus, par rapport à leurs possibilités, dans un moment donné de fonctionner par couples à une vitesse différente. Par exemple un couple des unités plus lentes peuvent échanger des données à la vitesse de 100Mbps, tandis qu'un autre couple des unités de haute vitesse, branchées au même bus – à la vitesse de 400Mbps.

➤ **USB.**

USB – Universal Serial Bus est un standard, le plus souvent utilisé de connecter un dispositif externe à un ordinateur. Ce standard est élaboré d'un consortium de sept compagnies: **Compaq, Digital, IBM, Microsoft, NEC, Northern Telecom.**

L'interface **USB** est avec la spécification ouverte, qui signifie, que chacun puisse produire ce dispositif d'entrée / sortie, sans payer la licence. Il est créé encore en 1994 et longtemps était nommé **Useless Serial Bus – Inutile Bus Série**. A peine en 1999 ce type d'interface obtient la popularité.

Ces caractéristiques techniques / paramètres sont les suivants:

- **Rapidité (Débit) de transfert** pour **USB version 1.1** est maximum 12Mbps;
- **USB** utilise quatre fils: T_x, R_x, terre, tension;
- **USB** câble puisse être utilisé aussi bien comme un type du dispositif d'alimentation;
- Dans un réseau local à l'aide de l'**USB nœud de réseau (USB network hub)** peuvent être connectés jusqu'à 127 dispositifs à **USB version 2.0**.

L'**USB version 2.0** est la plus répandue version au marché. Elle se caractérise avec son débit jusqu'à **480 Mbps**, très proche de celle de **Fire Wire**. Assi bien elle est compatible avec les versions plus vieilles de ce type d'interface.

En 2008, l'**USB 3.0** introduit le mode **vitesse supérieure (Super Speed)**, qui débite théoriquement à **5 Gbps**. Mais ce nouveau mode utilisant un codage des données de type 8b/10b, la rapidité de transfert réelle est de seulement **4 Gbps**. L'**USB 3.0** délivre une puissance électrique de 4,5 watts.

Les nouveaux périphériques disposent de connexions à 6 contacts au lieu de 4, mais la compatibilité ascendante des prises et câbles avec les versions précédentes est assurée. En revanche, la compatibilité descendante est impossible, les câbles USB 3.0 de type B n'étant pas compatibles avec les prises **USB 1.1/2.0**, mais il existe des adaptateurs.

Début 2010, l'**USB 3.0** est introduit dans des produits grands publics. Les prises femelles correspondantes sont signalées par une couleur bleue. Apparition aussi des prises femelles **USB rouges**, signalant une puissance électrique disponible supérieure, et appropriée au chargement rapide de petits appareils y compris (à condition de le paramétrer dans le BIOS), lorsque l'ordinateur est éteint.

Un standard **USB 3.1 à 10 Gbps** est annoncé en août 2013 ; ses spécifications techniques sont finalement publiées par le consortium **USB Implementers Forum** en août 2014.

L'**USB 3.1** promet des débits doubles de ceux de l'**USB 3.0**, soit **10 Gbps**. Le nouveau standard (câbles, interface) sera rétro compatible avec l'**USB 3.0** et l'**USB 2.0**. Toutefois la connectique change, elle sera plus fine et n'imposera pas de sens de branchement. Pour tout de même permettre la connexion vers des connecteurs **USB 2.0** et **3.0**, le standard devra prendre en compte la possibilité d'avoir des adaptateurs passifs (à l'inverse des adaptateurs **Lightning**, le connecteur réversible qu'Apple a lancé avec l'**iPhone 5** en 2012), pour garder une taille réduite et un coût de fabrication mesuré. Cette nouvelle connectique se nommera **Type C**.

L'**USB 3.1** introduit l'**Alternate Mode (Mode Alternatif)** qui est utilisé par exemple pour faire passer de la vidéo avec le protocole **Display Port**, contrairement à des technologies existantes comme **Display Link** qui encapsulaient de la vidéo au travers du protocole **USB** standard. Ce mode permet aux

constructeurs d'étendre l'usage de l'**USB** à d'autres fonctions. Pendant la négociation de protocole de l'**USB-PD 2.0** (*Power Delivery*, rendu obligatoire avec les connecteurs **Type-C**), un identifiant assigné par l'**USB-IF** est échangé pour déterminer le mode de fonctionnement alternatif. On peut utiliser seulement les canaux supplémentaires, mais aussi ceux destinés à l'**USB 2.0**.

La table suivante donne une vue sur toutes les modifications des **USB**:

Débit des normes USB						
Version	USB 1.0	USB 1.1	USB 2.0	Wireless USB	USB 3.0	USB 3.1
Année	1996	1998	2000	2005	2008	2013
Débit	1,5 Mbps	12 Mbps	480 Mbps	480 Mbps	5 Gbps	10 Gbps

Figure 29. Débit des normes USB

Protocole d'USB

La communication entre l'hôte (l'ordinateur) et les périphériques se fait selon un protocole basé sur l'interrogation successive de chaque périphérique par l'ordinateur. Lorsque l'hôte désire communiquer avec un périphérique, il émet un jeton (un paquet de données, contenant l'adresse du périphérique, codée sur sept bits) désignant un périphérique. Si le périphérique reconnaît son adresse dans le jeton, il envoie un paquet de données (de 8 à 255 octets) en réponse. Les données ainsi échangées sont codées selon le codage NRZI. Puisque l'adresse est codée sur 7 bits, 128 périphériques (2^7) peuvent être connectés simultanément à un port de ce type. Il convient en réalité de ramener ce chiffre à 127 car l'adresse 0 est une adresse réservée.

USB définit quatre types de transferts :

- **transfert de commande, utilisé pour l'énumération et la configuration des périphériques.** Il convient pour des données de taille restreinte ; il y a garantie de livraison (renvoi des paquets erronés);
- **transfert d'interruption**, utilisé pour fournir des informations de petite taille avec une latence faible. Ce ne sont pas des interruptions au sens informatique du terme : le périphérique doit attendre que l'hôte l'interroge avant de pouvoir effectuer un tel transfert. **Ce type de transfert est notamment utilisé par les claviers et les souris;**
- **transfert isochrone**, utilisé pour effectuer des transferts volumineux (bande passante garantie), et en temps réel. Il n'y a pas de garantie sur l'acheminement des données. **Ce type de transfert est utilisé pour les flux audio et vidéo;**
- **transfert en masse (bulk)**, utilisé pour transférer des informations volumineuses, avec garantie d'acheminement, mais sans garantie de bande passante. **Ce type de transfert est utilisé par les dispositifs de stockage.**

Il est possible de structurer la communication entre un hôte et un périphérique en plusieurs canaux logiques (*pipes* et *end points*) pour simplifier la commande du périphérique du port **USB**.