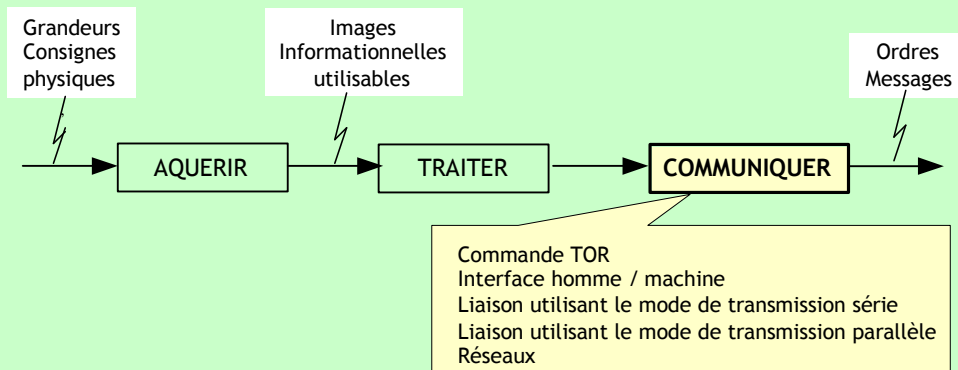


PRESENTATION

Après le traitement des informations, le résultat de traitement est communiqué à son environnement, qui est principalement représenté par :

- Les actionneurs, via les preactionneurs, qui agissent sur la matière d'œuvre ;
- L'Interface Homme/Machine qui reçoit de l'unité de traitement des messages renseignements sur l'état du système.

La position de la fonction "Communiquer" dans une chaîne d'information, ainsi que les différentes réalisations principales sont représentées par la figure suivante :



COMPETENCES ATTENDUES

A partir de tout ou partie d'un produit support avec son cahier des charges et son dossier technique :

- Enoncer les principaux paramètres caractérisant les différents types de liaisons
- Mettre en œuvre les éléments nécessaires à la réalisation d'une liaison

CHAPITRES INCLUS DANS LA FONCTION COMMUNIQUER

- Interfaçage en sortie
- Liaison parallèle - Liaison série

INTERFACE EN SORTIE

INTRODUCTION :

Dans un système automatisé, l'unité de traitement communique les informations traitées principalement vers :

- L'utilisateur, grâce aux organes de dialogue en sortie ? tel un voyant de signalisation ; il s'agit de l'interface Homme/Machine (IHM) ;
- des organes de puissance tel un moteur électrique triphasé ; il s'agit d'interface de puissance centré principalement sur l'isolation électrique ou galvanique.

1. INTERFACE HOMME/MACHINE (IHM) :

L'interface Homme/Machine en sortie est une fonction de communication indispensable pour bien gérer un système automatisé. Il consiste à fournir à l'opérateur l'ensemble des informations concernant l'état du système automatisé qu'on désigne par "messages". Le dialogue Homme/Machine se fait par l'utilisation de constituants regroupés dans ce qu'on appelle "pupitre" de commande. On en décrit quelques-uns des plus populaires.

1.1. Les voyants :

Les voyants sont des témoins lumineux qui constituent une interface de dialogue simple donnant à l'opérateur des informations sur l'état du système automatisé.

Les voyants possèdent un code de couleur (NORME EN 60204-1 :1992) qui permet d'orienter l'opérateur sur l'origine du message :

Fig 4. Exemple de voyant



COULEUR	SIGNIFICATION	EXEMPLE
ROUGE	Urgence ou condition dangereuse	Arrêt d'urgence
JAUNE	Condition anormale pouvant entraîner une situation dangereuse	Manque en matière d'œuvre
VERT	Préparation de conditions normales	Départ de cycle
BLANC	Information générale	Présence de la tension réseau

1.2. Les afficheurs numériques :

Dans certaines applications industrielles, le contrôle du système automatisé nécessite de surveiller les paramètres avec une grande précision qui n'est pas permise par les voyants.

Les afficheurs numériques permettent d'effectuer une surveillance précises et informent l'opérateur des résultats de mesure (température, pression, etc.) ou de comptage (nombre de cycle, quantité de pièces produites, etc.).

Fig. 5 Exemple d'afficheur numérique



1.3. Les afficheurs alphanumériques :

C'est un constituant de dialogue programmé permettant d'afficher des messages clairs concernant l'exploitation, le dépannage ou le réglage du système automatisé.

1.4. Les terminaux d'exploitation :

Les terminaux d'exploitation sont des constituants de dialogue programmés permettant à l'opérateur :

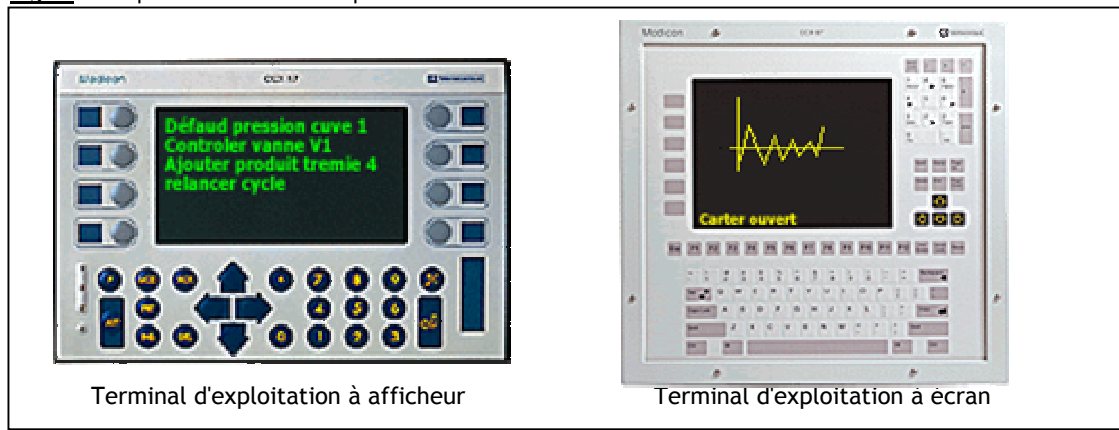
- D'être informé clairement sur l'état du système automatisé ;
- D'intervenir facilement et rapidement sur les paramètres de fonctionnement du système automatisé.

Ces terminaux peuvent être programmés à l'aide d'un terminal de programmation ou d'un micro-ordinateur, et peuvent éditer les événements et les messages grâce à des imprimantes.

On distingue deux types de terminaux d'exploitation :

- Les terminaux d'exploitation à afficheurs ;
- Les terminaux d'exploitation à écran permettant l'affichage simultané d'un nombre plus important de données en plus des représentations graphiques concernant le système automatisé.

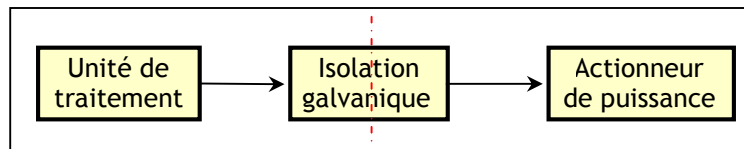
Fig. 6 Exemples de terminaux d'exploitation



2. ISOLATION GALVANIQUE :

2.1. Principe et rôle :

L'unité de traitement travaille typiquement avec une tension de 5V DC, nécessaire pour alimenter principalement les circuits intégrés logiques. Alors qu'un actionneur tel un moteur triphasé a besoin d'une forte tension (220V AC, par exemple). Pour protéger l'unité de traitement contre une éventuelle liaison directe avec la forte de tension, il faut une isolation électrique ou galvanique. Le principe est résumé par le schéma fonctionnel suivant :



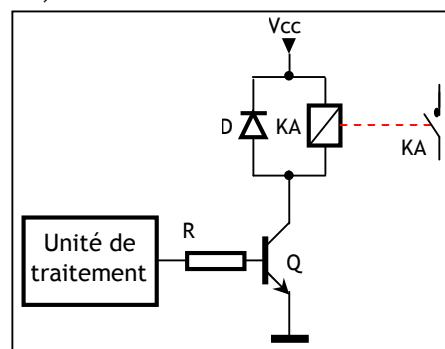
2.2. Exemple de schéma :

2.2.1- Principe :

Le schéma le plus classique dans ce domaine est à base de relais électromagnétique, ce qui donne le schéma de base suivant ; un tel montage, on le trouve déjà intégré dans un API :

- Quand l'unité de traitement communique l'ordre d'action (5 V), alors :

- Le transistor Q, travaillant en commutation se sature ;
- Le relais KA est excité avec la tension Vcc, ce qui ferme son contact ; ce contact isolé électriquement par rapport à la commande peut être utilisé pour établir le courant dans un organe de puissance (moteur, lampe, résistance chauffante, etc.) ;
- La diode D n'a aucun rôle dans cet état ; elle est bloquée.

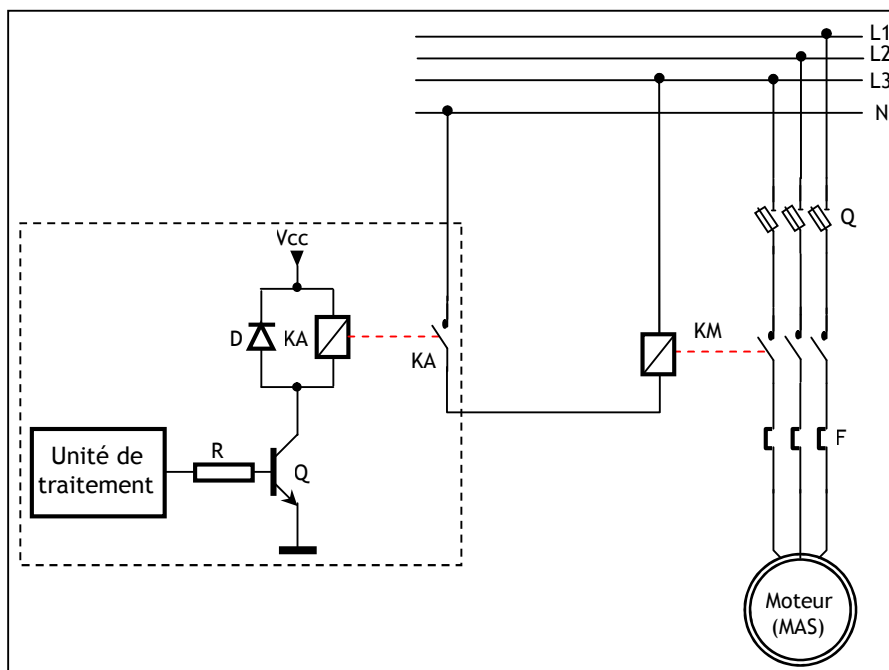


- Quand l'unité de traitement communique l'ordre de commande d'arrêt (0 V), alors :

- Le transistor Q se bloque ;
- Le relais KA est désexcité, ce qui ouvre son contact ; la charge est alors désalimentée.
- dans cet état, la diode D joue le rôle de roue libre pour protéger le transistor Q contre la surtension qui apparaît aux bornes de la bobine du relais KA, par dV/dt .

2.2.2- Exercice d'application: Commande d'un contacteur

L'étage déjà étudié est généralement intégré dans un API, ou à intégrer dans une carte de logique câblée ou une carte à microprocesseur (μP). Pour commander une charge de puissance relativement plus grande que celle pouvant être admissible par le relais (un moteur triphasé), on utilise un contacteur KM :



- Le moteur est muni de son circuit de puissance (Sectionneur Q, contacteur KM, relais thermique F) ;
- Si KA est excité, KM l'est de même, le moteur tourne ;
- Si KA est désexcité, KM l'est de même, le moteur s'arrête ;

LIAISON PARALLELE ET LIAISON SERIE

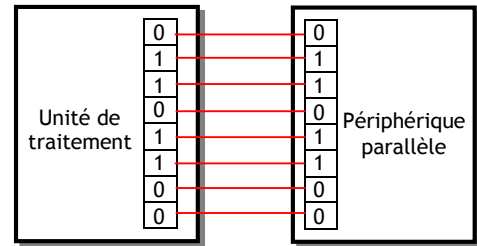
INTRODUCTION :

A l'intérieur, l'unité de traitement d'un système automatisé (API, etc.) communique les informations naturellement en parallèle sous un format de 8 bits ou plus. Mais, pour communiquer avec le milieu extérieur, l'unité de traitement communique en général en série, ce qui réduit le câblage.

1. LIAISON PARALLELE :

Exemple : Imprimante parallèle

Un texte est un ensemble de caractères ; chaque caractère est codé par un nombre de 8 bits (code ASCII). Avec une imprimante parallèle, le PC envoie le texte caractère par caractère, mais en parallèle, c'est à dire 8 bits / 8bits ;

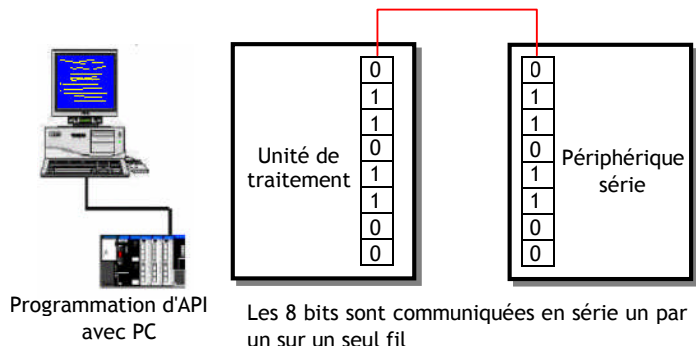


Les 8 bits sont transmis en parallèle sur 8 fils

2. LIAISON SERIE : PRINCIPE

2.1- Introduction

Lorsque la distance devient grande, la liaison parallèle devient techniquement difficile à réaliser à cause de la longueur du câblage et des parasites de transmission ; on utilise alors la liaison série. Par exemple, un API est généralement, programmé avec un PC ; le mode de communication dans cette situation est le mode série.



Les 8 bits sont communiquées en série un par un sur un seul fil

Dans une liaison série, on distingue plusieurs procédés et techniques de transmission ; dans ce qui suit on s'intéresse à la liaison série asynchrone, en particulier avec les normes RS 232 et RS 485.

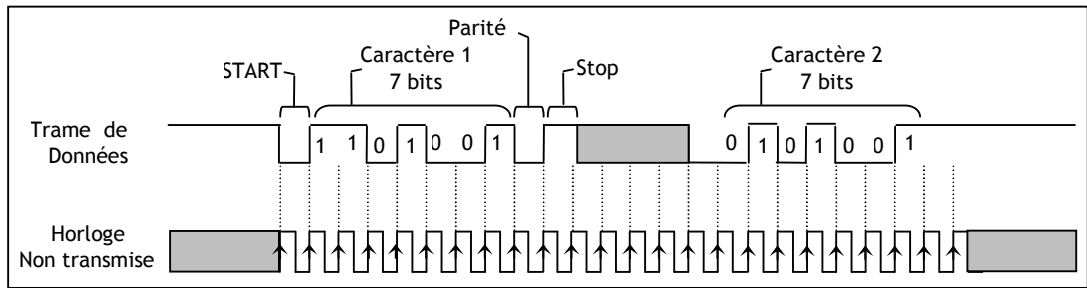
2.2. Principe de la liaison série asynchrone :

2.2.1- Format :

La liaison série asynchrone est orientée pour une transmission par caractères ; ces derniers sont envoyés individuellement et l'intervalle séparant 2 octets est quelconque. Ce mode asynchrone utilise un format où chaque caractère :

- A une longueur de 5 à 8 bits
- Est encadré par des bits délimiteurs :
 - 1 bit START au début de chaque caractère.
 - 1 à 2 bits STOP à la fin de chaque caractère.
- Peut être protégé contre les parasites de transmission , par un bit de parité optionnel (suivant la configuration du système) destiné à la détection d'erreurs ; il est généré à l'émission et testé à la réception. Il existe deux types de parité :
 - Parité paire : la parité est dite paire si le nombre de bits (donnée + bit parité) au niveau logique 1 est paire.
 - Parité impaire : la parité est dite impaire pour un nombre impaire de bits à 1.

L'ensemble {Bit Start, Bits données, Bit parité, Bits Stop} est appelé "trame" (frame). Voici un exemple de trame série asynchrone avec une parité paire ; elle représente la transmission de la lettre "K" dont le code ASCII est $(75)_{10} = (1001011)_2$: on remarque que le bit LSB est transmis le premier :



2.2.2- Fonctionnement :

Le fonctionnement est donc comme suit :

- Au repos, la ligne de transmission se trouve dans l'état logique 1 ;
- Au début de la transmission d'un caractère, on commence par le bit de Start, qui dure une période d'horloge ;
- On enchaîne par les bits du caractère (suivant la configuration du système), en commençant par le LSB ;
- On termine par le bit de stop.
- A la réception de chaque caractère, l'initialisation a lieu par la transition (haut-bas) du bit START qui assure la synchronisation des deux horloges (émetteur et récepteur) . Dès la réception du bit Stop, il n'y a plus de synchronisation.

3. NORME RS232 :

3.1- Liaison possible :

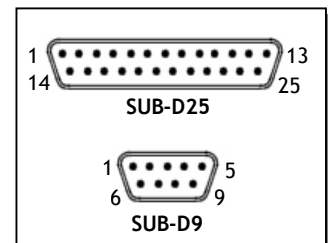
La norme RS232 définie par l'EIA (Electrical Industry Association), correspond à la norme ISO 2110. Elle permet une liaison "point à point". Il ne peut y avoir que 2 éléments communicants.

3.2- Caractéristiques :

3.2.1. Définition des signaux et connectique :

La liaison RS232 est une interface de tension pour la transmission série aussi bien synchrone qu'asynchrone, utilisée pour les liaisons point à point. Elle est définie pour un connecteur SUB-D25 ou SUB-D9. Elle comporte plusieurs signaux qu'on peut rassembler en deux groupes de fonctions :

- Signaux de communication principaux : Tx et Dx.
- Signaux de dialogue "optionnels" : RTS, DTR, etc.



3.2.2. Longueur de ligne et vitesse :

La norme RS232 est aussi caractérisée par :

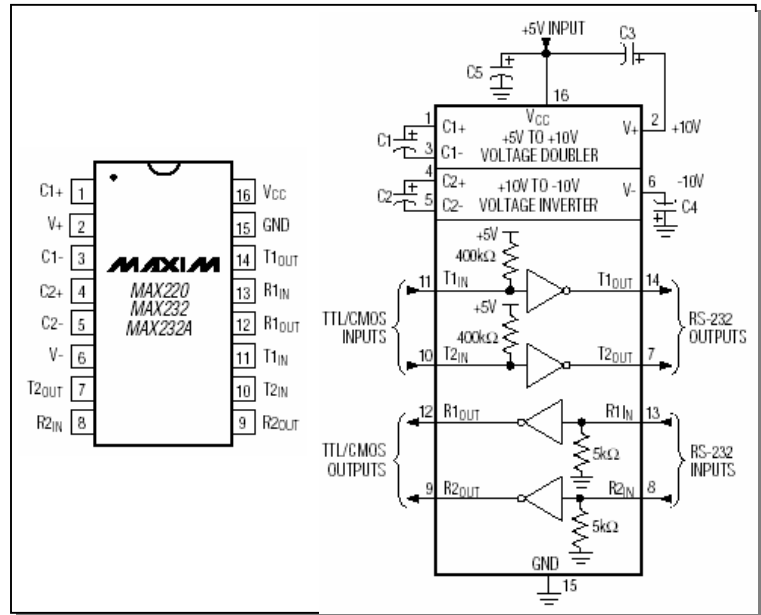
- La longueur maximale du câble qui est d'environ 15 mètres.
- Le débit maximal qui est à présent de 20Kbits/s. La norme prévoit ainsi des débits (bits/s) de 75, 150, 300, 600, 1200, 2400, 4800, 9600 et 19200.

3.2.3. Les niveaux de tension :

Pour la résistance au bruit, les niveaux de tension de la RS232 sont plus grands que ceux de la TTL/CMOS. L'équivalence avec les niveaux logiques sont décrits par le tableau suivant :

N. logique	Niveau électrique
'0' logique	+12V (de +5V à +15V)
'1' logique	-12V (de -5V à -15V)

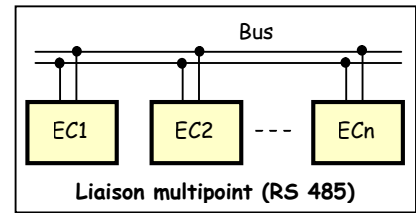
Les circuits logiques à l'origine de la transmission sont compatibles TTL/CMOS, il faut alors des circuits d'adaptation à la norme RS232 :



4. NORME RS485 :

4.1. Liaison possible :

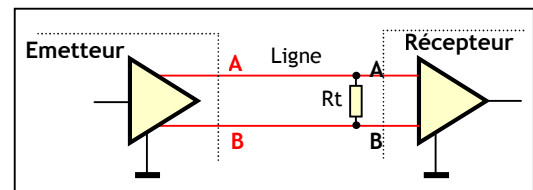
La norme RS 485 est définie par l'EIA correspondant à la norme ISO 2593. Elle permet une liaison "multipoint", c'est à dire, entre plusieurs Eléments Communicants (EC1, EC2,..., ECn). Dans ce cas, il faut qu'il n'y aie qu'un seul élément (1/n) qui émet dans le Bus ; le reste des éléments reçoit (il est à l'écoute). Dans cette liaison multipoint, on désigne la ligne de transmission de "Bus", car c'est comme une route de bus qui dessert plusieurs stations. Il est évident que pour qu'une communication soit possible, il faut que chaque EC dispose d'une interface pour le bus, plus précisément la possibilité de se mettre en haute impédance ; ainsi, on évite les conflits de Bus, cas où 2 EC émettent sur le Bus.



4.2- Caractéristiques :

4.2.1- Définition des signaux et connectique :

La norme RS485 est surtout utilisée dans les réseaux locaux industriels aussi bien point à point que multipoints. Elle utilise un support de transmission différentiel : le signal est transformé en deux signaux complémentaires (A et B), ce qui assure une résistance aux parasites industrielles et augmente la longueur maximale de la ligne ; la norme ne précise pas de connecteur spécifique. R_t : Résistance de terminaison de câble. Elle boucle la ligne sur son impédance caractéristique et minimise le bruit pour une meilleure transmission.



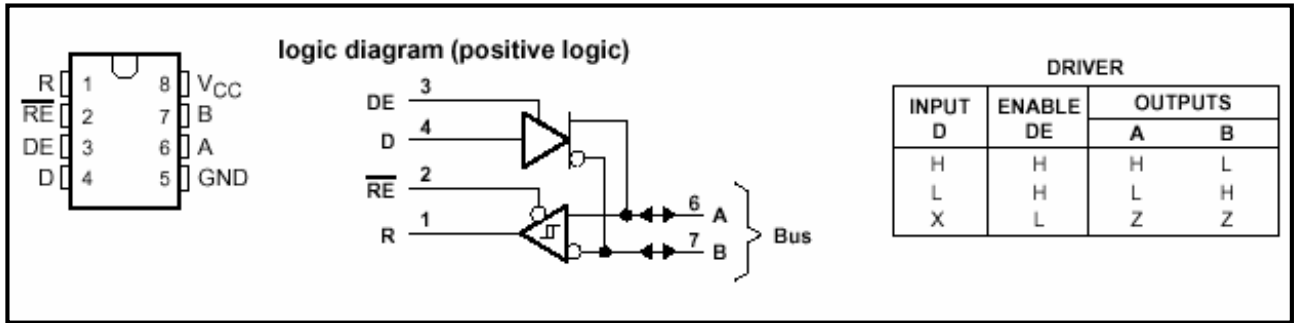
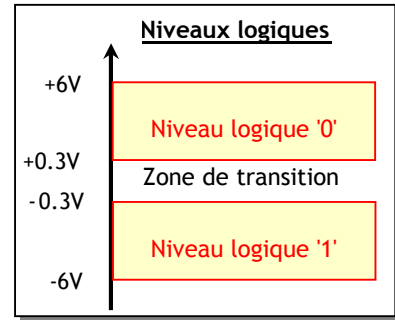
4.2.2. Longueur de ligne et vitesse :

La norme RS485 est aussi caractérisée par :

- La longueur maximale du câble qui est de 1000 m.
- Le débit maximal qui est à présent de 100 Kbits/s. La norme prévoit ainsi des débits (bits/s) de 75, 150, 300, 600, 1200, etc.

4.2.3. Niveaux de tensions :

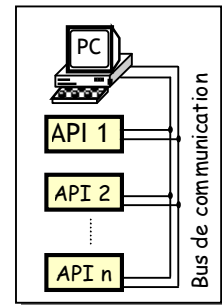
L'équivalence avec les niveaux logiques sont décrits par le graphique ci-contre. Les circuits logiques à l'origine de la transmission sont compatibles TTL/CMOS, il faut alors des circuits d'adaptation à la norme RS485, pour convertir une tension bipolaire en une tension différentielle. Parmi les circuits les plus utilisés à cette fin, on trouve l' AD485 et SN75176 :



4.3. Notion de protocole :

La figure ci-contre montre une architecture typique dans un environnement industriel, avec la norme RS485. Les n API de commande et le PC de supervision sont reliés donc en réseau ; ils peuvent donc échanger des informations. Cela a pour avantage :

- Une programmation structurée ; par exemple, pour un système automatisé à plusieurs postes, on réserve à chaque poste un API ;
- Un gain en câblage dans l'application ;
- Une facilité de maintenance.

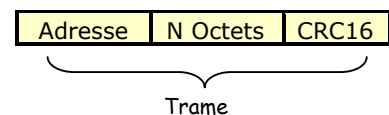


Il va sans dire que le Bus est partagé et doit donc connaître un "arbitrage" ; en effet, d'après la structure physique du réseau (norme RS485), il ne peut y avoir qu'un seul élément qui émet sur le bus, le reste écoute ; d'où la nécessité de s'accorder sur des règles de communication, qu'on appelle **protocole**. Un protocole doit donc résoudre les problèmes liés aux questions suivantes :

- Quel est le format de la trame, etc. ?
- A qui s'adresse la trame sur le bus ?

A titre d'exemple, on donne le principe d'un protocole largement diffusé dans ce domaine ; il s'agit de "ModBus" de MODICON. C'est une structure "Maître/Esclave" (Master/Slave). Dans ce protocole , il y a un seul maître (exemple le PC) et n esclaves (exemple les API). Dans le cas de la figure ci-dessous :

- Le PC est le maître ; les n API sont les esclaves ;
- Chacun des API a une adresse ;
- Le PC envoie une demande à un API et attend une réponse ;
- L'API interrogé répond à la demande du PC ;
- Chaque demande du maître ou réponse d'esclave est un ensemble d'octets (trame) qui a le format ci-contre :
 - Adresse : 1 octet représentant l'adresse de l'esclave ;
 - N octets : Ces N octets représentent l'objet de la demande du maître ou de la réponse d'un esclave ;
 - CRC16 : 2 octets de détection d'erreur, calculé suivant un algorithme précis, d'après les octets (Adresse + N Octets).



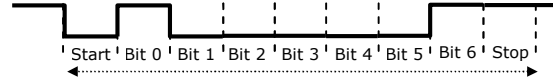
EXERCICES RESOLUS

Pour une configuration avec 7 bits de données, pas de parité et un bit de stop :

- 1.1- Donner la trame pour la transmission de la lettre "A", en indiquant la signification des différents bits.
- 1.2- Calculer le rendement de ce mode transmission, en calculant le rapport : (Bits de données)/(Bits de la trame).

CORRIGE :

1.1- Le code ASCII de A est $(65)_{10} = (1000001)_2$.



1.2- On a 7 bits de données à transmettre, mais on en transmet 10 → Rendement = $7/10=70\%$.

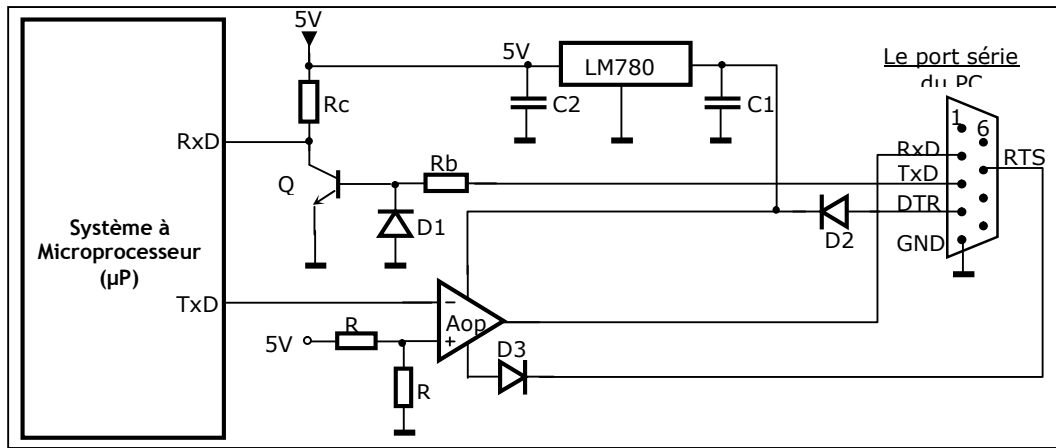
EXERCICES NON RESOLUS

Le schéma ci-dessous représente un convertisseur TTL/RS232 et un convertisseur RS232/TTL. La communication entre un système à μP et un PC se fait à l'aide d'une liaison série asynchrone RS 232 en utilisant :

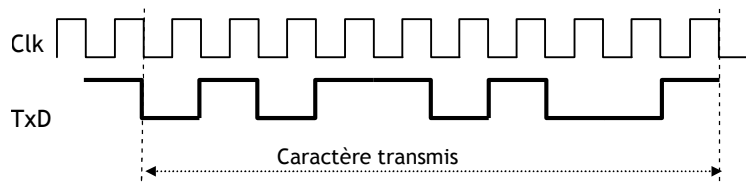
- ✓ Du côté système à μP , l'interface série est compatibles TTL ;
- ✓ Du côté PC, le port série COM1 ou COM2 utilise la norme RS 232 caractérisée par :
 - Une tension +12 V pour le niveau bas ;
 - Une tension -12 V pour le niveau haut.

Les signaux RTS et DTR non utilisés, alors ils sont maintenus à -12V et +12V pour servir d'alimentation.

- 1- La conversion RS232/TTL se fait par le montage autour du transistor Q.
 - 1.1- Préciser les valeurs de la tension de la ligne RxD côté μP correspondantes respectivement au niveau haut et au niveau bas de TxD côté PC.
 - 1.2- Quel est le rôle de la diode D1 ?
- 2- La conversion TTL/RS232 se fait par le montage autour de l'amplificateur opérationnel Aop.
 - 2.1- Préciser les valeurs de la tension de la ligne RxD côté PC correspondantes respectivement au niveau haut et au niveau bas de TxD côté μP .
 - 2.2- Quel est le rôle des diodes D2 et D3 ?



- 3- L'allure de la ligne TxD côté μP , lors de la transmission d'un caractère de 8 bit (D0 à D7), est donnée par le chronogramme suivant :



- 3.1- A chaque période d'horloge, correspond un bit. Donner le nom de chaque bit.
- 3.2- Tracer l'allure de la ligne RxD côté PC correspondant.