



PRÉSENTATION

- Audrey SIEPRAWSKI
- Concepteur logiciel
- Naval Group







- ISEN Promotion 2000
- o Option « Informatique Temps Réel »



A18 D:

Characteristics

Body diameter: 18 inches (46.5 cm)

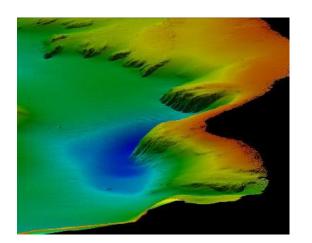
Length: 4.5 to 5.5 m Weight: 550-690 kg



Endurance: up to 24 hours Max speed: up to 6 knots Nominal speed: 3/4 knots Operational depth: 5-3000 m



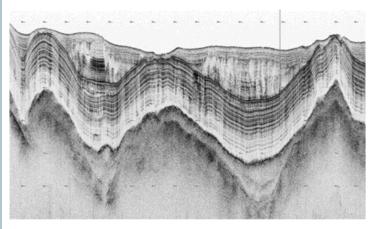
Wideband Multibeam Sonar (WBMS)



Synthetic aperture sonar (SAS)



Sub Bottom Profiler (SBP)



Side Scan Sonar (SSS)

HTTPS://WWW.YOUTUBE.COM/WATCH?V=UFIR2MRZBVM&FEATURE=YOUTU.BE



K-Ster:

Characteristics

- Length: 1.5 m
- Weight: 50 kg
- 2 horizontal and 2 vertical thrusters
- Dual frequency sonar for long range target detection and accurate approach
- Color video camera with automatic day/night function and searchlight

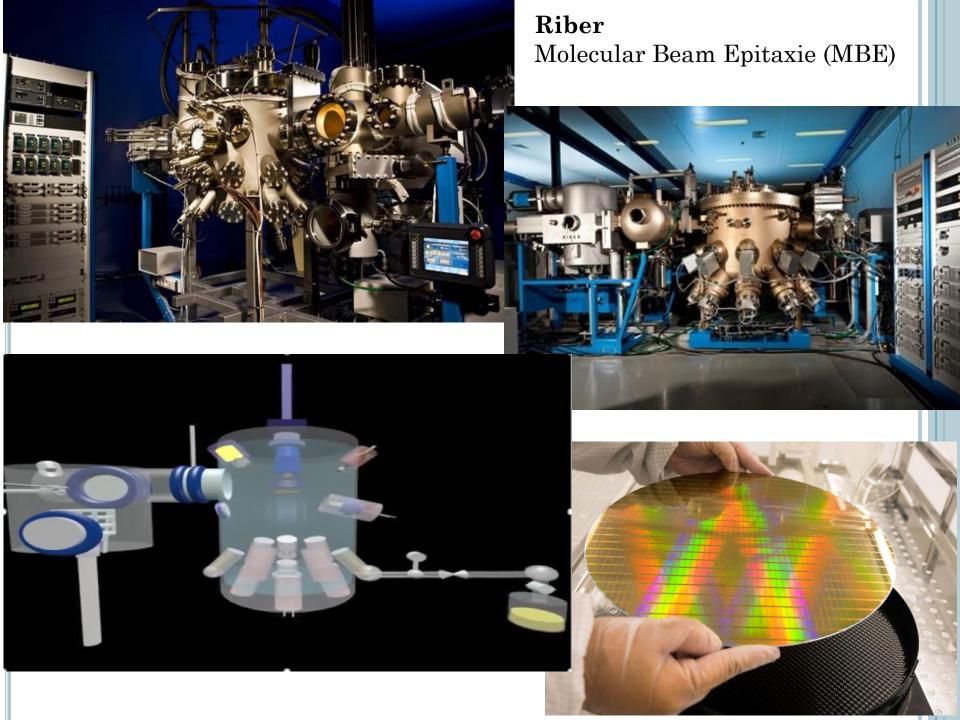
Operational Performances

- Endurance: up to 2 hours
- Max speed: up to 6 knots
- Nominal speed: 0 to 3 knots
- Max operational depth: 300m
- Operational range: up to 2000m









Jalan MBE Lab



Department of Chemical Engineering and Materials Science

University of Minnesota

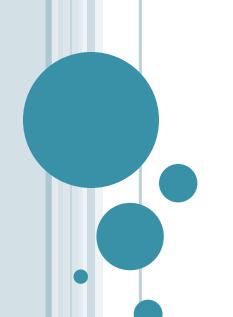
Group Webpage: www.research.cems.umn.edu/jalan

Animation by Christopher Cheng 3rd year undergraduate student CEMS, UMN



SOMMAIRE

- LES SYSTÈMES DE SUPERVISION,
- LES DIFFÉRENTES TECHNOLOGIES DE COMMUNICATION
 - COMMUNICATION SUR PORT SÉRIE
 - ⇒ Réalisation d'un programme de test type « HyperTerminal ».
 - COMMUNICATION SUR PORT IP
 - ⇒ Réalisation d'un programme de test type client/serveur.
 - PROTOCOLE DE COMMUNICATION: EXEMPLE DE DEUX PROTOCOLES NMEA ET MODBUS
 - Protocole Nmea:
 - o Format des trames Nmea,
 - Exemple d'un trame GPS,
 - Protocole Modbus:
 - o Format des trames d'échanges,
 - o Calcul du CRC (Cyclical Redundancy Check).
- RÉALISATION DU DRIVER D'UN RÉGULATEUR DE TEMPÉRATURE : EUROTHERM 2704



LES SYSTÈMES DE SUPERVISION

DÉFINITION DES SYSTÈMES DE SUPERVISION

Un logiciel de supervision est un système informatique interactif qui se situe entre:

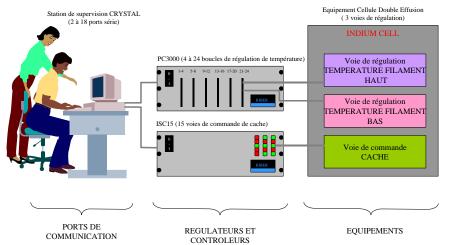
• Les unités de traitement (automates, régulateurs, chaînes d'acquisition, cartes E/S, ...),

ET

la gestion de la production.

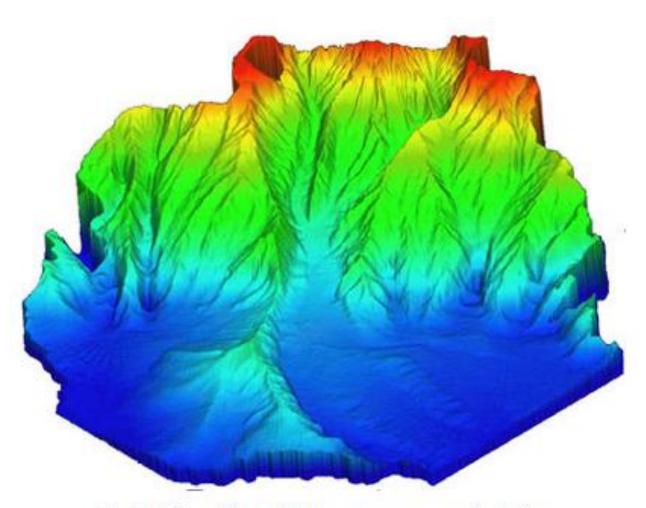
Il vise à fournir aux opérateurs les informations leur permettant de prendre les bonnes décisions au bon moment.

- <u>Acquérir</u> les données du système en temps réel par l'intermédiaire de capteurs,
- <u>Centraliser</u> ces données => vecteur d'état du système,
- Effectuer des <u>traitements</u> en vue de l'exploitation => traitement du signal (ex: filtrage), moyenne,
- <u>Automatiser et contrôler</u> la production => pilotage des équipements, recette de production,
- Assurer la <u>sécurité machine</u> en cas d'évolution anormale des paramètres clés,
- <u>Mémoriser</u> les mesures effectuées sur la machine et permettre leur restitution sous la forme de rapports imprimés





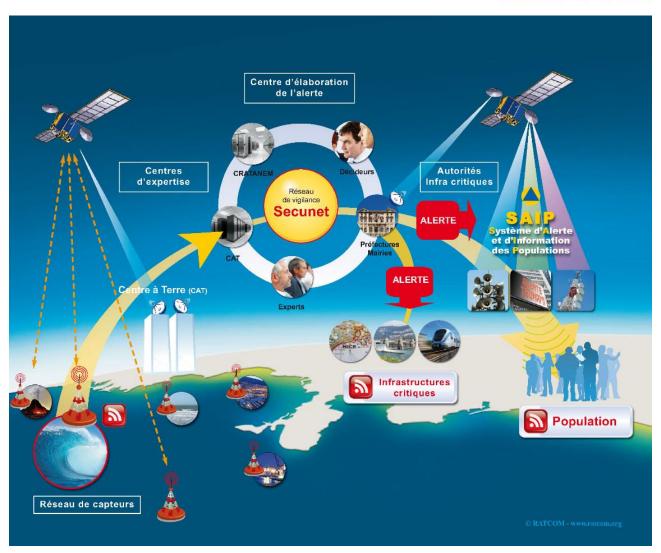
PROJET RATCOM



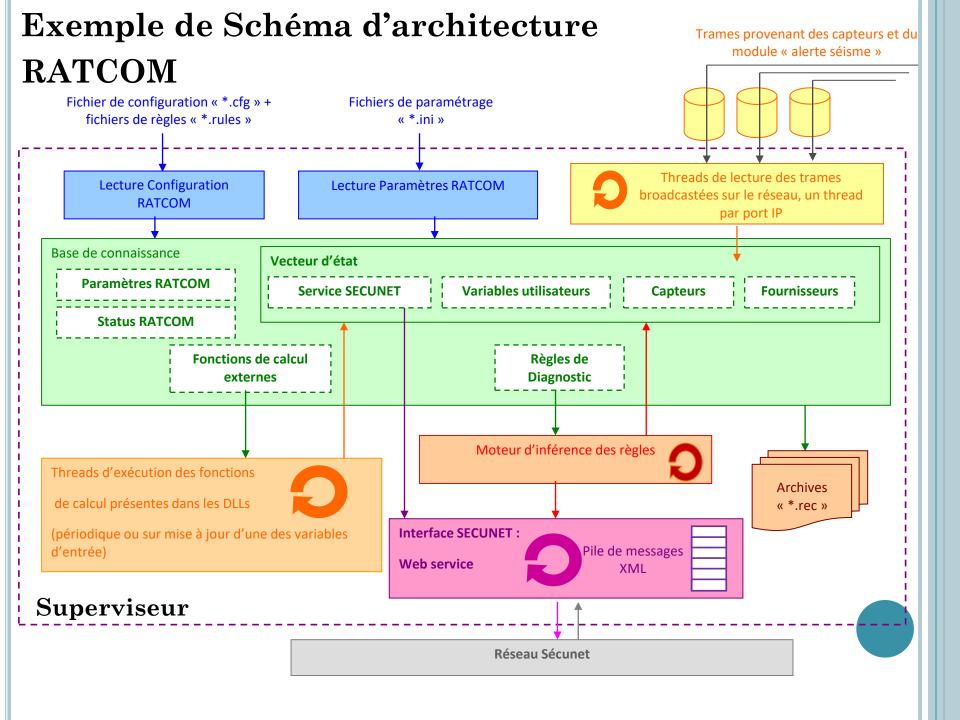
Carte bathymétrique 3D du canyon sous-marin du Var © Ifremer / Géosciences Marines

RATCOM

PROJET RATCOM







SUPERVISION LOCALE OU DÉPORTÉE

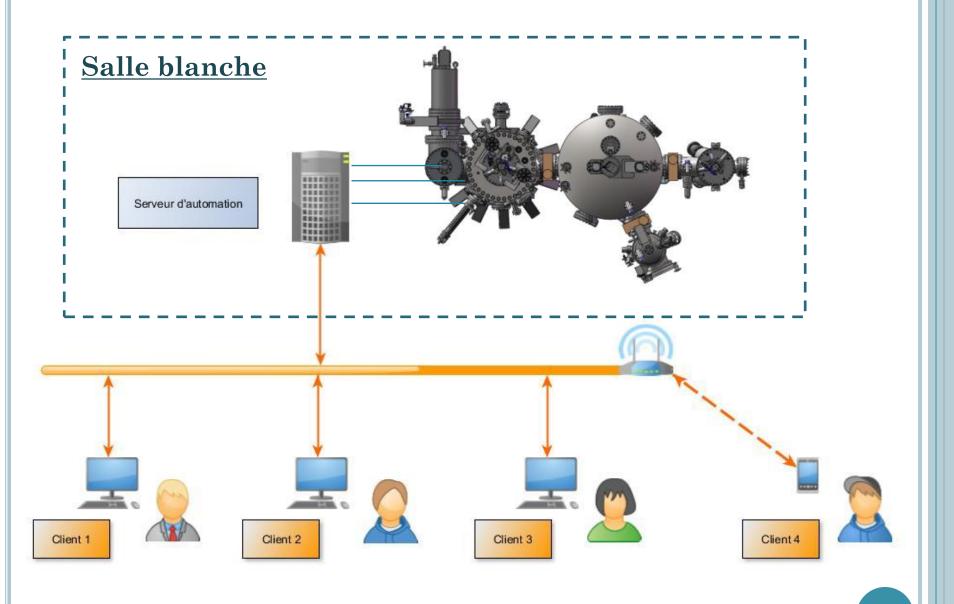
• Architecture Locale: Le système à superviser est directement relié à un ordinateur réalisant l'ensemble des tâches de l'application de supervision : acquisition, commande, enregistrement, affichage, édition...

o Architecture déportée:

- Architecture Centralisée: Client/serveur,
- Architecture Distribuée.

ARCHITECTURE CLIENT/SERVEUR

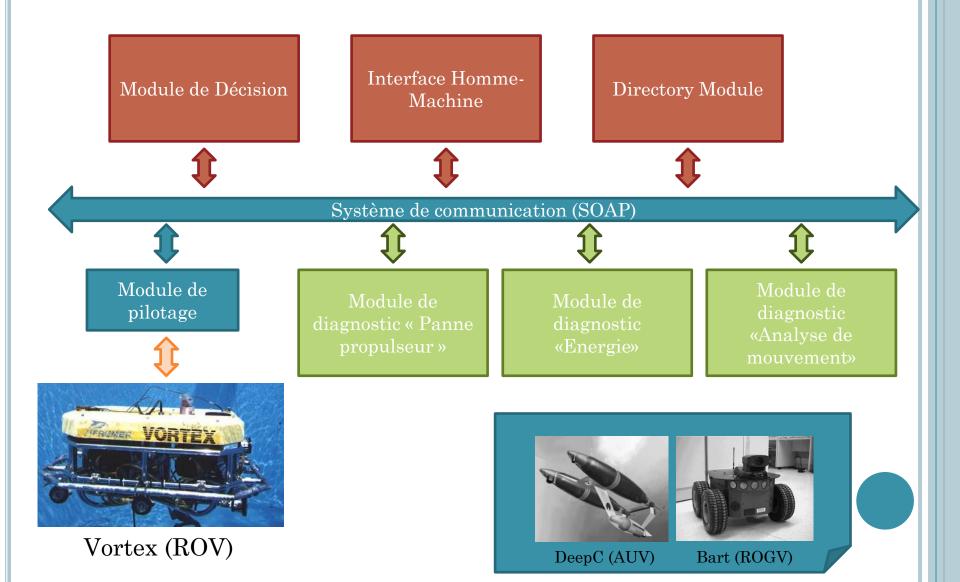
- L'ordinateur local relié à la machine devient « serveur de donnée »:
 - met à la disposition de ses clients les informations qu'il prélève localement sur la machine,
 - reçoit de ses clients réseau, des commandes qu'il interprète et exécute localement.
- Les postes clients (IHM):
 - Délocalisation de la fonction de contrôle/commande : exploitation locale et/ou déportée, fixe ou mobile, des moyens de production hors des salles de production,
 - La mutualisation des ressources : un même opérateur peut contrôler plusieurs machines simultanément,
 - Le partage des informations : les informations provenant d'une machine peuvent être diffusées simultanément vers plusieurs clients utilisateurs,
 - La connectivité : la définition d'une interface de communication réseau permet à des processus externes de s'interfacer avec les machines de production pour réaliser tout ou partie des fonctions proposées dans l'interface graphique native.



ARCHITECTURE DISTRIBUÉE

- Architecture distribuée => répartition des connaissances, des traitements et des données sur différents serveurs.
- Un système distribué = ensemble d'entités autonomes de calcul interconnectées et qui peuvent communiquer.
- Partage des ressources et des services disponibles.
- o Parallélisation des calculs.
- Chaque entité rend un service et peut être client d'un autre service.

« ADVOCATE II » : PROJET DE RECHERCHE EUROPÉEN





Les périphériques à superviser sont généralement hétérogènes.

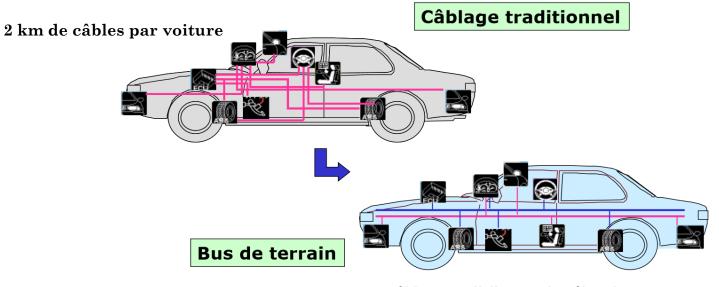
- Liaisons physiques diverses:
 - > liaison série standard RS 232, RS 422 ou RS 485,
 - > liaison USB,
 - > bus de terrain (CAN, Profibus, ...)
 - > réseau Ethernet ...
- Utilisant des protocoles différents:
 - > Modbus ASCII, RTU, NMEA 2000, ...
 - CanOpen, DeviceNet, ...
 - ➤ ModbusTCP, ...
 - > Protocole propriétaire mis au point par le fabriquant de l'équipement.

Ex. de bus de terrain: le bus CAN

CAN (Controller Area Network):

- ≻développé par Bosch et destiné à ses débuts à l'équipement automobile.
- ➤Permet de raccorder à un même câble (un bus) un grand nombre de calculateurs qui communiqueront donc à tour de rôle.

Objectif: réduire la quantité de câbles dans les véhicules.



2 câbles pour l'alimentation électrique 12 V DC 1 paire torsadée pour les transferts d'informations

COMMUNICATION PAR PORTS SÉRIE

Introduction

- Port série: l'échange de donnée se fait par ligne unique.
 - ⇒ Données sont envoyées bits par bits.
- Port de communication remplacé par l'USB sur les PCs,
- Mais reste très utilisé dans l'industrie grâce à:
 - sa robustesse (moins sensible à l'environnement que liaison //),
 - sa simplicité (nécessite peu de fils).
- Pour transporter l'information entre deux équipements, on utilise la tension ou le courant.
 - Liaison série tension: RS232, RS422, RS485,
 - Liaison série courant: boucle de courant 20mA.

TRANSMISSION SYNCHRONE/ASYNCHRONE

• Un seul fil transporte l'information ⇒ problème de synchronisation entre émetteur et récepteur (distinguer et reconnaître les séquences de bits utiles).

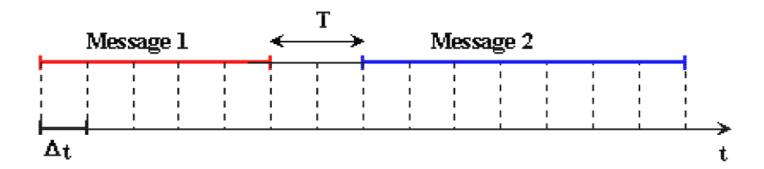
• Transmission synchrone:

- l'émetteur et le récepteur sont cadencés à la même fréquence d'horloge (nécessite un fils d'horloge),
- Le matériel le plus lent impose le rythme des communications.
- Le temps qui sépare l'envoi de 2 messages doit être un multiple de Δt (temps pour envoie d'1 bit),
- Envoi d'un ou plusieurs caractères de synchronisation puis la totalité des données.
- Efficace mais plus risqué (en cas d'erreur de transmission, retransmettre l'intégralité du flot de données).

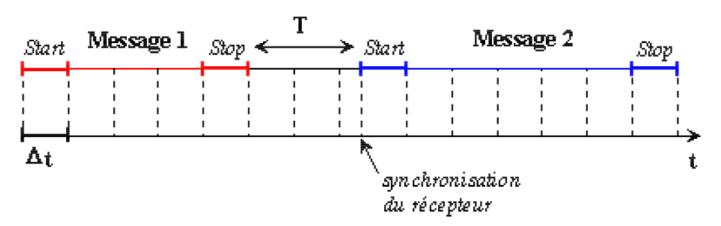
Transmission asynchrone

- Chaque caractère est émis de façon irrégulière dans le temps,
- Les éléments doivent communiquer à la même vitesse,
- Chaque caractère doit commencer par un bit de start et finir par 1 bit de stop,
- Avantage: simplicité de la méthode (le caractère est envoyé dès que la touche est appuyée) + peu cher (3 fils minimum),
- Inconvénient: plus lent.
- Question: liaison série sur PC est synchrone ou asynchrone?

• Transmission synchrone



• Transmission asynchrone

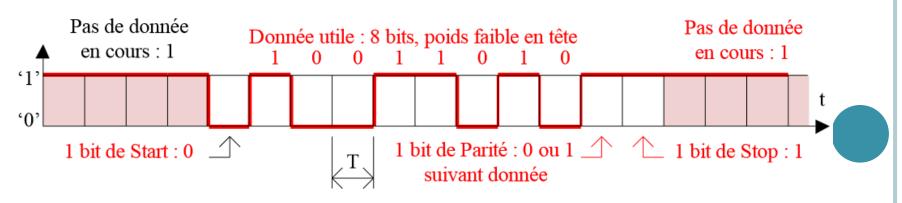


Protocole série asynchrone

- Pour chaque caractère émis:
 - Un bit de Start,
 - 5 à 8 bits de données (poids faible/poids fort),
 - Un bit de vérification de Parité (optionnel) ⇒ permet de déceler les erreurs de transmission des bits de données,
 - 1 ou 1,5 ou 2 bit de Stop.

Nb total de bits ne doit pas dépasser 11 ⇒ protocole suivant interdit: 1 bits de start, 8 bits de données, 1 bit de parité et 2 bits de stop.

La ligne au repos est au niveau logique '1'.



VITESSE DE TRANSMISSION

- Quantité d'informations qui peut être transportée en 1 seconde.
- Exprimée en bits/seconde.
- Ces vitesses sont normalisées :
 - 1200 bits/seconde,
 - 2400 bits/seconde
 - 4800 bits/seconde
 - 9600 bits/seconde
 - 19200 bits/seconde
 - 38400 bits/seconde
 - 57600 bits/seconde
 - 115200 bits/seconde

Parité paire ou impaire

- Parité paire (even)
 - Le bit ajouté à la donnée est positionné de telle façon que le nombre des états « 1 » soit <u>pair</u> sur l'ensemble donnée + bit de parité.

Exemple:

- Soit la donnée 11001011 contenant 5 états «1»:
- ⇒ le bit de parité paire est positionné à « 1 », ramenant ainsi le nombre de « 1 » à 6.

- Parité impaire (odd)
 - Le bit ajouté à la donnée est positionné de telle façon que le nombre des états « 1 » soit <u>impair</u> sur l'ensemble donné + bit de parité.

Exemple:

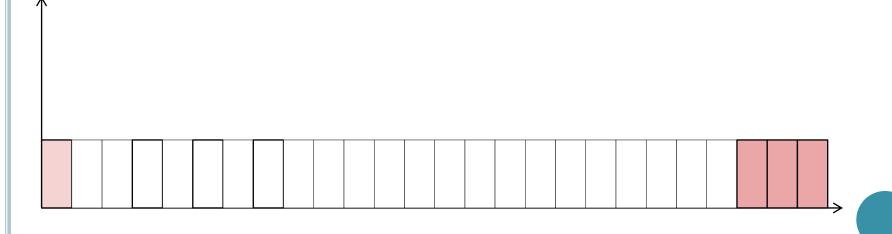
- o Soit la donnée 11001011 contenant 5 états «1»
- ⇒ le bit de parité paire est positionné à « 0 », laissant ainsi le nombre de « 1 » impair

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	
1	1	[START OF HEADING]	33	21	1	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22		66	42	В	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	е
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27		71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49		105	69	
10	A	[LINE FEED]	42	2A	*	74	4A	1	106	6A	i
11	В	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	1
13	D	[CARRIAGE RETURN]	45	2D		77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	0,000	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	0	111	6F	0
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	р
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	S
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	V
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	У
26	1A	[SUBSTITUTE]	58	3A		90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	1	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	1	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	-	93	5D	1	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F		127	7F	[DEL]

EXEMPLE DE TRANSMISSION

On souhaite envoyer le code ASCII du caractère « *W », sur 8 bits, parité paire avec 1 bit STOP.



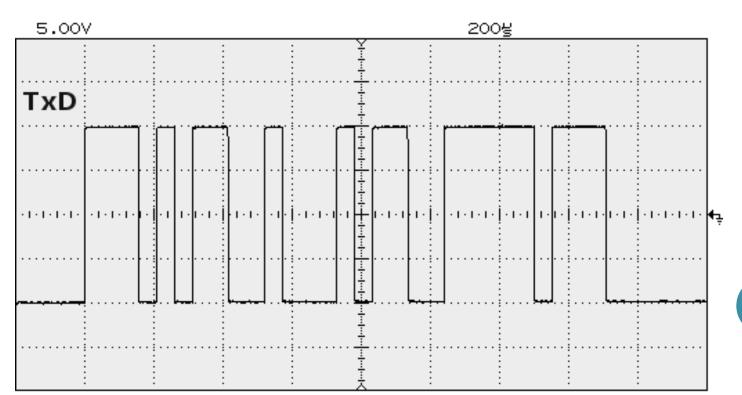
EXEMPLE DE TRANSMISSION

Les paramètres de la liaison RS-232 sont :

- 8 bits de données,
- pas de bit de parité.

Etat logique « 1 » = -5VEtat logique « 0 » = +5V

On transmet 3 octets:

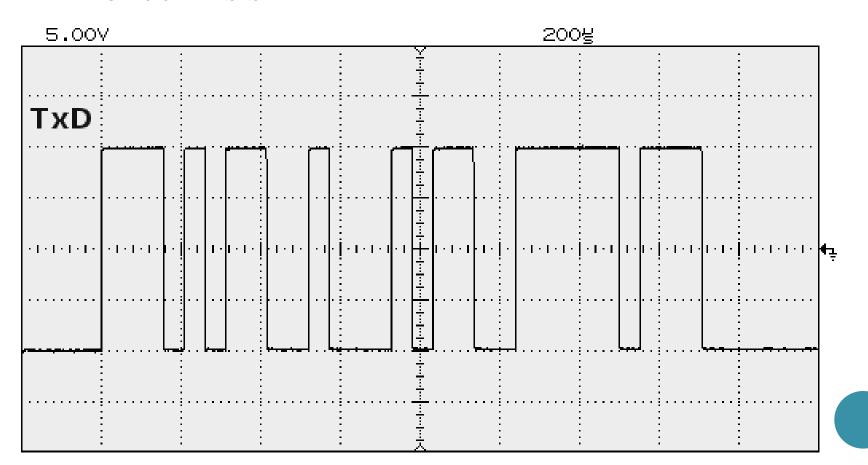


QUESTIONS:

- Quelle est la vitesse de cette transmission?
 - 4800 bits/s,
 - 9600 bits/s,
 - 19200 bits/s.
- Quelle est la durée du stop?
 - 1 bit,
 - 1,5 bit,
 - 2 bits.
- Quels sont les valeur hexadécimal des trois octets transmis:
 - 0x49 0x5B 0x08,
 - $0x94\ 0x97\ 0x10$,
 - 0x56 0x2A 0x3F,
 - 0x29 0xE9 0x08,

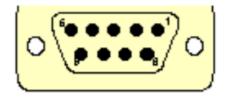
RÉPONSES:

- 4800 bits/s,
- 9600 bits/s,
- 19200 bits/s



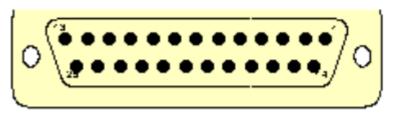
CONNECTEURS UTILISÉS

• Connecteur DB-9



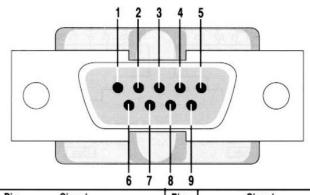
Connecteur DB 9

• Connecteur DB-25

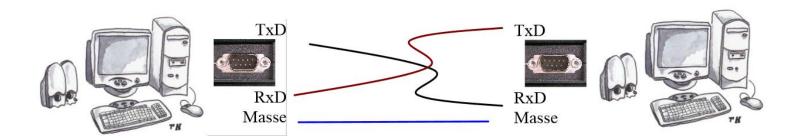


Connecteur DB 25

CONNECTEUR DB9 ET BROCHAGE



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		



CONTRÔLE DE FLUX

- Le rôle du contrôle de flux est de permettre d'éviter de perdre des informations pendant la transmission.
- o Différents types de contrôle de flux :
 - Contrôle de flux matériel,
 - Contrôle de flux logiciel.

CONTRÔLE DE FLUX MATÉRIEL

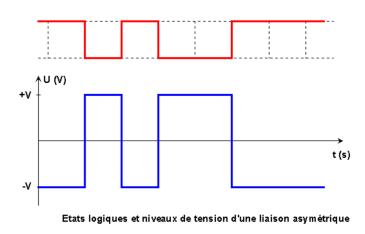
- Il est possible de contrôler le flux de données entre deux équipements par deux lignes de « handshake » (poignée de main en français). Ces lignes sont RTS (Ready To Send) et CTS (Clear To Send).
- \Rightarrow liaison série 5 fils.
- Le principe de fonctionnement est le suivant :
 - L'émetteur informe le récepteur qu'il est prêt à envoyer une donnée en agissant sur RTS et en le mettant à l'état bas.
 - Le récepteur informe l'émetteur qu'il est prêt à recevoir en mettant le signal CTS à l'état bas.
 - La transmission devient effective.
- Cette technique permet d'éviter d'envoyer des données quand le récepteur n'est pas prêt et donc permet d'éviter la perte d'information.

CONTROL DE FLUX LOGICIEL

- Il est également possible de contrôler la transmission à l'aide de deux codes ASCII « XON » et « XOFF »:
 - XOFF (code ASCII 17 ou Ctrl-Q) : le récepteur demande l'arrêt de la transmission.
 - XON (code ASCII 19 ou Ctrl-S) : dès qu'il est à nouveau prêt, le récepteur demande le départ d'une transmission.
- Permet de contrôler le flux sur une liaison série 3 fils.

ASYMÉTRIQUE

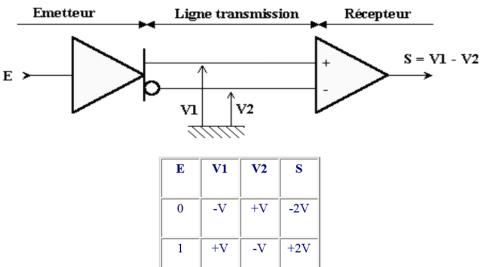
• En mode asymétrique, les états logiques sont transmis sur la ligne par deux niveaux de tension, l'un positif, l'autre négatif.



- ⇒ sensibles aux parasites induits.
- ⇒ le débit nominal maximum et la longueur maximum du câble sont de 20 kBauds et de 15 mètres.

Symétrique / différentielle (1/2)

- En mode symétrique, deux conducteurs actifs par sens de transfert.
- L'émetteur possède un amplificateur différentiel ⇒ transmet les états logiques à la double ligne de transmission.
- Le récepteur est un montage à amplificateur opérationnel ⇒ concerné que par la différence de tension entre les deux fils de ligne.



Symétrique / différentielle (2/2)

• La liaison symétrique ou différentielle permet de transmettre des données sur de grandes distances à des vitesses élevées.

E	V1	V2	Parasite	S = V1 - V2
0	-V	+V	vp	(-V+vp) - (+V+vp) = -2V
1	+V	-V	vp	(+V+vp) - (-V+vp) = +2V

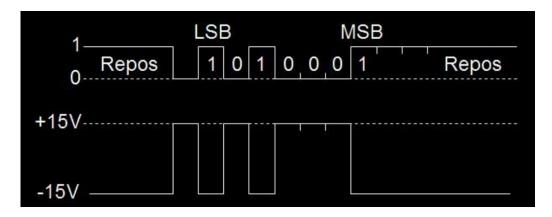
- ⇒ peu sensible aux parasites induits (parasites affectent les deux fils de la ligne et sont inhibés par l'entrée différentielle),
- ⇒ le débit nominal =10 M Bauds et longueur de 1200 mètres.

NORME RS 232

- Liaison série tension point à point asymétrique
- La norme RS232 prévoit la conversion logique électrique suivante :

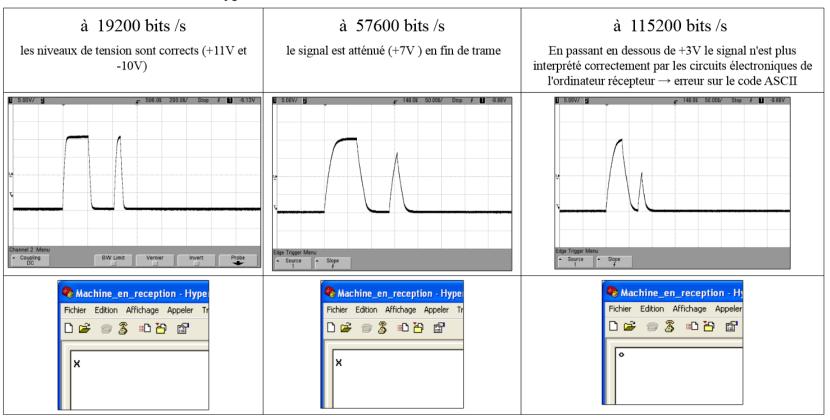
	Emetteur	Récepteur
Etat Logique « 1 »	-5 / -15 V	-3 / -25V
Etat Logique « 0 »	+5 / +15 V	+3 / +25 V

• Correspondance trame électrique / trame logique



LIMITE DE FONCTIONNEMENT

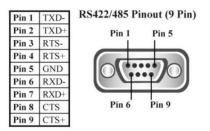
Le signal a été relevé sur la broche Rx du connecteur SUB-D9 de l'ordinateur récepteur lors de l'envoi du caractère 'x' sur l'ordinateur émetteur avec hyperterminal.



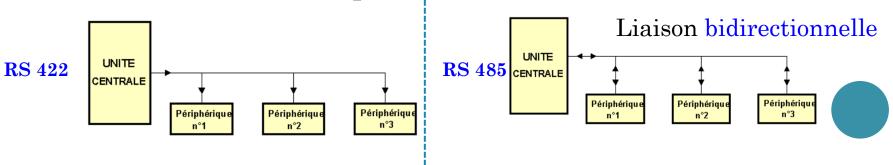
NORME RS 422/485

• Adaptation en tension différentielle des signaux afin d'être transmis sur des distances supérieures (1200m),

RS422/485



• Transmission multipoint.



DIFFÉRENCE RS232/422/485

Paramètre	RS 232 D	RS 422 A	RS 485
Mode fonctionnement	Asymétrique	Symétrique différentiel	Symétrique différentiel
Nombre émetteurs	1	1	32
Nombre récepteurs	1	10	32
Longueur maximum (m)	15	1 200	1 200
Débit maximum (bauds)	20 K	10 M	10 M

CODAGE C SOUS WINDOWS EN UTILISANT L'API WIN32 (1/2)

- o Ouverture du port série
 - CreateFile(

LPCTSTR lpFileName,
DWORD dwDesiredAccess,
DWORD dwShareMode,
LPSECURITY_ATTRIBUTES lpSecurityAttributes,
DWORD dwCreationDisposition,
DWORD dwFlagsAndAttributes,
HANDLE hTemplateFile);

- Valeur de retour:
- ⇒ Succès = Handle: identifiant du port,
- ⇒ Echec = INVALID_HANDLE_VALUE.
- Fermeture du port
 - CloseHandle(HANDLE hObject);

Codage C (2/3)

- Paramètrage du port
 - GetCommState,
 - SetCommState,
 - GetCommTimeouts,
 - SetCommTimeouts,

Codage C (3/3)

- Ecriture sur le port série
 - WriteFile
- Lecture sur le port série
 - ReadFile

https://msdn.microsoft.com/en-us/library/ff802693.aspx

CODAGE C

- Fonctions utiles
- Initialise la mémoire:
 void * memset (void * ptr, int value, size_t num);
- Renvoie la taille d'une C string: size_t strlen (const char * str);
- Récupère une C string depuis un flux: char * fgets (char * str, int num, FILE * stream);
- Split une C string suivant un séparateur donné:
 char * strtok (char * str, const char * delimiters);
- Compare deux C Strings: int strcmp (const char * str1, const char * str2);
- Copie une C string dans une autre: char * strcpy (char * destination, const char * source);

QUELQUES NOTIONS SUR LE MULTITHREADING

DÉFINITION

- Tâche = exécution d'une séquence de code sur un équipement,
- Multitâches = consiste pour un système à pouvoir dérouler plusieurs tâches simultanément.
- Les systèmes d'exploitation actuels des PC sont multitâches:
 - Ordonnanceur: répartit les différentes tâches sur les ressources matérielles disponibles \Rightarrow découpe le temps processeur et le distribue sur chaque tâche en basculant les contextes d'exécution.



EXEMPLE:

- Un processeur a une fréquence de 1.8 GHz: $1/(1.8 \times 10^9 \text{ Hz}) = 0.55 \times 10^{-9} \text{ s} = 0.55 \text{ ns}$
- \Rightarrow cycle de base de 0,55 ns

Si un quantum de temps = 1ms

- \Rightarrow 10⁵ / 0.55 = 1 818 181 instructions en 1 ms
- \Rightarrow Si 2 threads chaque seconde => 500ms par thread = 500 * 1 800 000 = 900 000 000 instructions par thread en un seconde

MULTIPROCESSUS

- Un système multiprocessus orchestre plusieurs applications concurrentes sur un même équipement.
- Chaque processus est indépendant ⇒ zone de mémoire physique distincte,
- Un processus ne peux pas corrompre la zone mémoire d'un autre processus,
- Deux processus ne peuvent interagir qu'à travers une API,
- La durée de vie d'un processus indépendante des autres processus.

MULTITHREAD

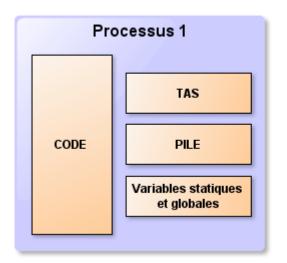
- Thread (processus léger) : unité d'exécution rattachée à un processus et chargée d'exécuter une partie du processus,
- A la création d'un processus : 1 seul flot d'exécution = thread principal,
- Multithreading: exécution de plusieurs threads simultanément dans un même processus:
 - Partage des ressources du processus par tous ses threads,
 - Bascule rapide d'un thread à l'autre,
 - Fin du processus entraîne la fin de tous ses threads.

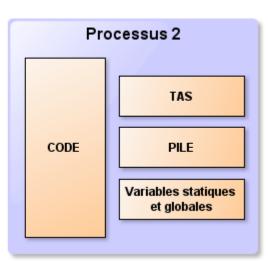


Accès concurrent à la mémoire, Deadlock.

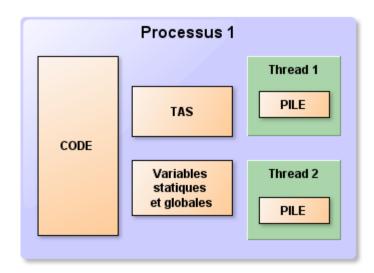
Multiprocessus / multithreads

<u>► Multiprocessus</u>

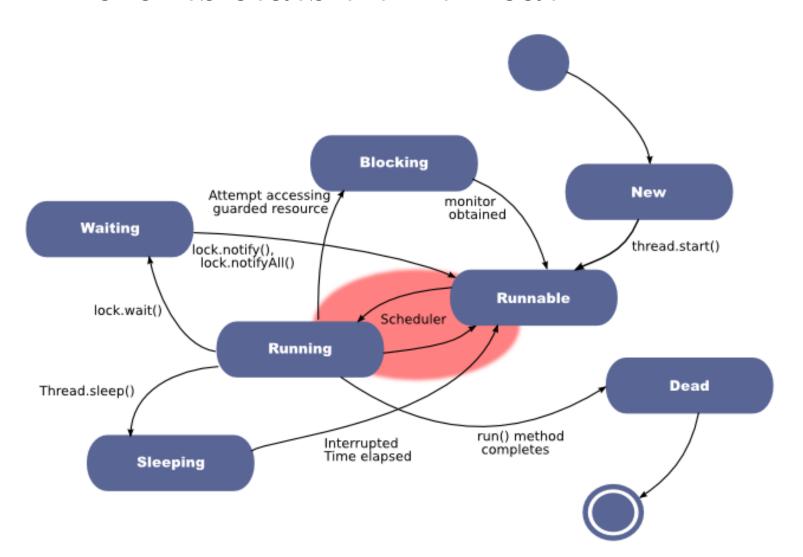




<u>► Multithreads</u>

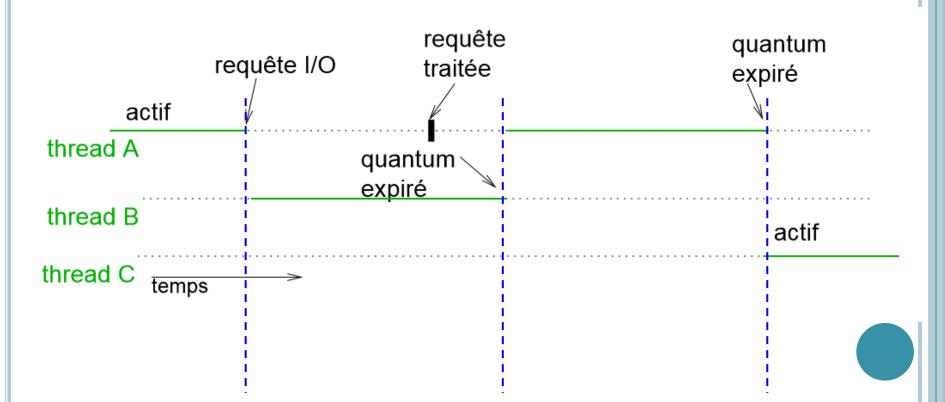


Différents états d'un thread



- Un thread en cours d'exécution passe à l'état suspendu:
 - Implicitement lorsqu'il attend qu'une ressource se libère et lors d'une opération d'E/S (ex: lecture sur un port série),
 - Explicitement lors de l'appel à:
 - Sleep (suspension pendant un temps exprimé en milli secondes)
 - Wait (suspension jusqu'à ce qu'un autre thread le notifie).

- L'ordonnanceur donne la main à un autre thread quand:
 - Le thread courant a épuisé le quantum temps qui lui était alloué,
 - Le thread courant est suspendu.



CRÉATION D'UN THREAD SOUS WINDOWS

```
#include <windows.h>
DWORD WINAPI ThreadFunc (void* data)
    DWORD* param = (DWORD*) data;
    return 0;
int main()
       DWORD dwThreadId = -1;
DWORD dwThreadParam = 1;
       \label{eq:handleThread} \begin{split} & HANDLE\ HandleThread = \textbf{CreateThread}(NULL,\ 0,\ \textbf{ThreadFunc},\ ,\ \&dwThreadParam,\ 0,\ \&dwThreadId\ ); \end{split}
      if (HandleThread)
      return 0;
```

COMMENT TERMINER UN THREAD

- Un thread s'exécute jusqu'à ce qu'un des événements suivants surviennent:
 - Arrêt normal du thread ⇒ le thread a terminé son traitement et appelle:
 - la fonction « ExitThread »,

ou

- o l'instruction « return »,
- Arrêt forcé du thread: un autre thread appelle la fonction « TerminateThread »:

```
BOOL WINAPI TerminateThread(
```

```
HANDLE hThread, \Rightarrow handle du thread à détruire DWORD dwExitCode);
```