

# Le port série et bus USB



# Examen final — logistique

- mardi 26 avril de 14h30 à 16h20 (même heure que le cours)
- Nous diviserons le groupe en deux:
  - Andande à Gaudy: PLT-2341
  - Gauvin à Weber-Boisvert: PLT-2751

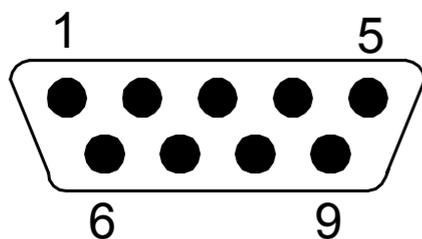
# Série vs parallèle

- La majorité des bus à l'intérieur des ordinateurs modernes adoptent des protocoles de type série. Pourquoi?
  - À très haute fréquence, les délais de transmission d'un signal électrique dans un fil court peuvent devenir considérable par rapport à la période d'un bit.
  - Lorsque plusieurs traces (fils électriques sur des PCBs) sont en parallèle, elles ne peuvent pas avoir la même longueur pour des raisons purement mécaniques.
  - Des signaux électriques qui partent en même temps d'un transmetteur n'atteignent pas en même temps le receveur si les fils qui propagent les signaux n'ont pas la même longueur. À haute fréquence, il est possible que des bits d'un fil arrivent décalés par rapport au bits des autres fils. De ce fait, un bus parallèle est peu envisageable pour des fréquences supérieures à 1 GHz.

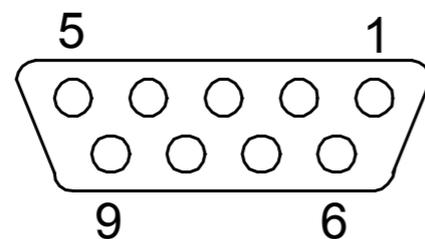
# Matériel et Connecteur

- Dans le protocole RS-232, il existe deux types d'appareils:
  - les Data Terminal Equipment (DTE), équivalent aux ordinateurs;
  - les Data Communication Equipment (DCE) qui communiquent des données à l'ordinateur.
- Trois lignes, en vert, servent pour communiquer: RD, TD et la masse du signal (ground).
- Les lignes RD et TD contiennent les signaux transmis du DTE au DCE et ceux du DCE au DTE, respectivement. Il est possible de faire de la communication par le port série avec ces trois lignes seulement!
- Les autres lignes servent au contrôle de flux de données entre le DTE et le DCE. Elles indiquent que le DTE ou le DCE est prêt à recevoir ou à émettre des données. Les lignes en bleu (DTR, DSR, RTS et CTS) sont couramment utilisées.

DTE, mâle,  
comme sur PC



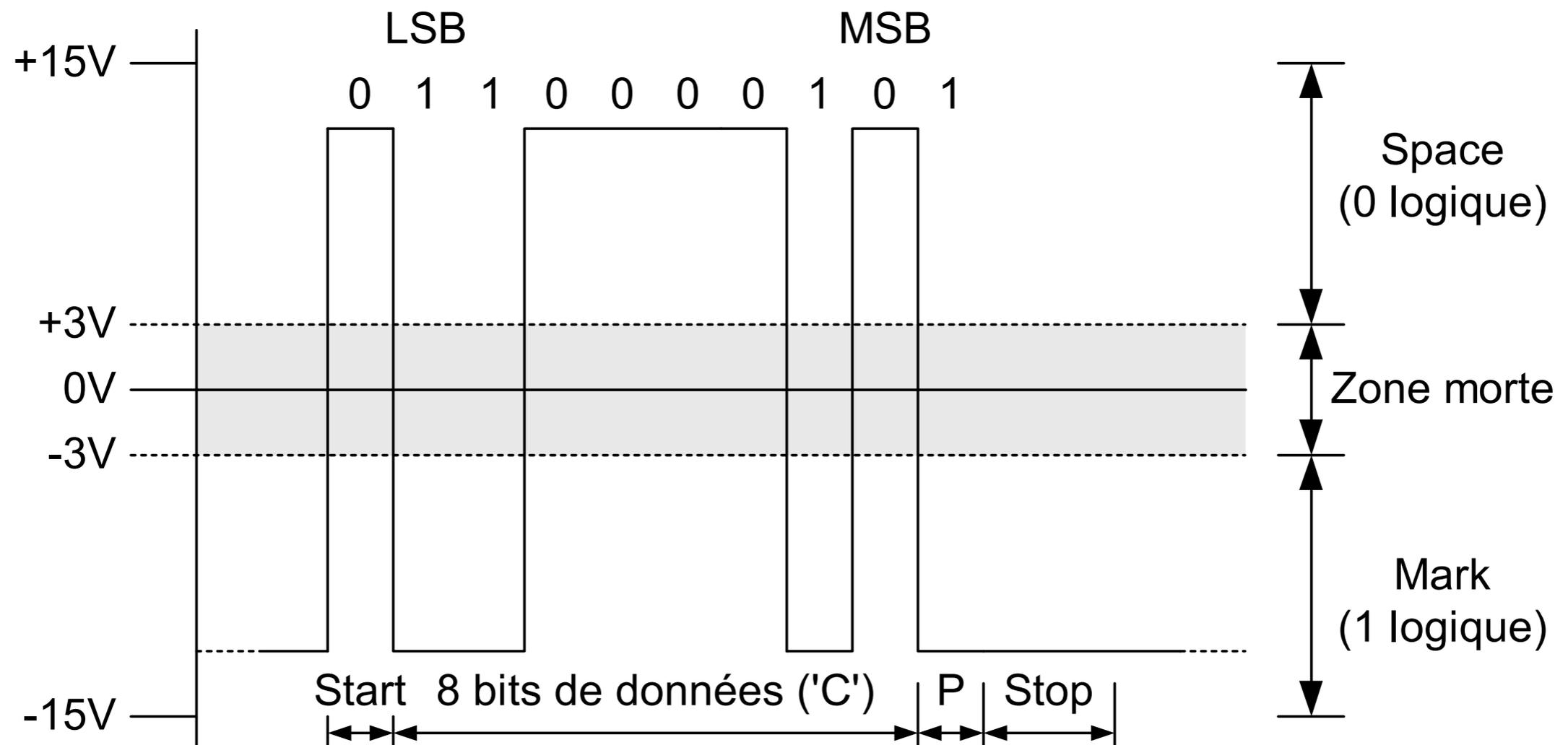
DCE, femelle,  
comme sur modem



Nom	9-pin DTE	25-pin DTE	Contrôle
Carrier Detect (DCD)	1	8	DCE
Received Data (RD)	2	3	DCE
Transmitted Data (TD)	3	2	DTE
Data Terminal Ready (DTR)	4	20	DTE
Signal Ground	5	7	DCE
Data Set Ready (DSR)	6	6	DCE
Request To Send (RTS)	7	4	DTE
Clear To Send (CTS)	8	5	DCE
Ring Indicator (RI)	9	22	DCE

# Signaux

- Le signal transmis sur les pins RD et TD va de +15V à -15V:
  - entre +3V et +15V, il est interprété comme un 0 logique;
  - entre -3V et 15V, il est interprété comme un 1 logique;
  - entre -3V et 3V, le signal est considéré invalide.
- Des bits de départs et de fins servent à délimiter les bits de données.
- Il peut y avoir un bit de parité servant à détecter les erreurs. Ce bit est décrit plus loin.



# Protocole de communication

- Le principal protocole de communication utilisé sur le port série est le RS-232. Cette spécification détermine:
  - Les caractéristiques des signaux électriques transmis (voltages, vitesse, transitions, longueurs de fils, etc.).
  - Le connecteur utilisé.
  - Les fonctions de chaque partie du port.

# Paramètres du port série

- **Baud Rate:** La fréquence des bits transmis sur le port série.
  - Les fréquences disponibles sont pré-établies: 300bps, 600bps, 1200bps, ... 19200bps, 38400bps, etc. Défaut = 9600bps
- **Parité:** Le bit de parité sert à vérifier s'il y a eu des erreurs dans l'octet transmis (on ne peut cependant pas *corriger* l'erreur)
  - En transmission, on compte le nombre de fois "1" apparaît dans l'octet transmis, et on ajuste le bit de parité:
    - En parité "paire", le bit de parité est mis à 1 pour que le nombre total de "1" soit pair. Vice versa en parité "impaire".
  - En réception, on compte le nombre de 1, puis on vérifie si le bit de parité est bon.
- **Bits d'arrêt:** Nombre de bit d'arrêt (1) qui suivent le byte transmis.
  - Défaut = 1.
- **Nombre de bits par octet:** Nombre de bit transmis par octet.
  - Peut être 5, 6, 7, 8 et 9 (très rare!). Défaut = 8.

# Exercice

- Transmettre le caractère 'B' en ASCII (0x42) sur le port série.
- On emploie la configuration suivante:
  - mot de 7 bits (LSB en premier);
  - parité paire;
  - 1 bit d'arrêt.

- Questions:

- Quelle séquence de bits sera-t-elle transmise? Dessinez le signal correspondant.

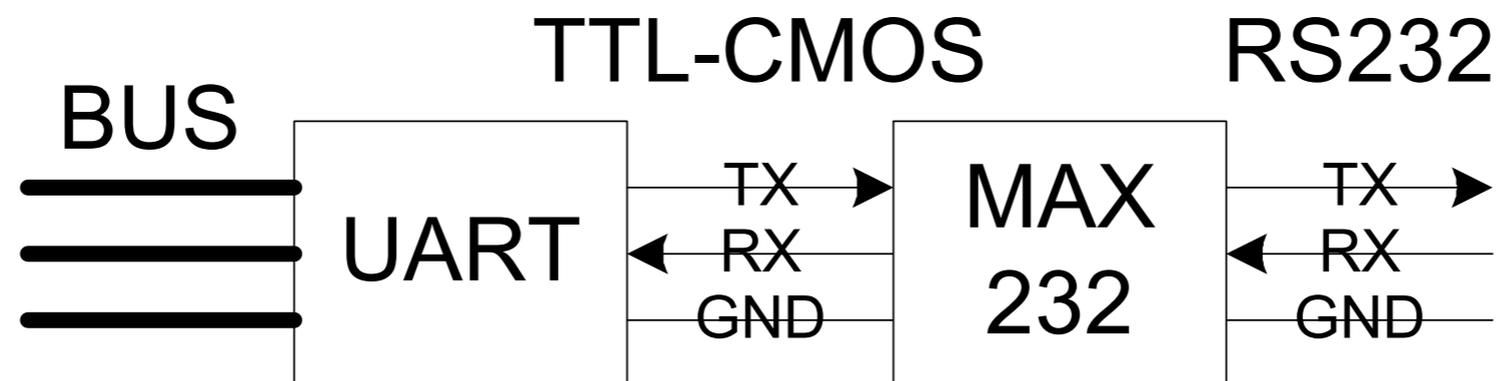
- La séquence sera:

start	caractère (0x42, LSB en premier)							parité	stop
0	0	1	0	0	0	0	1	0	1

- Si la vitesse est de 100 bps, combien de temps prendra la transmission de cette séquence de bits?
  - $10 \text{ bits} / 100 \text{ bits/s} = .1\text{s}$

# UART et RS232

- UART signifie: “**U**niversal **A**synchronous **R**eceiver **T**ransmitter”
- Un UART est un module d’E/S qui convertit les signaux parallèles d’un bus en signaux en série.
- Le signal série sortant d’un UART est comme le signal RS232 (Start bit, Octet de données, Parité, Stop Bit), mais avec des niveaux de tension TTL ou CMOS (-3V—3V).
- Le UART utilise un registre à décalage pour convertir les signaux parallèles en signaux série.

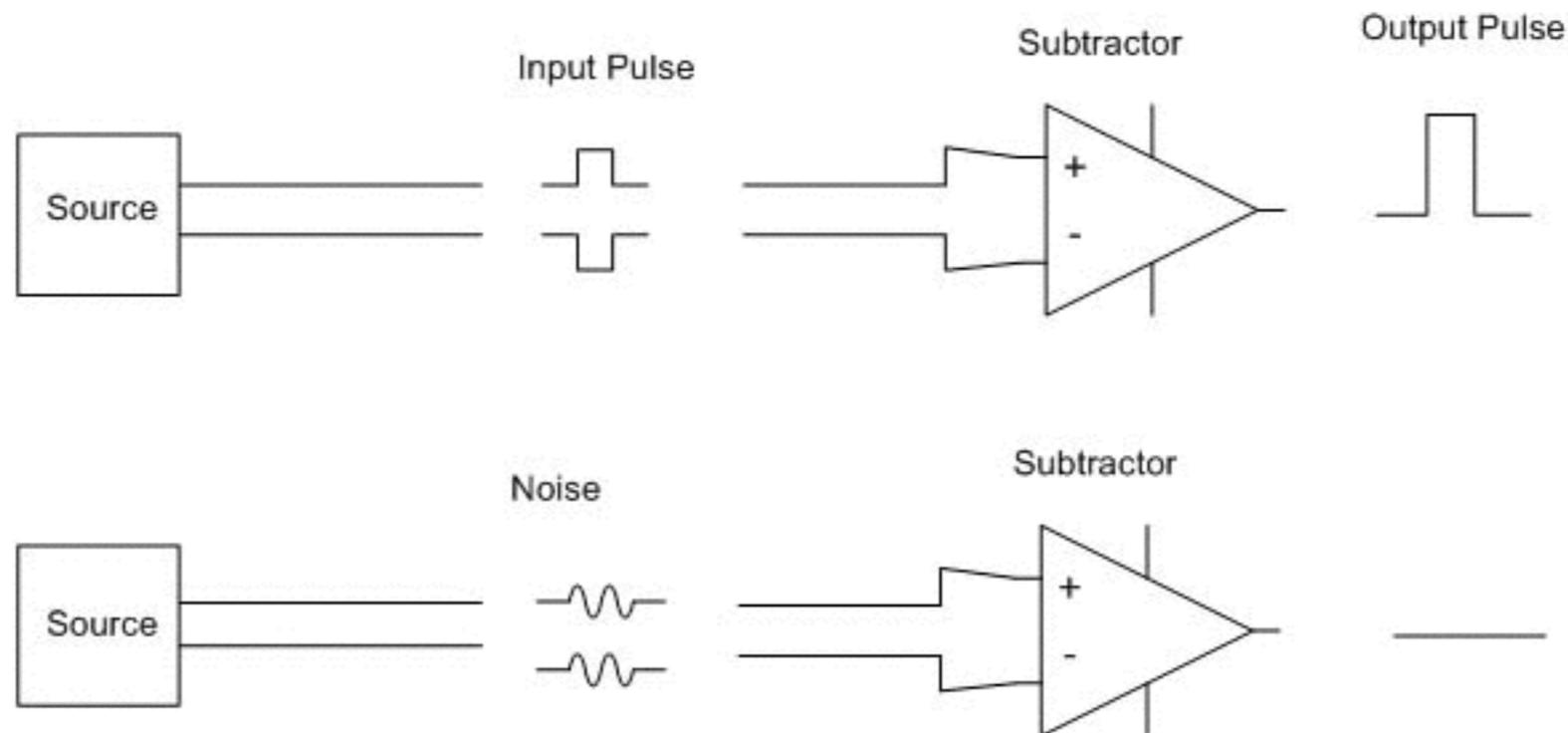


# Problème?

- Qu'arrive-t-il s'il y a du bruit sur la ligne de communication?

# Mode différentiel

- Les bits transmis sont encodés en mode différentiel. Pourquoi?
  - La différence de tension entre deux signaux propagés sur deux lignes différentes détermine la valeur d'un bit transmis. Des symboles différents sont transmis si la différence est positive ou négative.
  - Le bruit commun sur les deux lignes propageant le signal est éliminé lorsque la différence est effectuée. Très robuste.
  - Lorsque la différence est nulle, le bit est invalide ou une autre information peut être transmise.



# Aspects importants (port série)

- On « emballe » les données avec:
  - bit de départ, bit d'arrêt, bit de parité
- (en RS-422) On utilise deux lignes pour transmettre les données en mode différentiel pour être plus robuste au bruit sur la ligne

# Universal Serial Bus (USB)





# *Pimp My Ride*

# *Pimp My Serial Port*



# Pourquoi le USB?

Circa 1997



Circa maintenant

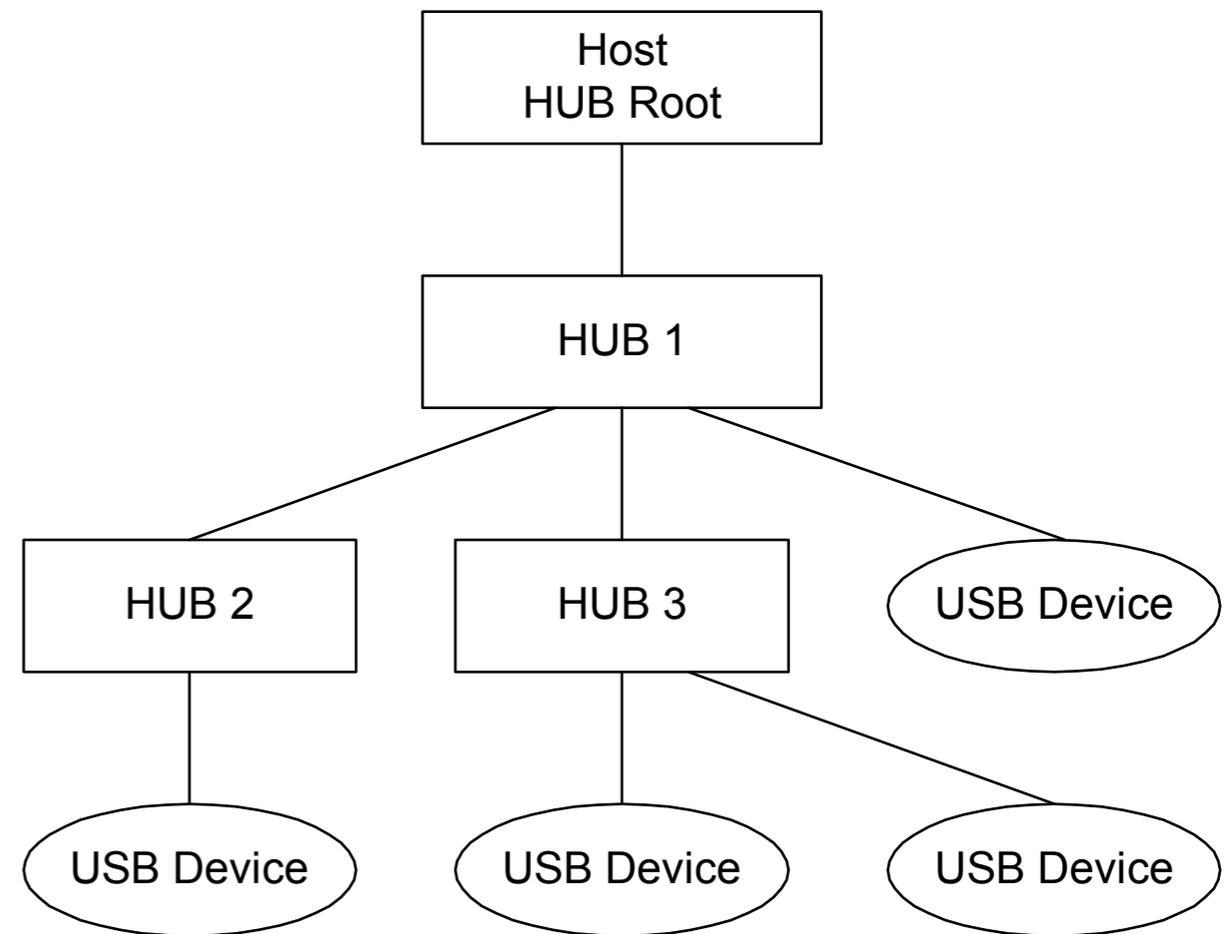


# Qu'est-ce qui utilise USB?

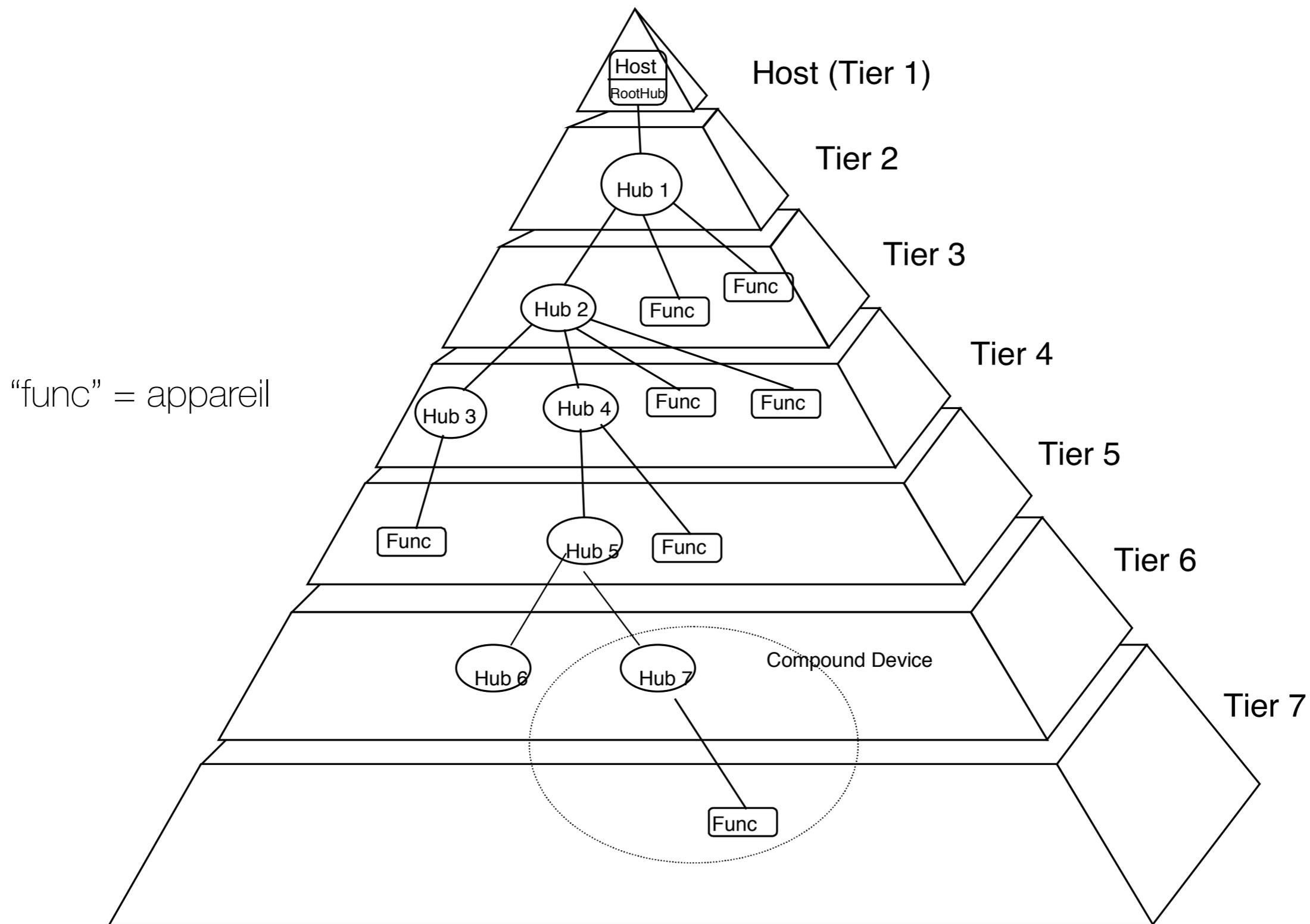
[https://youtu.be/1rlwBz\\_1\\_1w?t=5m6s](https://youtu.be/1rlwBz_1_1w?t=5m6s)

# Topologie d'un réseau USB

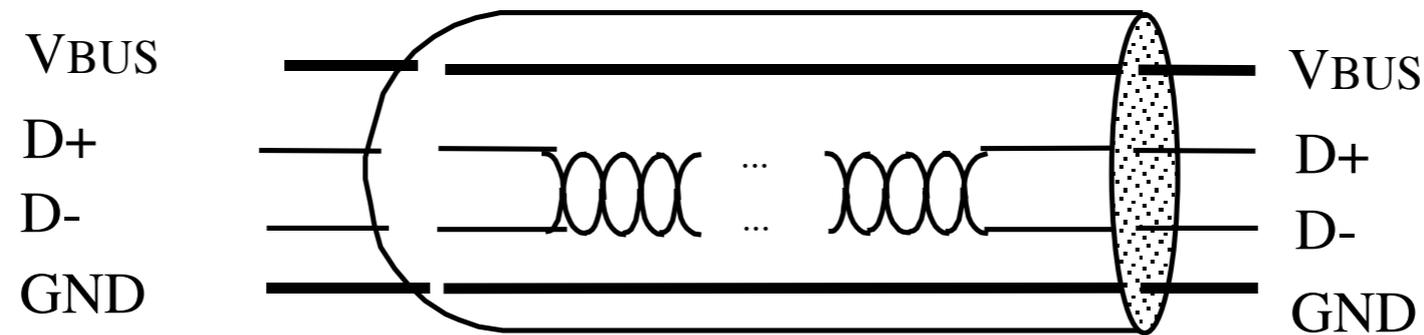
- Un réseau USB a une topologie en étoile.
- Le port USB est contrôlé entièrement par un contrôleur unique appelé hôte ("host"). Souvent l'ordinateur, il initie toutes les communications, et est le maître absolu du bus.
- Les "hubs" permettent de relier plusieurs appareils à un seul port USB.
  - Le rôle principal des hubs est de transférer les données de l'hôte aux périphériques.
  - Chaque hub contrôle ses ports afin de savoir si un appareil s'y connecte
  - Il peut y avoir 5 niveaux de hub en plus du hub racine.
- Il y a 127 appareils maximum dans un réseau USB. Chaque appareil a son adresse.



# Topologie d'un réseau USB



# Matériel — fils

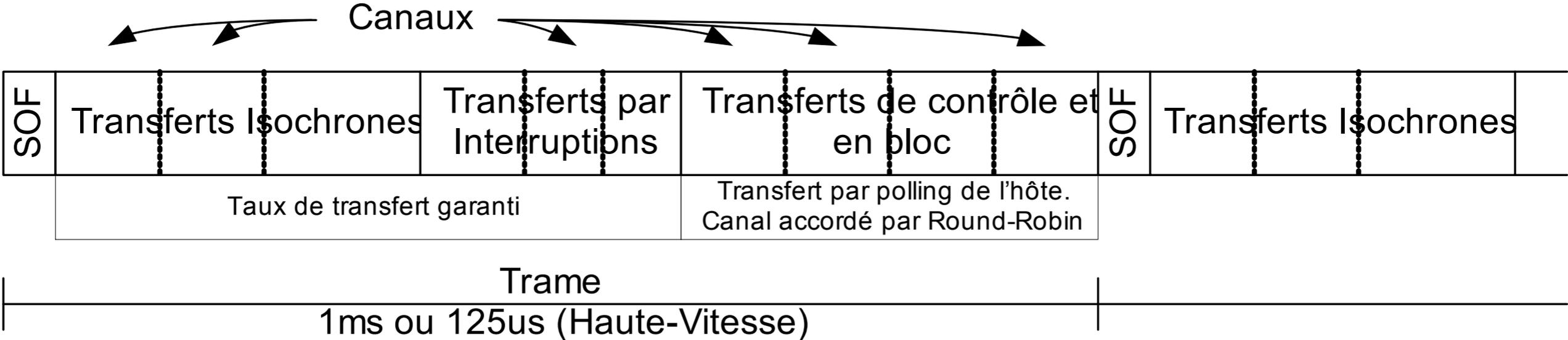


- Le câble USB est constitué de 4 fils:
  - Vbus** est l'alimentation 5Vdc (entre 4.75V et 5.25V)  
(peut alimenter les appareils branchés sur le bus!)
  - D-** et **D+** servent au transport des données.
  - GND** est la référence électrique

# Types de transfert

- Le USB supporte 4 types de transfert:
  - **de contrôle**: sert à la configuration et à la commande d'un appareil. Il est effectué à partir de la terminaison 0.
  - **isochrone**: est un mode de transfert pour lequel les données sont transmises à vitesse constante, et garantie. Idéal pour les flux de données ("streaming").
  - **par interruption**: est utilisé par les appareils ayant peu de données à transmettre, mais ayant des données qui doivent être transmises rapidement (exemple: clavier ou souris). Ce ne sont pas de "vraies" interruptions: elles sont détectées par interrogation successives ("polling") de provenance l'hôte. La fréquence de ces interrogations est donnée par les descripteurs de l'appareil.
  - **par bloc**: permet de transférer des volumes importants de données lorsqu'il n'y a pas de contraintes temporelles (exemple: imprimante).

# Trames USB



# Transactions

- Les transferts de données sont faits en mode half-duplex (*USB 3.0*: full-duplex)
- C'est l'hôte qui initie *tous* les transferts de données.
- La plupart des transactions nécessite l'envoi de 3 paquets:
  1. "Token packet": l'hôte envoie un paquet décrivant le type, et la direction de la transaction.
    - Le paquet contient:
      - L'adresse de l'appareil USB
      - Le numéro de terminaison sur cet appareil
  2. "Data packet": L'appareil USB correspondant s'active en fonction de l'adresse reçue. Un deuxième paquet est envoyé:
    - Le paquet contient les données correspondant à la transaction demandée,
    - Le paquet est envoyé selon la direction de la transaction (hôte vers appareil, ou appareil vers hôte),
  3. "Handshake packet": l'appareil de destination indique si la transaction a été complétée avec succès ou non.

# Protocole de bus



- Exemples de transactions:
  - Transfert de données, du périphérique vers l'hôte



- Transfert de données, de l'hôte vers le périphérique



- Configuration, de l'hôte vers le périphérique



↑  
"Token"

↑  
"Data"

↑  
"Handshake"