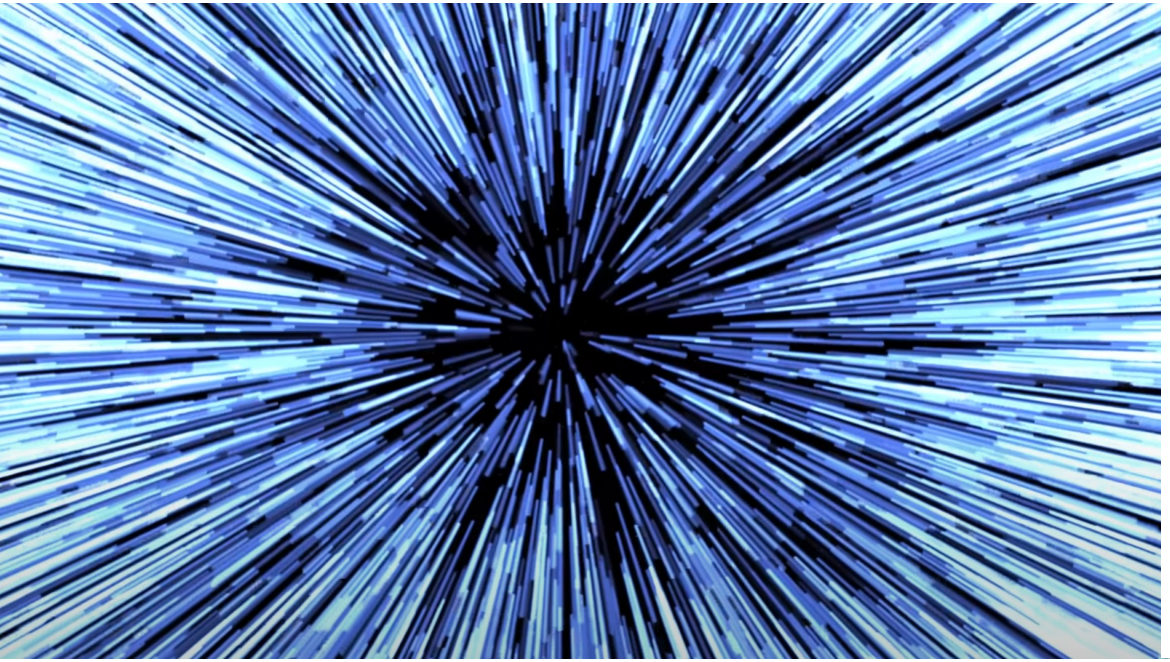
Bus de communication

Nano-ordinateurs

François Roland



Vitesse de transmission

Vitesse de la lumière dans le vide : $c \approx 3 \times 10^8 \frac{\text{m}}{\text{s}}$

Un signal électrique met un certain temps avant d'atteindre sa destination.

Problématique de la synchronisation

- Les processeurs exécutent des instructions à vitesse très élevée.
- Les composants doivent échanger des données.
- Plusieurs composants peuvent vouloir accéder à une même ressource en même temps.

Comment savoir quand lire ou écrire des données ?

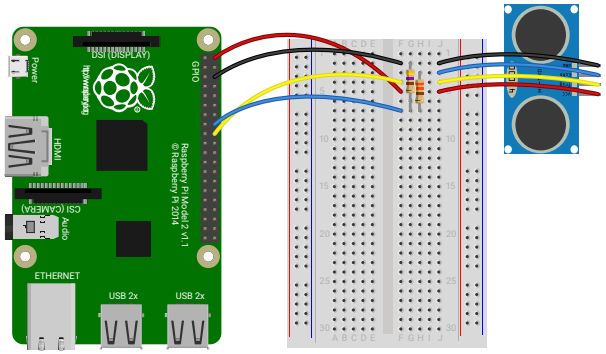
Conséquences d'une absence de synchronisation

- Données corrompues
- Comportement aléatoire
- Blocage ou crash du système

Démonstration

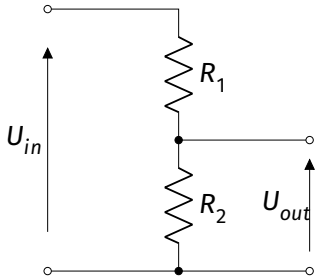
Capteur de distance à ultrason HC-SR04

1. Connecter le RPi et le capteur.
2. Exécuter le programme.
3. Tester.
4. Observer avec un analyseur logique.



Diviseur de tension

Le capteur fonctionne avec une tension U = 5 V.
Le RPi ne peut pas accepter U > 3,3 V.



$$U_{out} = U_{in} \times \frac{R_2}{R_1 + R_2}$$

U_{in} = 5 V
R_1 ≈ 10 kΩ

U_{out} = 3,3 V
R_2 ≈ 20 kΩ

Robustesse du code

Nettoyage avant fermeture

Le programme ne se termine jamais, sauf

- ctrl + c
- Extinction ou redémarrage de l'ordinateur

Il faut utiliser un mécanisme pour nettoyer les ressources avant de quitter.

```
def sig_handler(_1, _2):
    """Nettoie avant de quitter"""
    print("Exiting")
    GPIO.cleanup()
    sys.exit(0)
…
signal(SIGINT, sig_handler)
```


Robustesse du code

Timeout

Limiter les temps d'attente pour éviter les blocages.

```
response = GPIO.wait_for_edge(
    ECHO_PIN, GPIO.RISING, timeout=MEASURE_TIMEOUT)
start = perf_counter_ns()
if response is None:
    print("Timeout en attendant le flanc MONTANT")
    return None
```


Signal d'horloge

- Permet la synchronisation des composants (CPU, RAM, périphériques).
- Cadence les instructions.
- Coordonne le fonctionnement.

Bascule D

Définition

Circuit qui capture une valeur uniqument à l'arrivée d'un front montant de l'horloge.

Bascule D

Simulation

https://www.falstad.com/circuit/e-edgedff.html

Génération du signal d'horloge

- Circuit résonnant RC ou LC
- Oscillateur à quartz
- Boucles à verrouillage de phase (PLL)

PLL

Diviseur de fréquence

https://www.falstad.com/circuit/e-divideby2.html

PLL

Multiplicateur de fréquence

`https://www.falstad.com/circuit/circuitjs.html`

Circuits > Boucles à vérouillage de phase > Duplicateur de Fréquence

Bus

Définition

Un bus est un ensemble de lignes de communication permettant l'échange de données entre différents composants d'un système informatique. Il sert de canal de transmission pour les instructions, les adresses et les données entre le processeur, la mémoire et les périphériques.

Bus parallèle

Définition

Un bus parallèle est un bus qui transmet plusieurs bits simultanément sur plusieurs lignes de communication.

Bus parallèles les plus courants : ceux qui relient le CPU à la RAM.

Anciennement utilisés pour les périphériques (AGP, PCI).

Largeur d'un bus

Définition

La largeur d'un bus parallèle est le nombre de bits transmis simultanément. Il correspond donc au nombre de lignes de communication.

On trouve couramment des bus de 8, 16, 32 ou 64 bits, mais toutes les largeurs sont possibles.

Pour gérer la communication entre CPU et RAM, il faut plusieurs bus parallèles : données, adresses, commandes.

Bus série

Définition

Un bus série est un bus qui transmet les données bit par bit sur une seule ligne de communication.

Bus séries les plus courants : USB, SATA, PCIe, i^2c, SPI, UART.

Avantages et inconvénients

- En théorie, bus parallèle plus rapide que bus série
- En pratique, bus série plus rapide que bus parallèle
- Bus série = moins de lignes de communication
- Bus série plus efficace sur longues distances

Synchrone vs asynchrone

- Bus synchrone : horloge commune pour tous les composants
- Bus asynchrone : signaux de validation

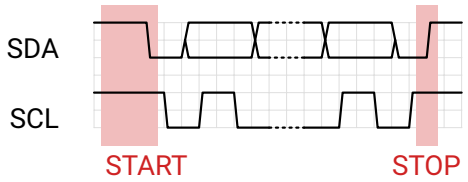
Protocole i^2c

- Bus série synchrone bidirectionnel
- 2 fils : données (SDA) et horloge (SCL)
- Contrôlé par un composant « maître »
- Peut gérer plusieurs composants « esclaves »
- Vitesse de transfert entre 100 kbps et 5 Mbps

NXP Semiconductors. Extrait d'une datasheet.

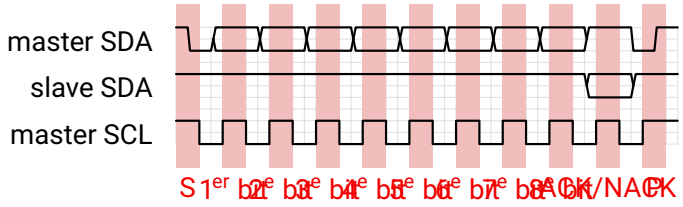
Protocole i^2c

Début et fin



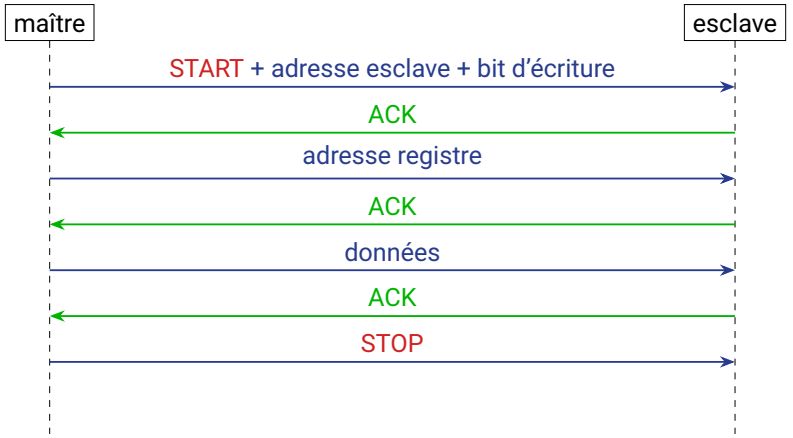
Protocole i^2c

Accusé de réception



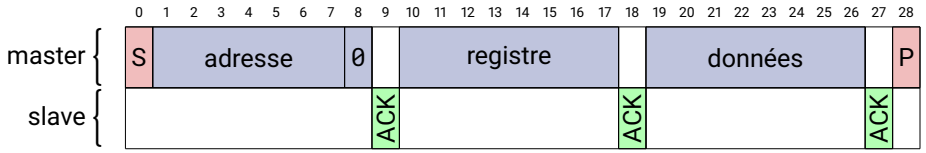
Protocole i^2c

Écriture d'un octet



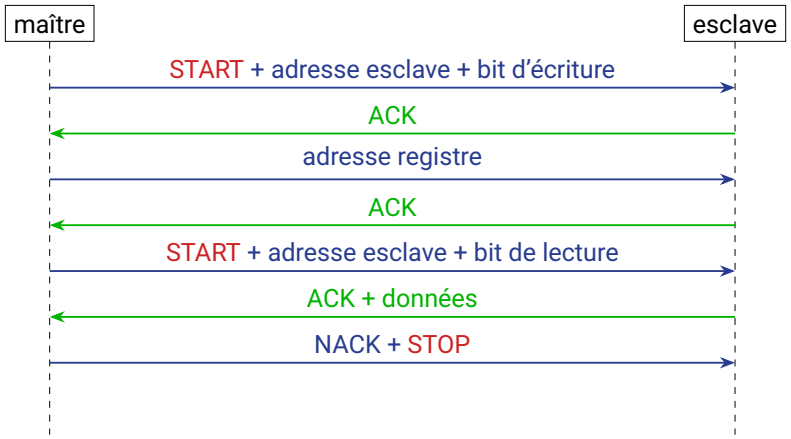
Protocole i^2c

Écriture d'un octet



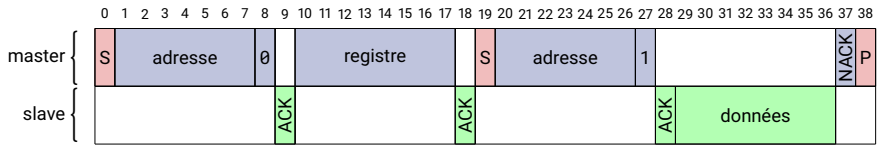
Protocole i^2c

Lecture d'un octet



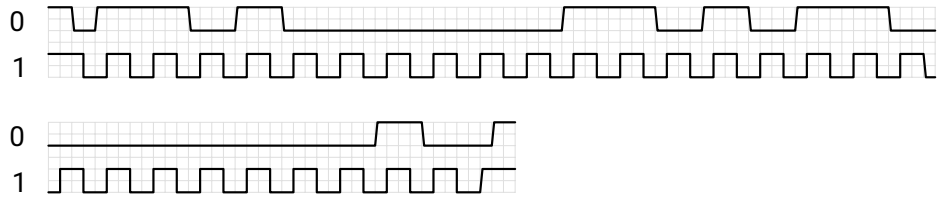
Protocole i^2c

Lecture d'un octet



Exercice

Décodage d'une communication i^2c

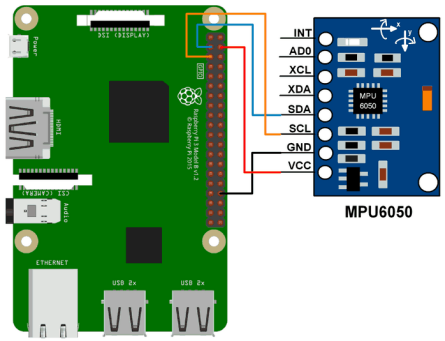


Voici une communication i^2c capturée par un analyseur logique.
Quelle est la ligne SDA ? Quelle est la ligne SCL ? Est-ce une lecture ou une écriture ? Quelles sont les adresses de l'esclave et du registre ? Quelles sont les données échangées ?

Démonstration i^2c

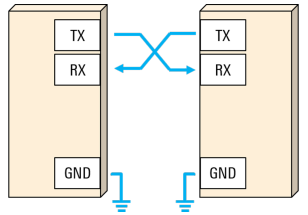
Accéléromètre MPU-6050

1. Connecter le RPi et l'accéléromètre.
2. Exécuter le programme.
3. Observer les signaux avec un analyseur logique.



Protocole UART

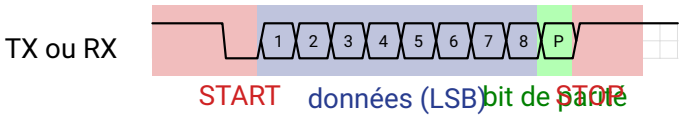
- Bus série asynchrone bidirectionnel.
- 3 fils : RX et TX pour les données, GND pour la tension de référence.
- 2 participants.
- Simplex, half-duplex ou full-duplex.
- Vitesse de transfert courantes entre 480 bps et 115 200 bps



Rohde & Schwarz, https://www.rohde-schwarz.com/nl/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html, consulté le 2025-02-09.

Protocole UART

Codage des données



Configuration

- Vitesse de transmission (baud rate).
- Nombre de bits de données : 5 à 9.
- Bit de parité : aucun (N), pair (E) ou impair (O).
- Nombre de bits de STOP : 1 ou 2.

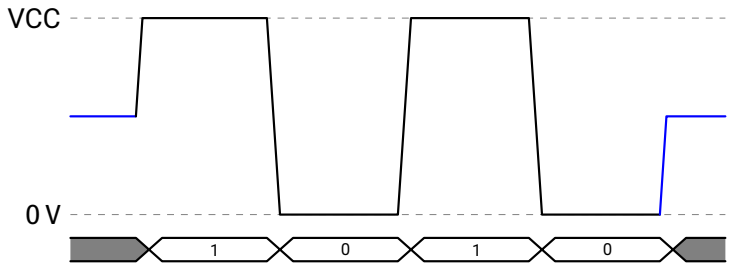
Exemple

- 9600-8-N-1 : 9600 baud, 8 bits de données, pas de bit de parité, 1 bit de STOP.
- 115200-7-E-2 : 115200 baud, 7 bits de données, bit de parité pair, 2 bits de STOP.

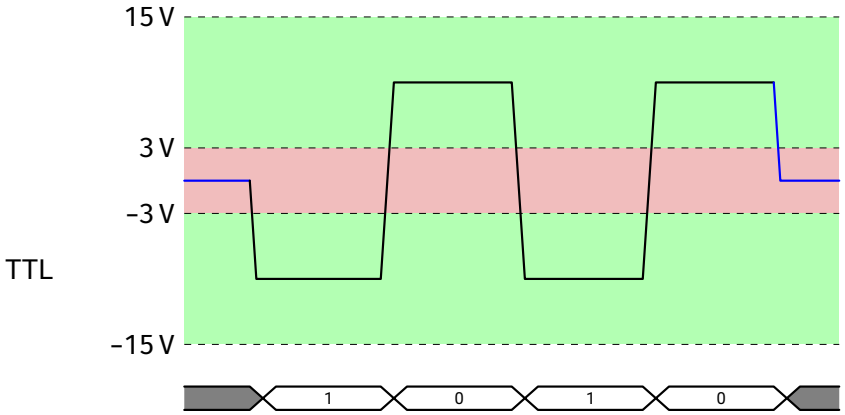
Signaux

- TTL : standard pour les communications série sur des courtes distances.
- RS-232 : standard historique pour les communications série.
- RS-485 : standard pour les communications série en réseau.

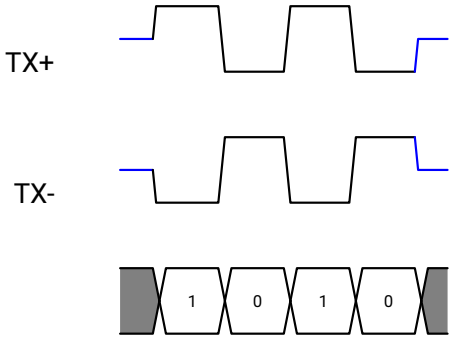
TTL



RS-232



RS-485



Au moins 1,5 V
entre TX+ et TX-

Exercice

Décodage d'une communication UART

Voici un signal UART TTL 115200-7-E-2 capturé par un analyseur logique.
Quelles sont les données échangées ?

Résumé

- Problématique de la synchronisation
- Génération d'un signal d'horloge
- Bus de communication, parallèle ou série, synchrone ou asynchrone
- Protocoles i^2c et UART
- Signaux TTL, RS-232, RS-485

Progression

- Systèmes embarqués ✓
- Bus de communication ✓
- Métrologie et gestion des capteurs
- Perception de l'environnement
- Contrôle de l'environnement