

Série d'exercices 3: Communications séries Synchrone/Asynchrone

Exercice 1 : Calcul du débit et de la durée d'une trame UART

On configure un UART avec les paramètres suivants :

- Débit = 9600 bauds
 - 1 bit de start
 - 8 bits de données
 - 1 bit de parité paire
 - 1 bit de stop
1. Calculer la durée totale d'une trame.
 2. Combien d'octets peut-on transmettre au maximum par seconde ?

Exercice 2 : Configuration et envoi d'un caractère en UART

Un microcontrôleur envoie le caractère 'A' (ASCII 0x41) sur sa liaison UART.
Format : 8 bits de données, 1 stop, sans parité, LSB first.

1. Dessiner le signal sur la ligne Tx (niveau logique 0 = 0V, 1 = Vcc).
2. Indiquer l'état de la ligne au repos.

Exercice 3 : I2C – Lecture d'un capteur avec adressage 7 bits

Un capteur I2C a l'adresse 0x40 (7 bits). On veut lire 2 octets du registre 0x10.

Décrire la séquence complète sur les lignes SDA et SCL, avec les conditions START, STOP, ACK/NACK, et le bit R/W.

Exercice 4 : I2C – Détection d'un périphérique

Vous voulez détecter si un périphérique I2C à l'adresse 0x30 (7 bits) est présent sur le bus.

Proposer un algorithme simple.

Exercice 5 : SPI – Configuration des modes CPOL/CPHA

Un périphérique SPI nécessite les paramètres :

- La donnée est capturée sur le front montant de SCK
 - Le repos de SCK est à l'état haut
1. Quel mode SPI (CPOL, CPHA) correspond ?
 2. Dessiner la forme d'onde pour l'envoi de 0b1010, MOSI, avec MSB first.

Exercice 6 : Comparaison UART / I2C / SPI

Pour chaque protocole, citer un avantage et un inconvénient, ainsi qu'une application typique.

Solution Série 3

Exercice 1 :

1. **Nombre total de bits par trame :**

Start (1) + données (8) + parité (1) + stop (1) = 11 bits

Durée d'un bit = $(1/9600) \cdot s \approx 104,17 \mu s$

Durée trame = $11 \times 104,17 \mu s \approx 1,1459 \text{ ms}$

2. **Nombre de trames par seconde :**

$1/1,1459 \text{ ms} \approx 872 \text{ trames/s}$

Chaque trame contient 1 octet de données utiles \rightarrow **872 octets/s.**

Exercice 2 :

Trame : Start (0) + 8 bits (0x41 = 0100 0001 en binaire, mais LSB first \rightarrow 1000 0010) + Stop (1).

- Start : 0
- Bit 0 (LSB) : 1
- Bit 1 : 0
- Bit 2 : 0
- Bit 3 : 0
- Bit 4 : 0
- Bit 5 : 0
- Bit 6 : 1
- Bit 7 (MSB) : 0
- Stop : 1

Signal (0V = 0, Vcc = 1) :

Repos (1) \rightarrow Start (0) \rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow Stop (1) \rightarrow Repos (1)

Ligne au repos : niveau 1 (logique haute).

Exercice 3 :

- **Adresse 7 bits :** 0x40 = 100 0000
- **Bit R/W :** 0 pour écriture (write), 1 pour lecture (read).

Étape 1 (écriture du registre) :

START \rightarrow Adresse (0x40) + R/W=0 \rightarrow ACK (esclave) \rightarrow Registre (0x10) \rightarrow ACK \rightarrow (RESTART)

Étape 2 (lecture des données) :

RESTART \rightarrow Adresse (0x40) + R/W=1 \rightarrow ACK \rightarrow Lecture octet1 \rightarrow MASTER ACK \rightarrow Lecture octet2 \rightarrow MASTER NACK \rightarrow STOP

Séquence résumée :

Étape	SDA (bit par bit)	SCL
START	SDA ↓ pendant SCL=1	
Write	10000000 (adresse+write)	8 pulses
ACK	Esclave tire SDA=0	1 pulse
Write	00010000 (registre 0x10)	8 pulses
ACK	Esclave tire SDA=0	1 pulse
START	SDA ↓ pendant SCL=1	
Write	10000001 (adresse+read)	8 pulses
ACK	Esclave tire SDA=0	1 pulse
Read	Octet 1 (maître lit)	8 pulses
ACK	Maître tire SDA=0	1 pulse
Read	Octet 2 (maître lit)	8 pulses
NACK	Maître tire SDA=1	1 pulse
STOP	SDA ↑ pendant SCL=1	

Exercice 4 :

Envoyer un **START**, puis l'adresse + bit R/W=0, puis attendre l'ACK.

Si l'esclave répond par ACK (tire SDA à 0), il est présent.

Sinon, s'il ne répond pas (SDA reste 1), il est absent ou HS.

Terminer par STOP.

Début

```
I2C_Start()

if I2C_Write(0x30 << 1 | 0) == ACK_reçu // adresse 0x30 + write

    Afficher "Périphérique trouvé"

else

    Afficher "Absent"

I2C_Stop()

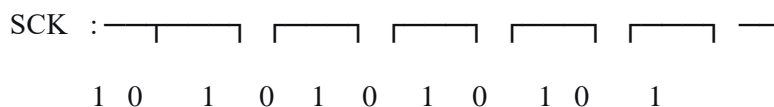
Fin
```

Exercice 5 :

1. **CPOL** : repos haut → CPOL = 1
CPHA : capture sur front montant → CPHA = 0 (car front montant = premier front si CPHA=0, sinon deuxième front)
 → **Mode 2** (CPOL=1, CPHA=0) selon la convention (Mode 0: CPOL=0, CPHA=0 ; Mode 1: 0,1 ; Mode 2: 1,0 ; Mode 3: 1,1)
2. SCK repos = 1.
 Capture sur front montant (donc donnée préparée sur front descendant précédent).
 Pour 0b1010 (MSB first : bit3=1, bit2=0, bit1=1, bit0=0) :

- Front descendant SCK : MOSI change
- Front montant SCK : échantillonné

Chronogramme :



MOSI : 1 0 1 0

(front montant capture : 1,0,1,0)

Exercice 6 :

Protocole	Avantage	Inconvénient	Application typique
UART	Simple, seulement 2 fils	Pas d'horloge commune, débit limité, un seul esclave	Module GPS, Bluetooth HC-05

Université Djilali BOUNAAMA, Khemis Miliana

مليانة خميس بونعامه جيلالي جامعة

Faculté des Sciences et de la Technologie

والتكنولوجيا العلوم كلية



Protocole	Avantage	Inconvénient	Application typique
I2C	Adressage multi-esclaves, 2 fils	Plus lent que SPI	Capteurs (BMP280, MPU6050)
SPI	Full-duplex, très rapide	Nécessite plus de fils (CS par esclave)	Écran LCD, carte SD