```
from machine import UART, Pin
import time

# ---------------------------------
# PARAMÈTRES À ADAPTER
# ---------------------------------
uart = UART(2, baudrate=9600)   # mets ici l'UART utilisé sur ta STM32

led_ok = Pin("PA5", Pin.OUT)    # LED pour signaler un badge autorisé
led_ok.off()

# Un seul badge autorisé (à remplacer par ton vrai ID)
BADGE_AUTORISE = "0F01A23B9C"

buffer = b""

print("Contrôle d'accès RFID 125 kHz prêt.")


while True:
    if uart.any():
        data = uart.read()
        if data:
            buffer += data

            # fin de trame (selon version : \n, \r ou 0x03)
            if (b'\n' in buffer) or (b'\r' in buffer) or (b'\x03' in buffer):

                # Nettoyage de la trame
                cleaned = (
                    buffer.replace(b'\r', b'')
                          .replace(b'\n', b'')
                          .replace(b'\x02', b'')
                          .replace(b'\x03', b'')
                          .strip()
                )

                if cleaned:
                    try:
                        tag_str = cleaned.decode('ascii')
                    except:
                        tag_str = str(cleaned)

                    print("Badge lu :", tag_str)

                    if tag_str == BADGE_AUTORISE:
                        print("Accès autorisé")
                        led_ok.on()
                        time.sleep(1)
                        led_ok.off()
                    else:
                        print("Accès refusé")

                # reset buffer pour le badge suivant
                buffer = b""

    time.sleep_ms(20)
```

CONTRÔLE TRAME RFID

Output Data Format

0x02	10ASCII Data Characters	Checksum	0x03

0x02 - 1 byte start flag

10 ASCII Data Characters – Card number info

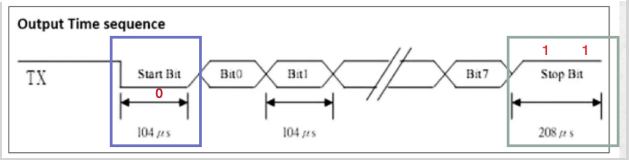
Checksum - 2 bytes

0x03 - 1 byte end flag



Exercice

Code RFID transmis : 0A23F64C7B



Travail demandé :

1) Déterminer le nombre total d'octets transmis.

2) Calculer le checksum (XOR des 5 octets du code RFID).

3) Calculer le code carte imprimé sur la carte.

4) Tracer la trame oscillogramme des 3 premiers octets transmis (0x02, puis 1er et 2e caractère ASCII).

1) Nombre total d'octets => Calcul :

Résultat : .....

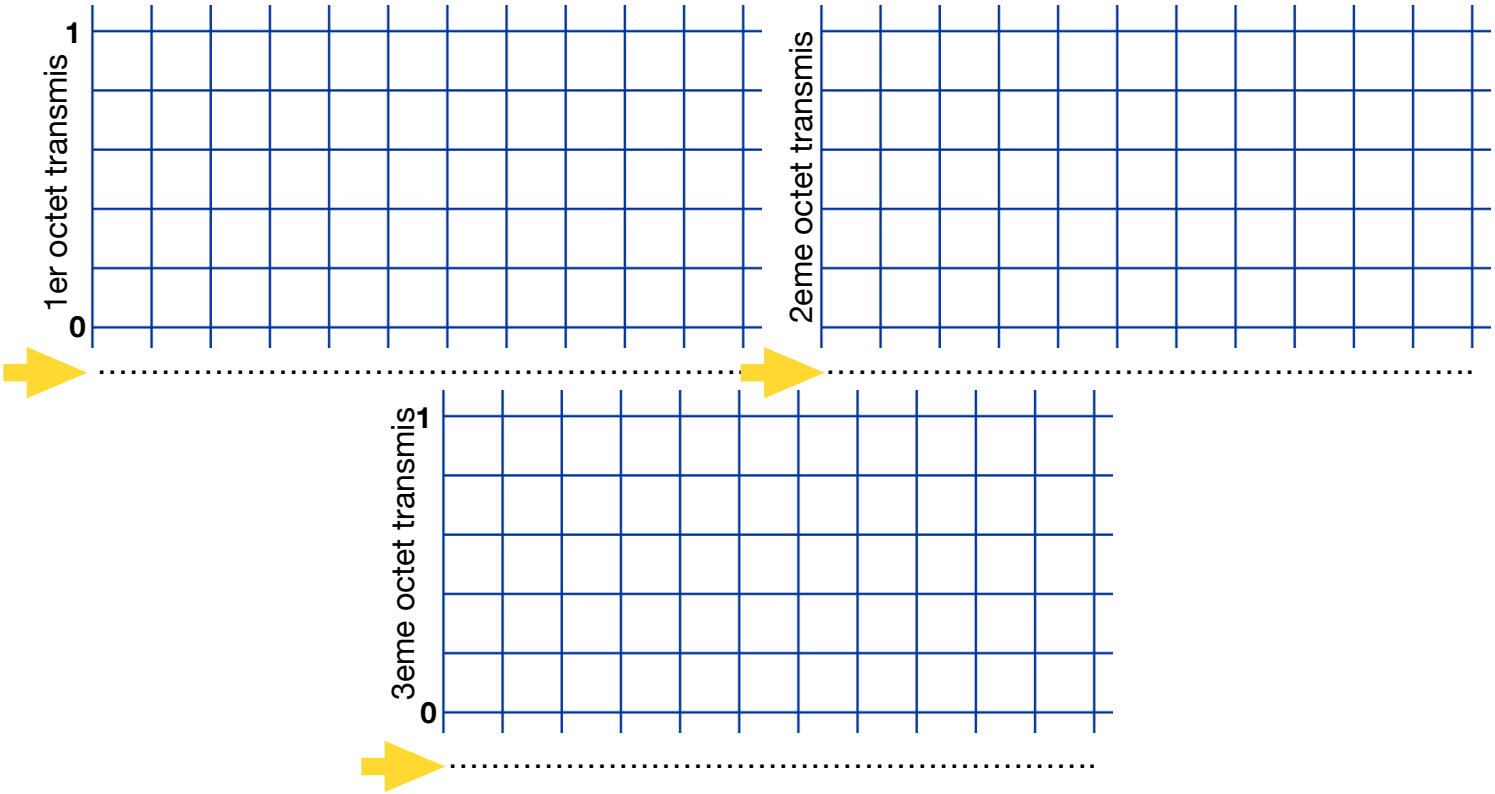
2) Checksum => Calcul :

Résultat : .....

3) Code carte => Calcul :

Résultat : .....

4)



CONTRÔLE TRAME RFID

Output Data Format

0x02	10ASCII Data Characters	Checksum	0x03

0x02 - 1 byte start flag

10 ASCII Data Characters – Card number info

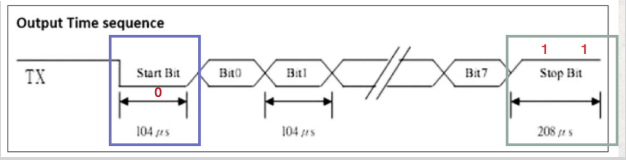
Checksum - 2 bytes

0x03 - 1 byte end flag



Exercice

Code RFID transmis : 7C01A9F433



Travail demandé :

1) Déterminer le nombre total d'octets transmis.

2) Calculer le checksum (XOR des 5 octets du code RFID).

3) Calculer le code carte imprimé sur la carte.

4) Tracer la trame oscillogramme des 3 premiers octets transmis (0x02, puis 1er et 2e caractère ASCII).

1) Nombre total d'octets => Calcul : .....

Résultat : .....

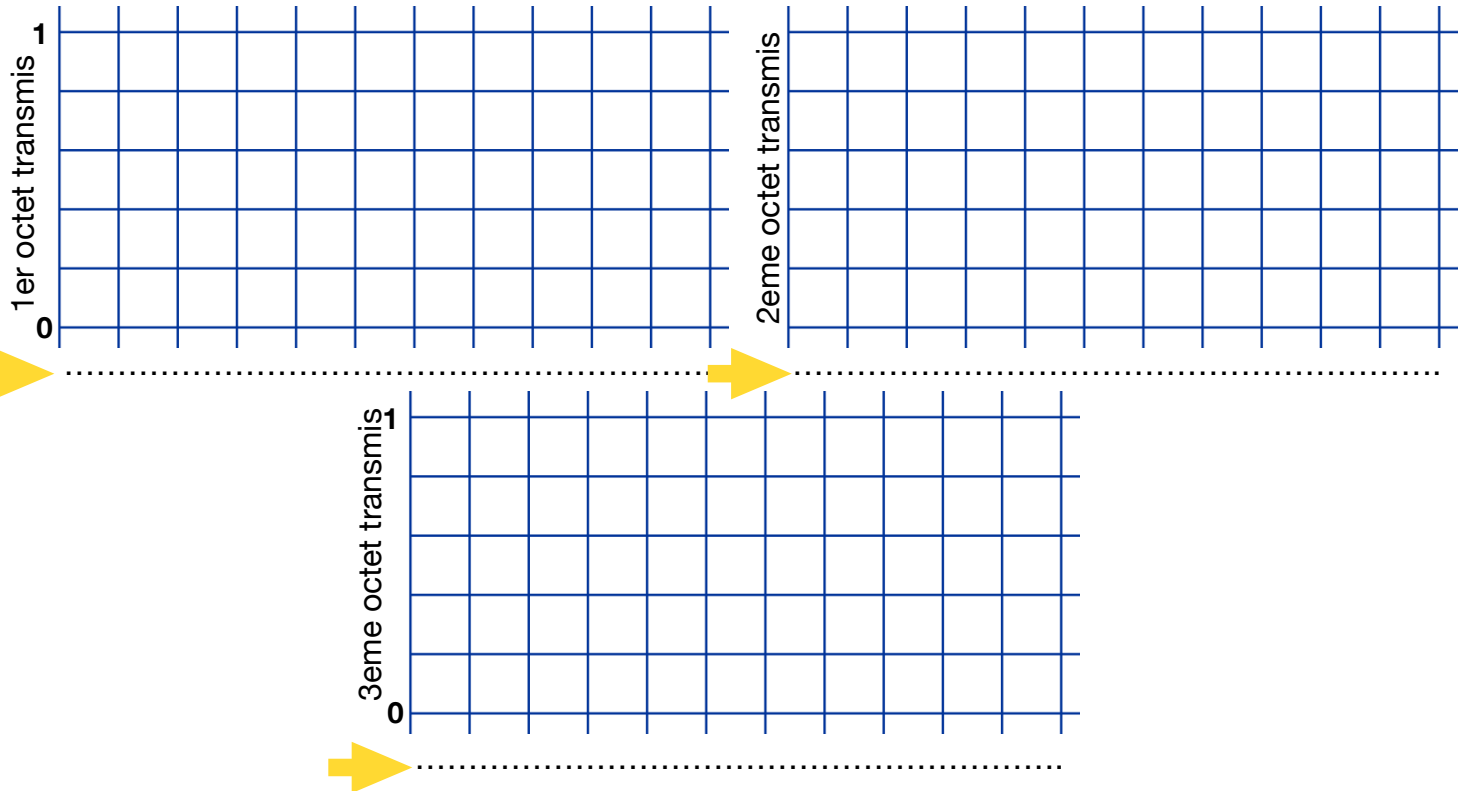
2) Checksum => Calcul : .....

Résultat : .....

3) Code carte => Calcul : .....

Résultat : .....

4)



CONTRÔLE TRAME RFID

Output Data Format

0x02	10ASCII Data Characters	Checksum	0x03

0x02 - 1 byte start flag

10 ASCII Data Characters – Card number info

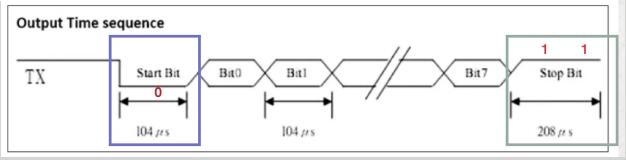
Checksum - 2 bytes

0x03 - 1 byte end flag



Exercice

Code RFID transmis : 12AF56B729



Travail demandé :

1) Déterminer le nombre total d'octets transmis.

2) Calculer le checksum (XOR des 5 octets du code RFID).

3) Calculer le code carte imprimé sur la carte.

4) Tracer la trame oscillogramme des 3 premiers octets transmis (0x02, puis 1er et 2e caractère ASCII).

1) Nombre total d'octets => Calcul : .....

Résultat : .....

2) Checksum => Calcul : .....

Résultat : .....

3) Code carte => Calcul : .....

Résultat : .....

4)

