TP Transmission Numérique

Série Asynchrone

Kimberley Jacquemot & Ayoub Ech Chamali

02/12/2022

I - Introduction

I.a. - Questions

1) Quel est le bloc logique interne à l'ATMEGA328P qui devra être utilisé afin de mettre en œuvre ce type de transmission ?

C'est le module USART0, nous utiliserons les horloges pour un type asynchrone comme détaillé dans l'extrait de la documentation ci-dessous.

8.1.2 I/O Clock – $clk_{I/O}$

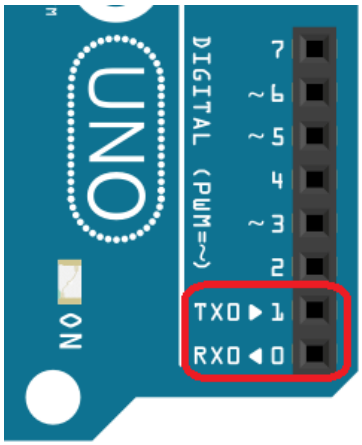
The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that start condition detection in the USI module is carried out asynchronously when $clk_{I/O}$ is halted, TWI address recognition in all sleep modes.

Figure 1 - Description fonctionnement asynchrone Clock

2) Quel est le principe physique mis en œuvre ? La sonde est-elle adaptée (tension) ?

La sonde n'est pas adaptée, le crochet n'est pas un embout adapté pour prendre des mesures sur des composants de microcontrôleur.

3) Quelles sont les pattes qui représentent Rx (réception) et Tx (Transmission) de l'ARDUINO UNO ?



Ce sont les pattes qui envoient et reçoivent les informations. Sur l'Arduino UNO on peut les identifier comme les pins 0 (RX → Réception) et 1 (TX → Transmission). On pourrait préciser que ce sont des digital I/O.

4) Quels sont tous les registres qui devront être utilisés ? Donnez leurs descriptions.

On peut lire ces informations dans la documentation.

6.3.1 SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

5) Quelles sont les informations disponibles dans ces registres ?

I : Autorisation des interruptions

T : Bit de stockage pour copier un bit (Source ou Destinataire)

H : Bit de demie-retenue pour les opérations de calcul

S : Bit de signe positif ou négatif

V : Bit de complément à 2 pour les opérations de calcul

N : Bit de négatif, toujours pour les opérations de calcul

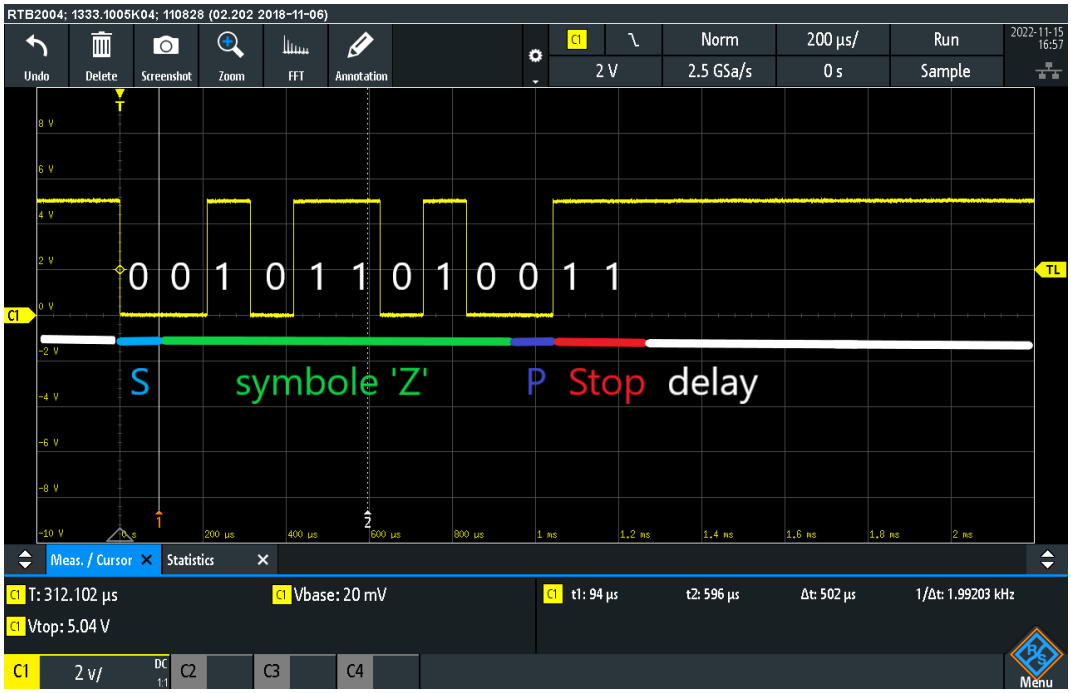
Z : Bit retournant un résultat nul

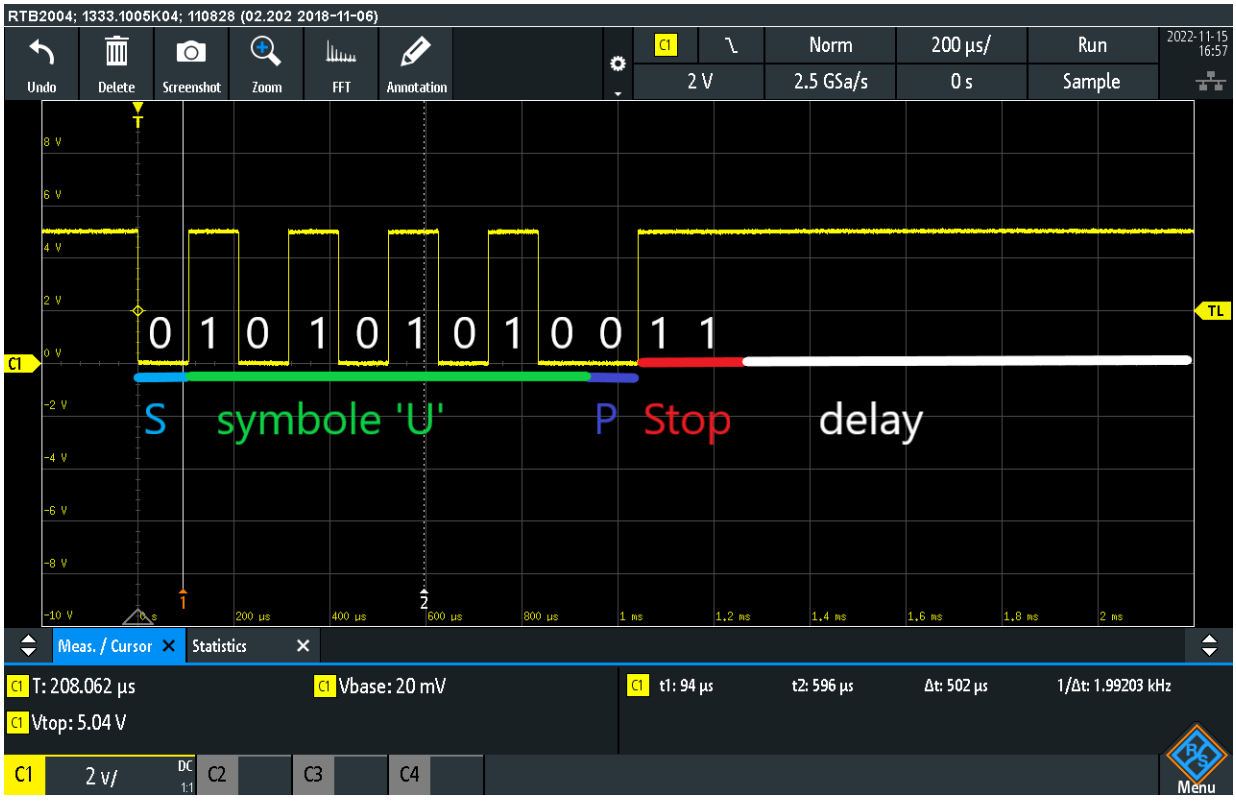
C : Bit de retenu pour les opérations de calcul.

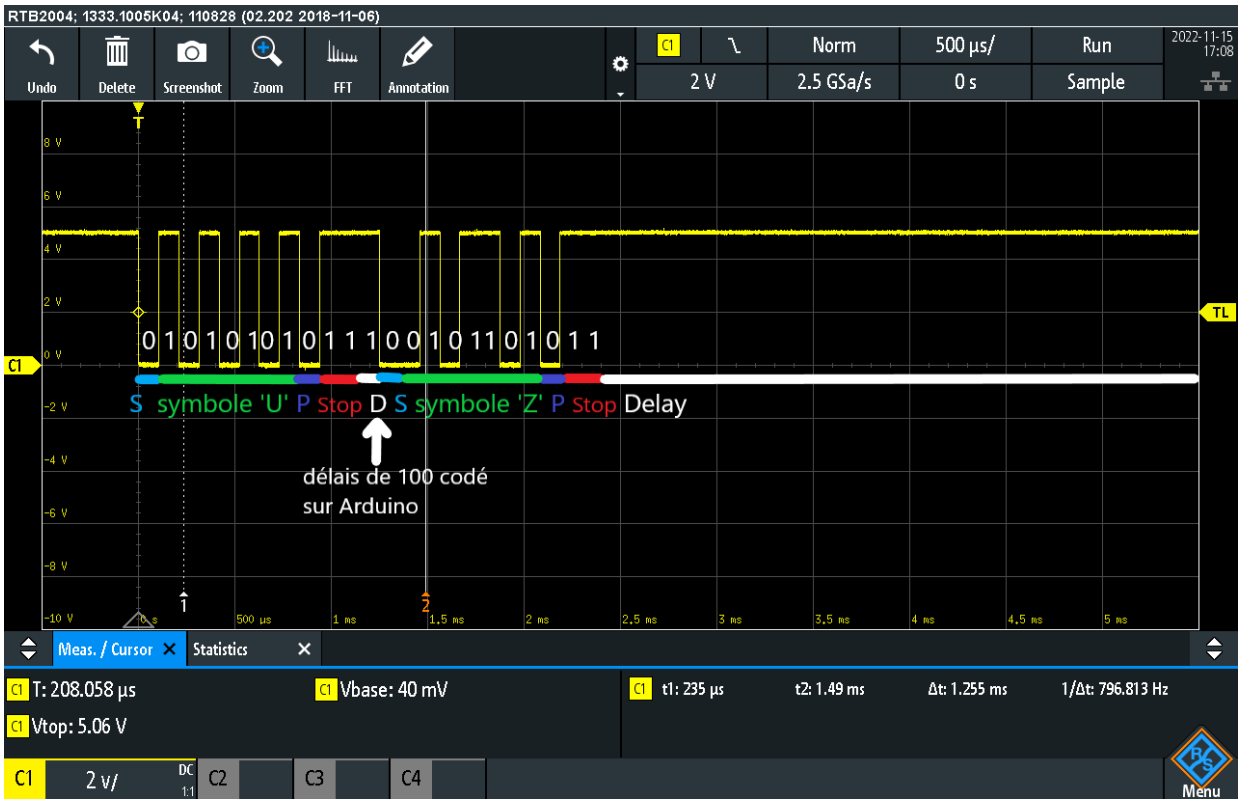
6) Écrire un Sketch dans l'IDE ARDUINO qui permet d'envoyer des symboles avec les caractéristiques de transmission suivantes : 9600 Bauds, 8 bits de données, parité paire, 2 Stops.

Figure 2 - Extrait du sketch Arduino

7) Capturez les diagrammes des temps obtenus pour la transmission du symbole : 'Z', 'U', puis 'U' suivi de 'Z'. Que concluez-vous concernant les bits Stop ?



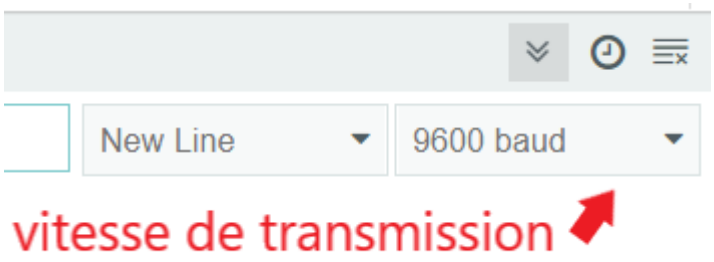

RTB2004; 1333.1005K04; 110828 (02.202 2018-11-06)
Undo
Delete
Screenshot
Zoom
FFT
Annotation
Norm
200 µs/
Run
2 V
2.5 GSa/s
0 s
Sample
2022-11-15
16:57
0 1 0 1 0 1 0 1 0 0 1 1
S
symbole 'U'
P
Stop
delay
Meas. / Cursor
Statistics
C1 T: 208.062 µs
C1 Vbase: 20 mV
C1 t1: 94 µs
t2: 596 µs
Δt: 502 µs
1/Δt: 1.99203 kHz
C1 Vtop: 5.04 V
C1 2 V/
C2
C3
C4
Menu


RTB2004; 1333.1005K04; 110828 (02.202 2018-11-06)
Undo
Delete
Screenshot
Zoom
FFT
Annotation
Norm
500 µs/
Run
2 V
2.5 GSa/s
0 s
Sample
2022-11-15
17:08
0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1
S symbole 'U' P Stop D S symbole 'Z' P Stop Delay
délais de 100 codé
sur Arduino
Meas. / Cursor
Statistics
C1 T: 208.058 µs
C1 Vbase: 40 mV
C1 t1: 235 µs
t2: 1.49 ms
Δt: 1.255 ms
1/Δt: 796.813 Hz
C1 Vtop: 5.06 V
C1 2 V/
C2
C3
C4
Menu

Les bits de Stop ne sont utiles que quand nous avons des informations à la suite, puisqu'il est à 1 il est similaire au temps de délais.

8) Mesurez le temps nécessaire à la transmission d'un bit. Capturez le diagramme temporel obtenu.

Sur notre graphique nous mesurons 100µs pour un bit et environ 1ms pour une trame d'un symbole.



La vitesse de transmission théorique est de 9600 bauds, on devrait mettre (1/9600) 104 µs à transmettre un symbole, nous sommes proche de cette valeur.

9) Quels sont les intérêts des bits de Parité et de Stop.

Le bit de parité permet d'éviter les erreurs en vérifiant qu'il n'y a pas d'erreur à la fin.

Les bits de Stop permettent de faire une pause entre les trames de données.

10) En déduire le rendement de ce mode de transmission.

100/104 = 0,96 on a 96% de rendement pour la vitesse de transmission.

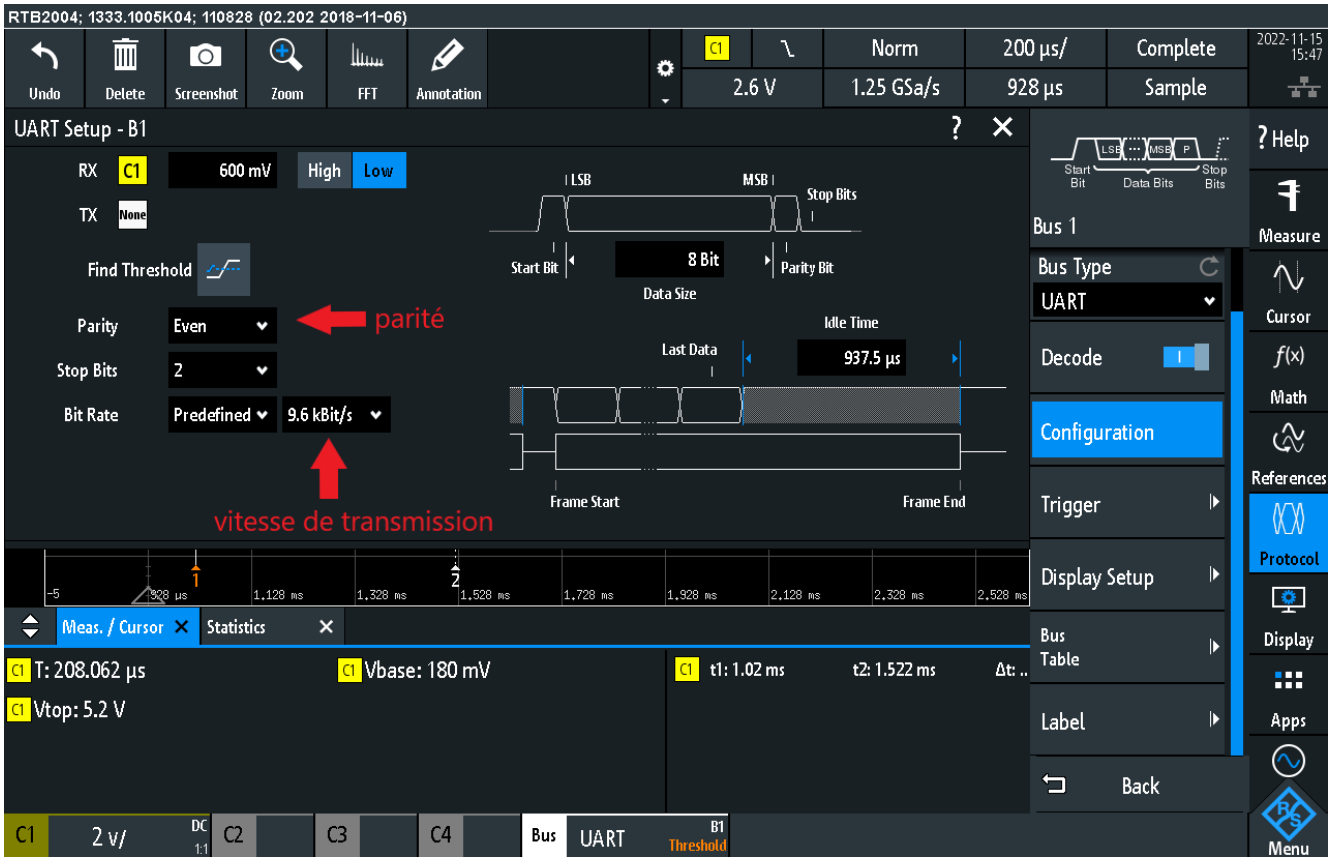
Les 4% restant peuvent s'additionner en fonction du nombre de caractères et créer un décalage par la suite.

II - Carte de mêmes caractéristiques

II.a. - Questions

1) Quels sont les critères techniques permettant que la communication fonctionne pleinement parfaitement ?

Pour un fonctionnement parfait il faudrait avoir une même vitesse de transmission (par exemple 9600 bauds) Une même trame avec un bit de parité ainsi que des pauses pour laisser les signaux se resynchroniser lorsqu'il y a un décalage (il faut paramétrer la trame des deux côtés). Sans cela, il est impossible pour la deuxième carte de comprendre le message envoyé.



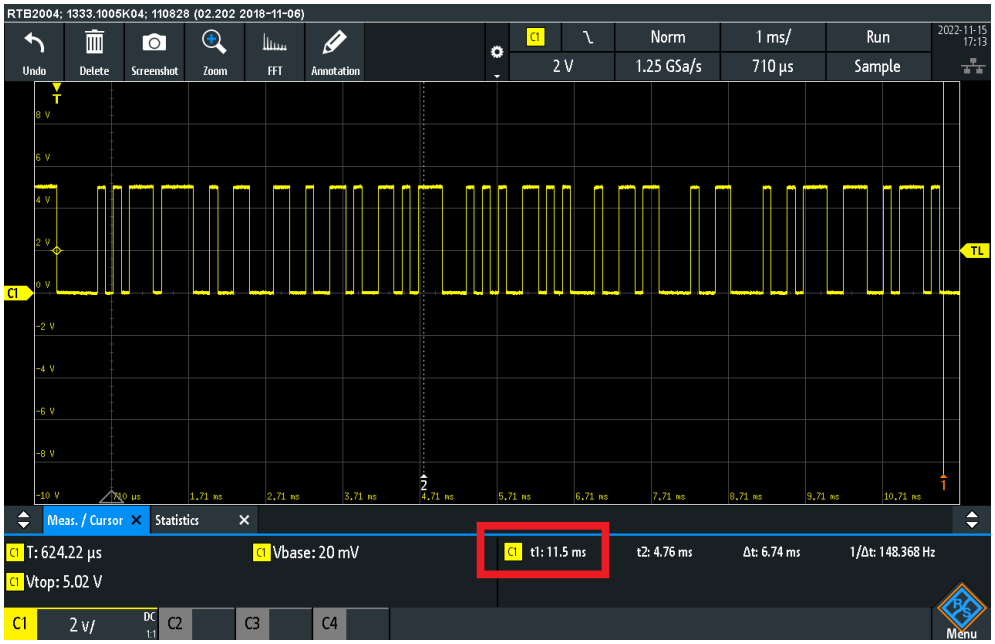
2) Écrire un Sketch dans l'IDE ARDUINO qui permet d'envoyer des symboles avec les caractéristiques de transmission suivantes : 19200 Bauds, 8 bits de données, parité paire, 2 Stops.

TPAsynchrone.ino

```
void setup()
{
  // On règle à une vitesse de 9600 bits par seconde.
Serial.begin(9600, SERIAL_8O2); //règle la vitesse de transmission à 9600 bauds, avec 8 bits de données, parité paire et un bit stop.
}
void loop() {

  Serial.print('P'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('O'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('L'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('Y'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('T'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('E'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('C'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('H'); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print(' '); //Mon symbole de 8 bits de données
  delay(100);
  Serial.print('N'); //Mon symbole de 8 bits de données
```

3) Echangez le texte « Polytech Nancy » de la carte 1 vers la carte 2.



4) Quel est le temps nécessaire ?

Pour un caractère on a : 1(start) + 7(data) + 1(parité) + 2(stop) = 11 bits.

La durée de transmission pour un bit étant de t = 1/9600, on a donc 11*t environ égal à 1ms.

La durée de transmission d'un caractère est d'environ 1ms.

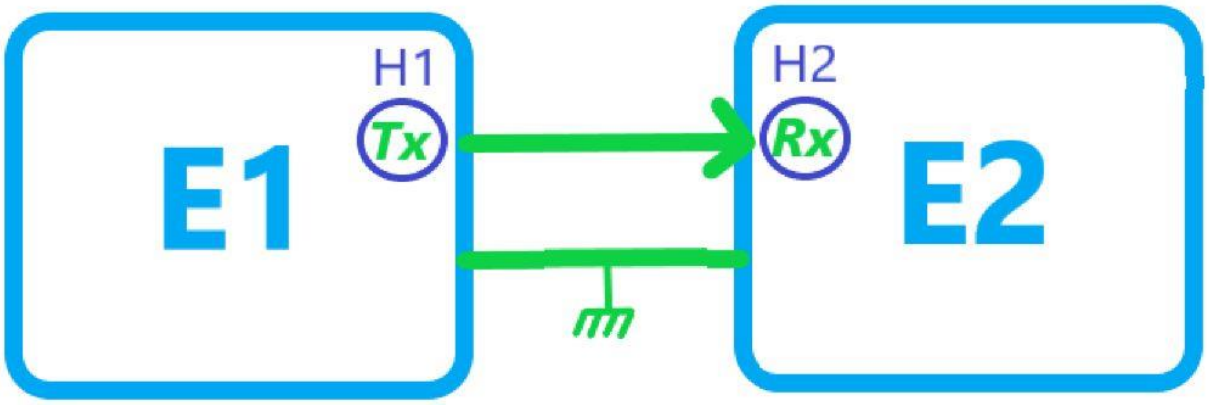
Pour t = 1/19200, la durée est d'environ 0,6ms.

III - Carte de caractéristiques différentes

III.a. - Questions

1) Quels sont les problèmes matériels rencontrés par cette architecture ?

Les horloges internes des appareils ne sont pas réglées de la même manière, ce qui engendre alors un décalage dans les transmissions qui s'aggravera en fonction de la taille du message.



2) Quelle est la solution obligatoire d'avoir des transmissions parfaitement opérationnelles ?

Il faut mettre un assez long délai pour laisser le temps à la réception pour se synchroniser entre chaque symbole.

IV - Standard matériel RS232/V24

IV.a. - Questions

1) Quels sont les composants nécessaires afin de respecter cette norme ?

Il nous faudrait des ports de type série.

2) Quel est le format du connecteur standardisé ?

Le connecteur standard est le DE-25.