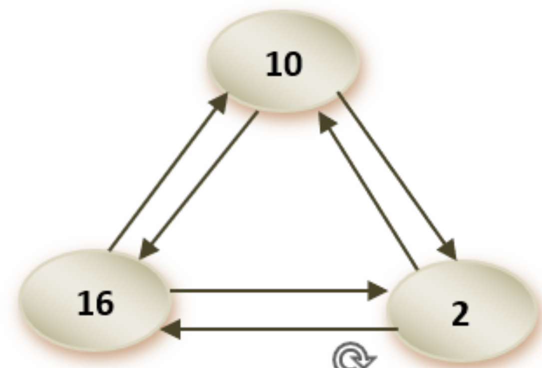
3. Transcodage

Unité de codage : Les composants constituant un système informatique réagissent de manière interne à des signaux « tout ou rien ». Il y a 2 états stables : **« 0 » = « L »** (low) et **« 1 » = « H »** (high)
La base 2 est la mieux adaptée, on parle de codage binaire.

Décimal	Binaire naturel	Hexadécimal
0	0000 0000	0
1	0000 0001	1
2	0000 0010	2
3	0000 0011	3
4	0000 0100	4
5	0000 0101	5
6	0000 0110	6
7	0000 0111	7
8	0000 1000	8
9	0000 1001	9
10	0000 1010	A
11	0000 1011	B
12	0000 1100	C
13	0000 1101	D
14	0000 1110	E
15	0000 1111	F
16	0001 0000	10



- 8 bits = 1 octet (byte)
- 16 bits = 2 octets (word) (mot)
- 1 Kio = 2^{10} octets = 1024 octets = 1 ko
- 1 Mio = 2^{20} octets = 1024 Kio = 1 Mo
- 1 Gio = 2^{30} octets = 1024 Mio = 1 Go
- 1 Tio = 2^{40} octets = 1024 Gio = 1 To

Pour plus d'info : http://fr.wikipedia.org/wiki/Octet

$1322_{(10)} = 1 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 2 \cdot 10^0$

	1	3	2	2
Rang	3	2	1	0
Poids	10^3	10^2	10^1	10^0
	1000	100	10	1

3.1. Transcodage 2 → 10

Rg 4 3 2 1 0
Pds 16 8 4 2 1

Il suffit d'appliquer la décomposition de base :

10110_2 ➔ $1.16 + 1.4 + 1.2 = 22_{(10)}$

ou $1.2^4 + 1.2^2 + 1.2^1 = 22_{(10)}$

Rg	2	1	0
Poids	2^2	2^1	2^0
	4	2	1

128 32 4 1
$10100101_{(2)} \rightarrow$

$128+32+4+1 = 165_{(10)}$

3.2. Transcodage 16 → 10

Il suffit une fois de plus de revenir à la définition du nombre, en multipliant chaque chiffre du nombre par le poids hexadécimal correspondant à son rang et en faisant la somme de ces produits :

$F8A_{16}$ ➔ $F.16^2 + 8.16^1 + A.16^0 = 15.16^2 + 8.16 + 10$

$= 3978_{(10)}$

rang
base
poids

3.3. Transcodage 2 → 16

16 étant égal à 2^4, la conversion de la base 2 à 16 consiste tout simplement à regrouper les termes du nombre en base 2 par groupes de 4 chiffres en commençant par la droite puis à convertir chaque groupe en écriture hexadécimale. On peut utiliser la table de correspondance ci-dessus pour la conversion.

Exemple :

$1101 \mid 0011$ (2)
$13 \mid 3$ (10) $\neq$
$\boxed{D \mid 3}$ (16)

HEX	D3
DEC	211
OCT	323
BIN	1101 0011

$1111 \mid 1111$
$F \mid F$

8 4 2 1 | 8 4 2 1
$1010 \mid 0111$
$10 \mid 7$ (10)
$\boxed{A \mid 7}$

3.4. Transcodage 10 → 2

On l'obtient par divisions successives par 2 du nombre. Le nombre en base 2 est constitué par les restes des divisions, la lecture s'effectuant du dernier reste vers le premier.

Exemple : $N_{10} = 237 \rightarrow N_2 =$

$$N_2 = 11101101_{(2)}$$

237 | 2
1 | 118 | 2
0 | 59 | 2
1 | 29 | 2
1 | 14 | 2
0 | 7 | 2
1 | 3 | 2
1 | 1 | 2
1 | 0

Sens lecture

3.5. Transcodage 10 → 16

On obtient par divisions successives par 16 du nombre. Le nombre en base 16 est constitué par les restes des divisions, la lecture s'effectuant du dernier reste vers le premier. Les restes devant être compris entre 0 et 15 soit 0 et F en hexadécimal.

Exemple :$N_{10} = 7346 \rightarrow N_{16} =$

Exemple :$N_{10} = 7346 \rightarrow N_{16} =$

$$7346 = 16 \times 459 + 2$$
$$459 = 16 \times 28 + 11$$
$$28 = 16 \times 1 + 12$$
$$1 = 16 \times 0 + 1$$

2

B

C

1

Sens de lecture

$$\boxed{N_{16} = 1CB2}$$

$$\begin{array}{r} 459,125 \\ -\ 459 \\ \hline 0,125 \end{array} \times 16$$

3.6. Transcodage 16 → 2

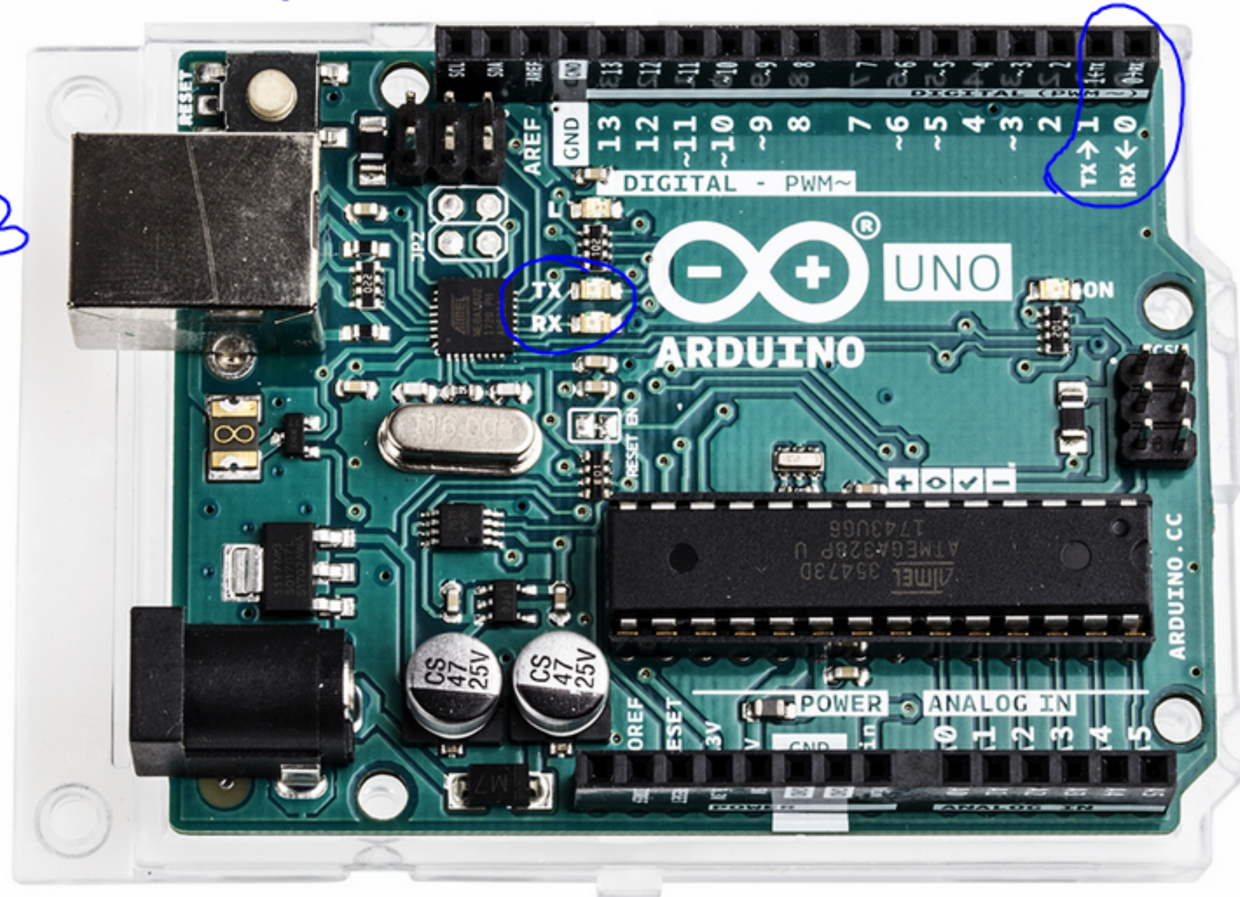
La technique consiste à former des groupes de 4 éléments binaires qui correspondent.

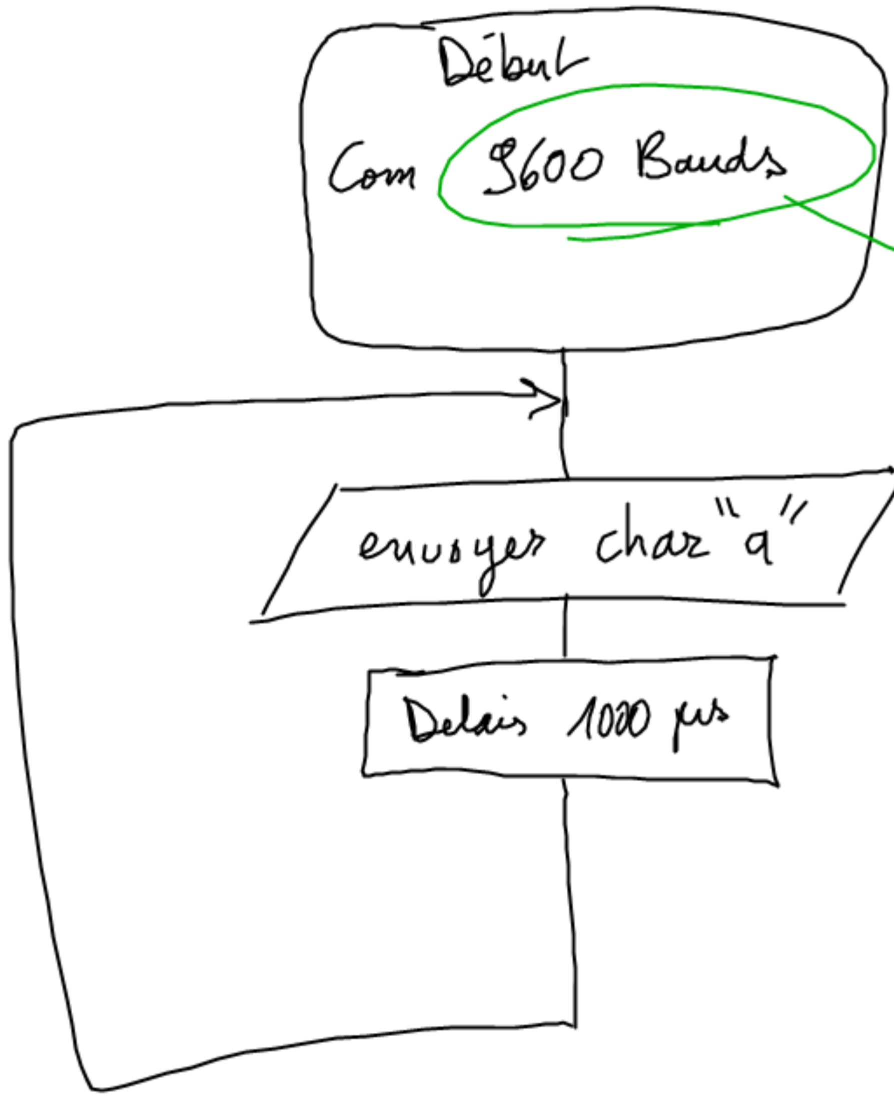
EDF_{16} → 1110 ⁝ 1101 ⁝ 1111

14 13 15

Tx = Transmit   Rx = Receive

USB





1 Baud

= 1 bit/s

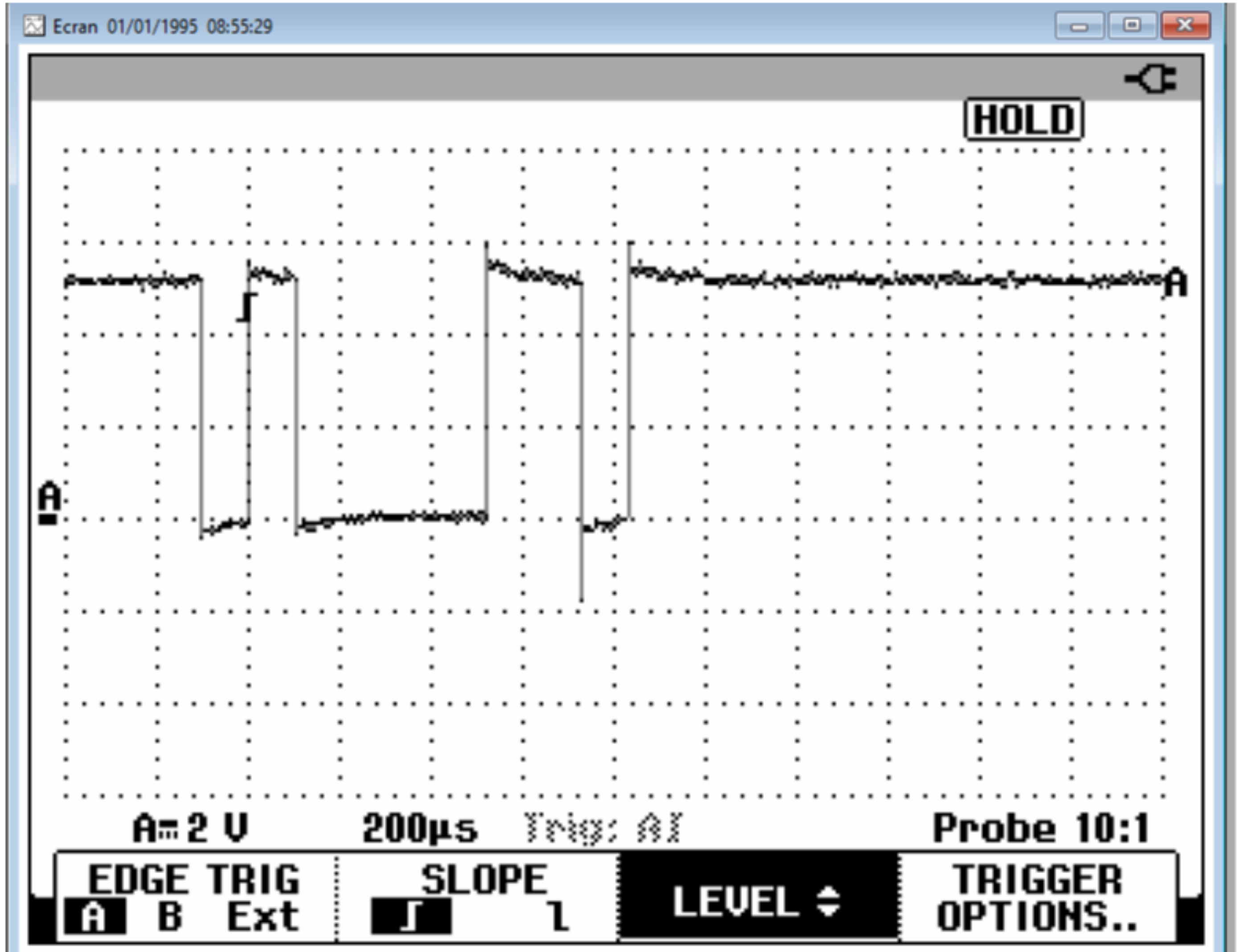
9600 bit/s

1 bit start
8 bit donnée
1 bit de stop

$$\frac{1}{9600} \cdot 10 = \frac{10}{9600} \simeq 1{,}04 \cdot 10^{-3}\,s$$

$$\simeq 1{,}04\ ms$$

(s/bit)


Ecran 01/01/1995 08:55:29
HOLD
A
A
A⎓2 V
200µs
Trig: A
Probe 10:1
EDGE TRIG
A B Ext
SLOPE
LEVEL
TRIGGER
OPTIONS..