

# Numérisation

ICN 1ère



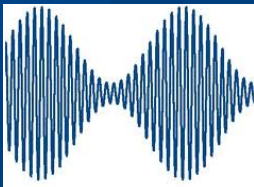
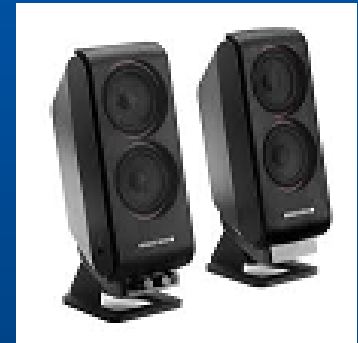
acquérir



traiter



communiquer



analogique



numérique

# Numérisation

ICN 1ère



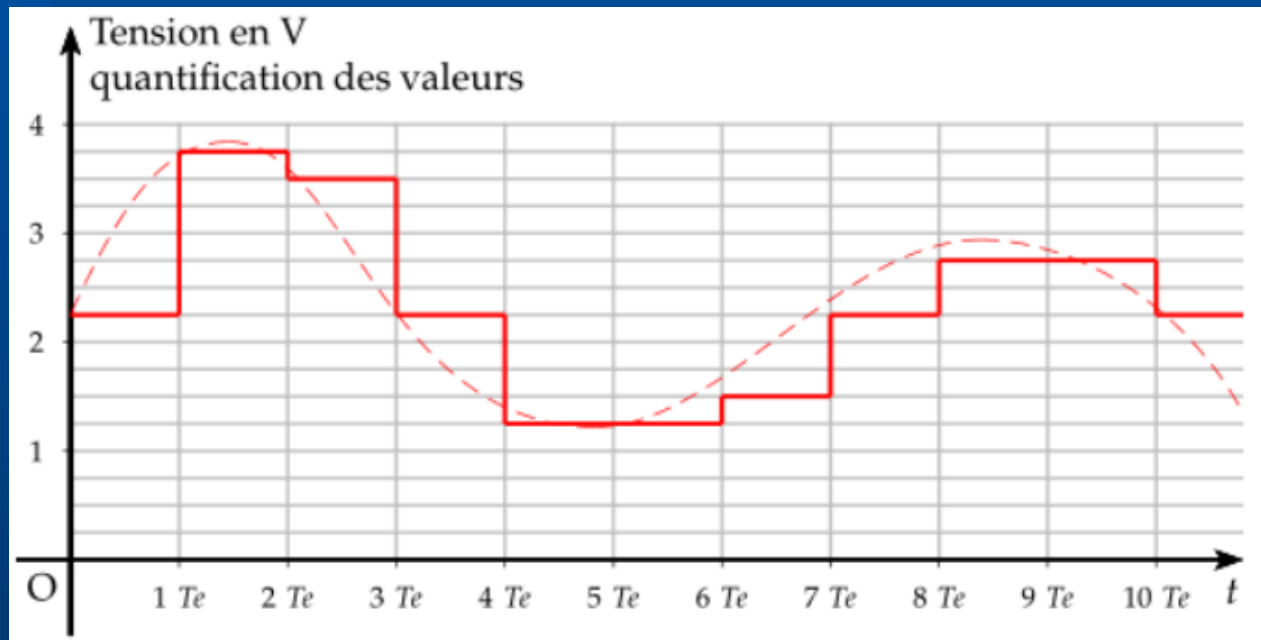
- **Analogique** : le signal varie de manière continue au cours du temps (mesure d'une grandeur physique).
- **Numérique** : la représentation du signal varie de façon discrète (ie : discontinue) dans une liste de valeurs.
- **Logique** : le signal est convertit dans un état binaire qui ne prend que deux valeurs, notées par convention 0 et 1 (logique Tout ou Rien, TOR).

# Numérisation

ICN 1ère



L'**échantillonnage** consiste à capturer les valeurs d'un signal **analogique** à **intervalles de temps** réguliers pour le convertir en signal **numérique**.



# Numérisation

ICN 1ère



**Numériser\*** : transformer les grandeurs continues dans le temps en grandeurs discontinues (discrètes) par intervalle de temps régulier



# Numérisation

ICN 1ère



## Base 2 ou binaire

Un mot est la plus petite unité manipulée par un microprocesseur.

Sa taille s'exprime :

- en **bits** (binary digit\*)
- ou en **octets** (1 octet = 8 bits).

Les ordinateurs utilisent des données de 8, 16, 32 ou 64 bits :

Ex : 01101111 11001100 10101111 11110111 (32 bits ou 4 octets).

# Numérisation

ICN 1ère



A chaque bit du mot en base 2 correspond une puissance de 2.

	Poids							
	forts ← MSD							→ faibles LSD
<b>Puissances</b>	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
<b>Valeurs en décimal du poids</b>	128	64	32	16	8	4	2	1
<b>exemple</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>Valeur en décimal correspondante</b>	64+32+8+4+2= 110							

Le digit de poids le plus fort est appelé **MSD** (Most Significant Digit).  
Le digit de poids le plus faible est appelé **LSD** (Least Significant Digit).

Exercice : Convertir le mot binaire %1100 1001 en décimal.

# Numérisation

ICN 1ère



## Méthode par division

On divise par 2 autant de fois que nécessaire pour obtenir un quotient nul et l'on écrit les restes dans l'**ordre inverse** où ils ont été obtenus.

$$\begin{array}{r} 211 \overline{) 2} \\ 1 \overline{) 105} \overline{) 2} \\ 1 \overline{) 52} \overline{) 2} \\ 0 \overline{) 26} \overline{) 2} \\ 0 \overline{) 13} \overline{) 2} \\ 1 \overline{) 6} \overline{) 2} \\ 0 \overline{) 3} \overline{) 2} \\ 1 \overline{) 1} \end{array}$$

Exercice : Convertir le nombre décimal 103 en binaire.

# Numérisation

ICN 1ère



## Base 16 ou Hexadécimal

Le codage en Hexadécimal (système de numération à base 16) permet de faciliter la représentation d'un mot binaire.

Ainsi l'octet %1101 0011 s'écrit D3 en hexadécimal

Ou encore : \$D3 ou 0xD3 ou D3h ...

A chaque **quartet** du mot binaire correspond un symbole en hexadécimal

Exercice : Convertir %0110 1011 en hexadécimale.

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

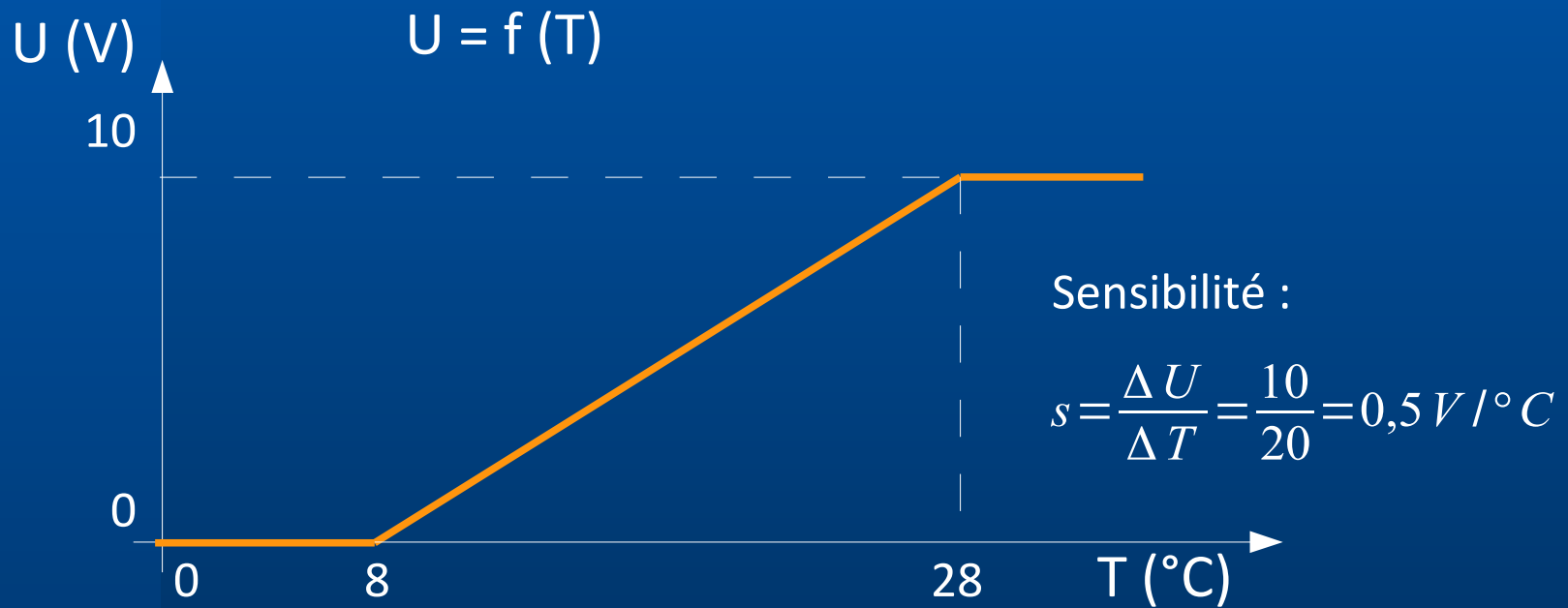


# Numérisation

ICN 1ère



Ex : capteur de t°



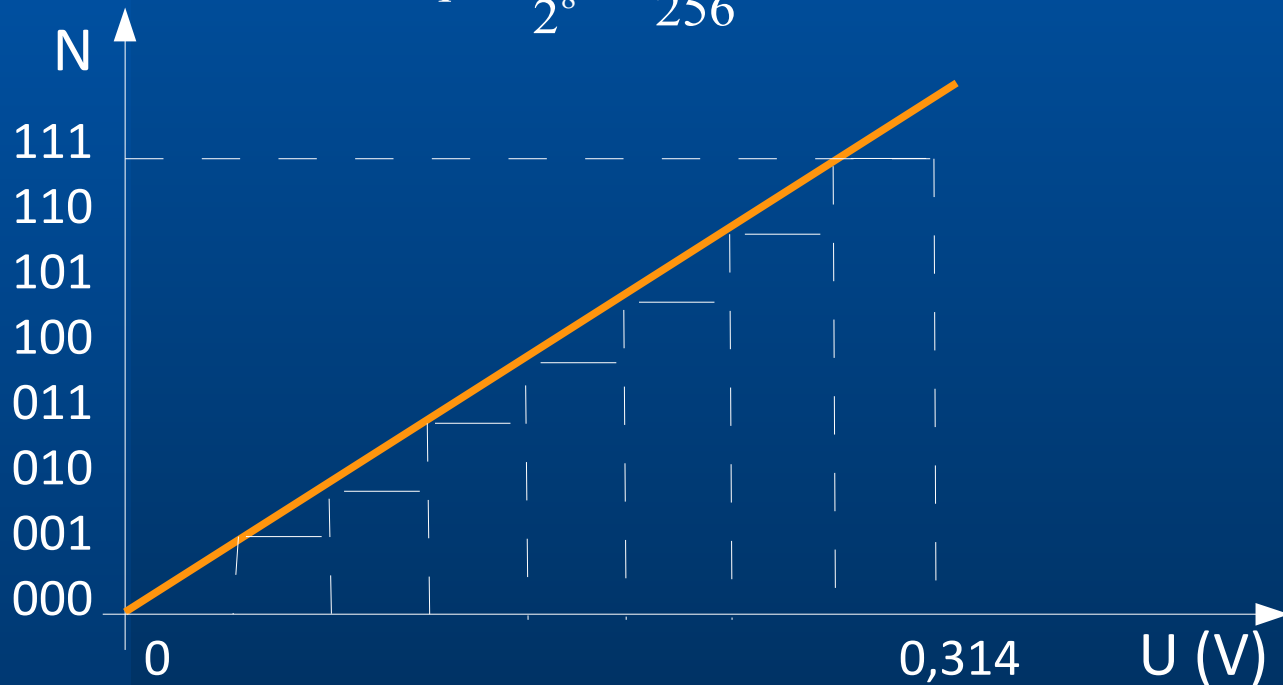
# Numérisation

ICN 1ère



## Le CAN

$$q = \frac{\Delta U}{2^8} = \frac{10}{256} = 39,06 \cdot 10^{-3} \text{ V} = 39,1 \text{ mV}$$



# Numérisation

## ICN 1ère



La table **ASCII** (American Standard Code for Information Interchange) permet de représenter un caractère sur **7 bits** :

**Table des 128 caractères ASCII**

PDF : fr en v - d - m	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
002	SP	!	"	#	\$	%	&	'	( )	*	+	,	-	.	/	
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

G est codé \$47  
ce qui donne  
% 0100 0111

11 Exercice : donner le code ASCII de @

# Numérisation

ICN 1ère



## Pixellisation

- **Résolution** : px / pouce (ppp\*)



- **Définition** : px qui composent une image

# Numérisation

ICN 1ère



## Codage RVB\*

	R	V	B
Décimal	0 - 255	0 - 255	0 - 255
Binaire	00000000 - 11111111	00000000 - 11111111	00000000 - 11111111
Hexadécimal	00 - FF	00 - FF	00 - FF

