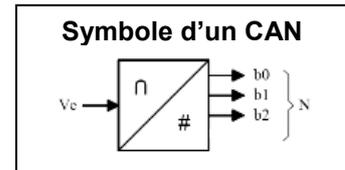
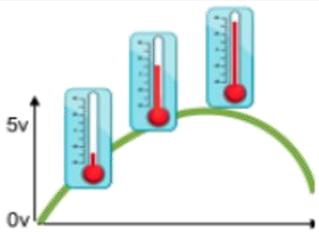
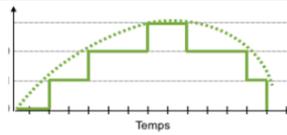
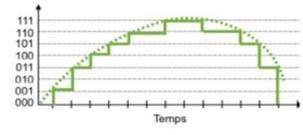


1. Conversion analogique- numérique

Lorsque le signal délivré par un capteur est un signal analogique variable au cours du temps (exemple : une tension), ce signal doit être converti en un signal numérique pour pouvoir être traité par la carte électronique.

Un **convertisseur analogique – numérique (CAN)** (ou DAC pour *Digital Analog Converter*) est un dispositif électronique permettant la conversion d'un signal analogique en un signal numérique.



Signal analogique	Signal numérique codé sur 2 bits	Signal numérique codé sur 3 bits																																																										
	 Soit 4 valeurs possibles : de 0 à 3 <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <thead> <tr style="background-color: #d9e1f2;"> <th>Puissance de 2</th> <th>2¹</th> <th>2⁰</th> </tr> </thead> <tbody> <tr style="background-color: #d9e1f2;"> <th>Décimal</th> <td>2</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Puissance de 2	2 ¹	2 ⁰	Décimal	2	1	0	0	0	1	0	1	2	1	0	3	1	1	 Soit 8 valeurs possibles : de 0 à 7 Exemple : 100 en binaire correspond à 4 en décimal. <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <thead> <tr style="background-color: #d9e1f2;"> <th>Puissance de 2</th> <th>2²</th> <th>2¹</th> <th>2⁰</th> </tr> </thead> <tbody> <tr style="background-color: #d9e1f2;"> <th>Décimal</th> <td>4</td> <td>2</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>2</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>3</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>4</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>5</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>6</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>7</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Puissance de 2	2 ²	2 ¹	2 ⁰	Décimal	4	2	1	0	0	0	0	1	0	0	1	2	0	1	0	3	0	1	1	4	1	0	0	5	1	0	1	6	1	1	0	7	1	1	1
Puissance de 2	2 ¹	2 ⁰																																																										
Décimal	2	1																																																										
0	0	0																																																										
1	0	1																																																										
2	1	0																																																										
3	1	1																																																										
Puissance de 2	2 ²	2 ¹	2 ⁰																																																									
Décimal	4	2	1																																																									
0	0	0	0																																																									
1	0	0	1																																																									
2	0	1	0																																																									
3	0	1	1																																																									
4	1	0	0																																																									
5	1	0	1																																																									
6	1	1	0																																																									
7	1	1	1																																																									

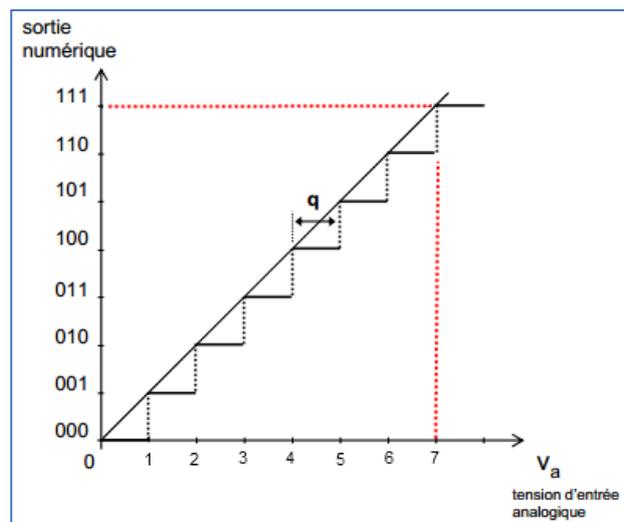
Le **quantum** (q) du convertisseur est la variation minimale du signal d'entrée qui provoque la variation d'une unité de la donnée numérique de sortie.

Le quantum s'exprime dans l'unité de la grandeur analogique d'entrée. Il dépend de l'amplitude du signal d'entrée et du nombre de bits du convertisseur.

$$q = \frac{\text{Amplitude du signal d'entrée}}{2^n}$$

n : nombre de bits du convertisseur

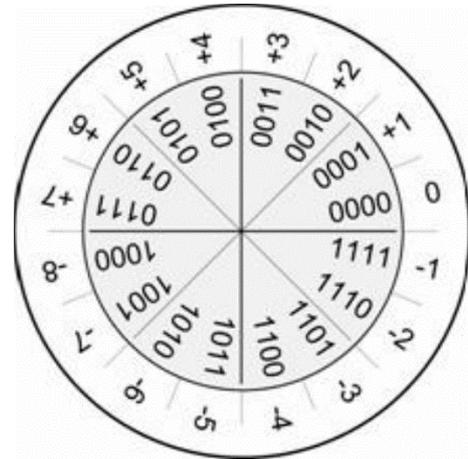
Plus le quantum est petit et plus le signal numérisé est fidèle au signal analogique de départ.



2. Conversion unipolaire / bipolaire

Un CNA peut fournir 2^n niveaux d'amplitude du signal analogique.

- Si la conversion est **unipolaire**, la grandeur de sortie est toujours de même signe et peut donc prendre les valeurs comprises entre 0 et $2^n - 1$.
- Si la conversion est **bipolaire**, la grandeur de sortie peut être négative ou positive, et peut donc prendre les valeurs comprises entre -2^{n-1} et $2^{n-1} - 1$.



La plupart du temps, pour une conversion bipolaire, le signal d'entrée est codé en code complément à deux.

3. Bases de numérations

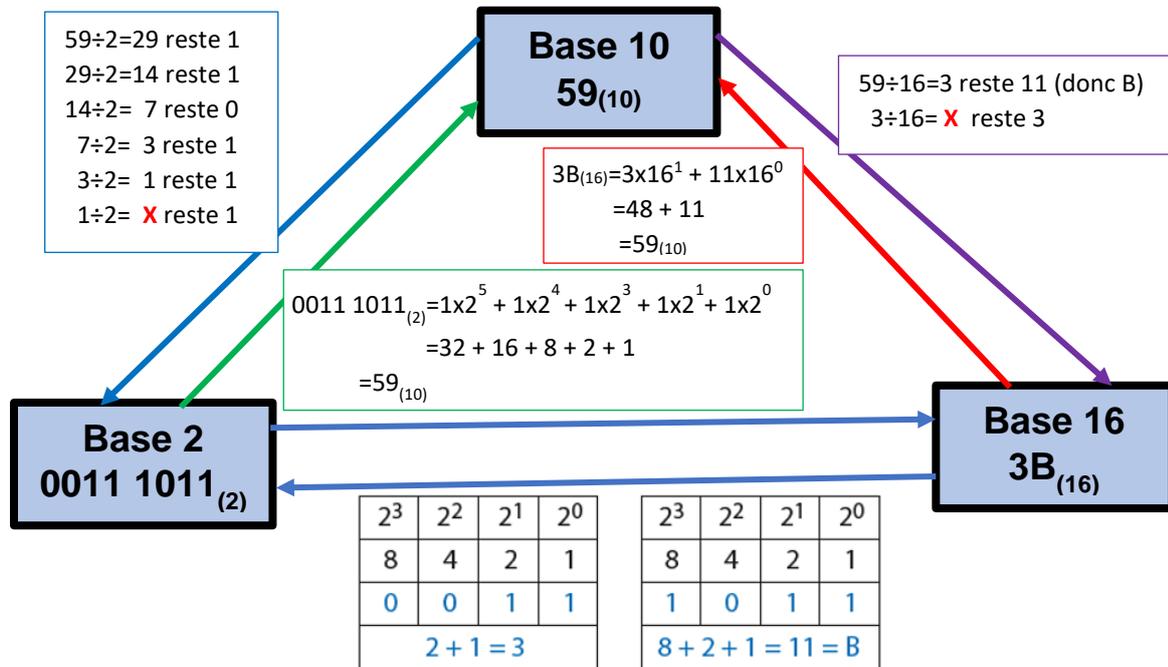
En fonction des besoins, on utilise 3 bases :

- la base 10 que nous utilisons dans la vie de tous les jours ;
- la base 2 utilisée par les ordinateurs ;
- la base 16 qui permet de compacter les mots écrits en base 2 pour les rendre plus lisibles par les humains.

Le tableau ci-dessous permet de lire directement comment sont écrits les 16 premiers nombres dans les 3 bases :

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

Voici, sur un exemple, les méthodes pour passer d'une base à l'autre.



Notation

Lorsqu'on écrit un nombre, il faut spécifier dans quelle base il est écrit, sinon il y a des risques de confusion. Par convention, on indique la base en indice :

- 3A9₍₁₆₎ = (3A9)₁₆ → base 16 : hexadécimal
- 238₍₁₀₎ = (238)₁₀ → base 10 : décimal
- 0100 1101₍₂₎ = (0100 1101)₂ → base 2 : binaire

En programmation, la notation permettant de voir que l'on est en hexadécimal dépend du langage :

Exemple avec (AE4F)₁₆ :

Langage	Préfix	Exemple
C, C++, Java	0x	0xAE4F
Pascal	\$	\$AE4F
Basic	&h	&hAE4F
HTML	#	#AE4F

4. Codages

A. Code ASCII

Le code ASCII (American Standard Code for Information Interchange) est principalement utilisé pour la transmission en série des caractères alphanumériques (Chiffres, lettres, caractères spéciaux) dans les systèmes informatiques.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETH	EOT	ENO	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		"	#	\$	%	&	'	()	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8	€	0	.	/	-	...	†	‡	™	%	§	¶	•	®	©	®
9	0	™	§	¶	•	®	©	®	©	®	©	®	©	®
A	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Chaque caractère en code ASCII est codé sur 8 bits, donc sur 2 caractères exprimés en hexadécimal.

Exemple :

Le caractère « A » est codé en hexadécimal par 41

B. Unicode

Pour coder de manière universelle l'ensemble des symboles utilisés, quelle que soit la langue (Français, Anglais, Grec, Chinois, ...), il faut attribuer à tout caractère ou symbole un nom et un identifiant numérique, et ce de manière unifiée. C'est ce que propose la norme Unicode.

Pour stocker sur un support informatique un texte constitué de caractères Unicode, il faut utiliser un procédé appelé processus d'encodage.

Actuellement un des systèmes d'encodage couramment utilisé est UTF-8.

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001