

DATAMATRIX



Dans cette étude,
nous allons apprendre

- Comment coder une information en utilisant un système de codage
- Quels outils permettent de coder / décoder

1. Pourquoi coder une information ?

Une information peut être amenée à voyager, à être traitée par un ordinateur, ou à être transmise. La chaîne d'information permet d'acquérir une information analogique (grandeur physique évolutive) et de la **numériser**. Manipuler des valeurs numériques est en effet beaucoup plus simple, et plus fiable, que de manipuler la grandeur physique elle-même.

Le problème est de donner une signification aux valeurs numériques. Car afficher des valeurs numériques (suite de 0 et 1) ne permet pas à l'utilisateur de comprendre l'information !

C'est tout l'enjeu du **CODAGE**. Coder, ou décoder, c'est extraire une signification d'une série de valeurs numériques.

Prenons des exemples :



Une image capturée par un appareil photo est **numérisée** pour être stockée sur une carte SD. Il faut alors utiliser un **décodeur** (par exemple le JPEG) pour donner une signification à chaque valeur numérique (position, couleur...)



Un son est **numérisé** par un microphone numérique, puis traité par un système de sonorisation, pour être restitué avec des corrections éventuelles (effacement du bruit de fond, écho, réverbération...). Pour l'entendre, il faut le **décoder** (Code MIDI, MP3...).

Un code consiste à donner une signification (lettre, chiffre...) à une valeur numérique écrite en binaire (suite de 0 et de 1).

Le plus connu est le **code ASCII**.

Le code ASCII (prononcez aski), est une norme informatique pour le codage des caractères. En adoptant le même codage, les systèmes informatiques conçus par n'importe quel fabricant savent ainsi échanger du texte, des nombres, des signes de ponctuation et bien d'autres symboles.

Q 1.1) Que signifie l'acronyme ASCII

Si l'ASCII, né dans les années soixante, est désormais détrôné par d'autres formes de codage plus étoffées et plus récentes comme l'Unicode, il a servi de base à ses successeurs. Ces derniers restent souvent compatibles avec lui. L'ASCII se code sur 7 bits dans la mémoire des ordinateurs, c'est-à-dire dans sept petites "cases élémentaires" ne pouvant contenir chacune que la valeur 0 ou 1, ce qui, en binaire, permet de représenter 128 valeurs (de 0000000 à 1111111).

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figure 1 : table d'encodage ASCII

Les 32 premiers codes, de 0 à 31, ne sont pas des caractères imprimables mais des caractères "de contrôle".

Q 1.2) Que représente le code 13₍₁₀₎, dans la table ASCII.

Q 1.3) Convertir 1010101₍₂₎ en décimale et en hexadécimale. À quel caractère cela correspond-t'il en ASCII ?

Le code ASCII ayant été établi par des Américains, dont la langue ne comporte pas d'accents, les caractères accentués du français et les signes diacritiques nécessaires à d'autres langues n'y étaient pas prévus. C'est ce qui a conduit les organismes de normalisation à définir d'autres codages plus étoffés, qui ajoutent à l'ASCII les caractères qui manquent à une langue, un pays ou une culture.

Par commodité, et comme les 128 premiers caractères restent souvent identiques à ceux de l'ASCII, on parle alors d'ASCII étendu. Cette appellation un peu fourre-tout recouvre en fait différents types de codage, comme la norme ISO-8859-1, qu'on appelle aussi Latin-1 ou Europe occidentale.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
Ax	NBSP	ı	ç	£	¤	¥	ı	§	¨	©	ª	«	–	_	®	ˆ
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure 2: table de codage ISO-8859-1 ou ASCII « étendu »

Q 1.4) Sur combien de bits se code la norme ISO-8859-1 ? En déduire le nombre de caractères disponibles.

Q 1.5) Selon la norme ISO-8859-1, quel est le code du caractère « U » en hexadécimale ? Convertir cette valeur en décimale. Que remarquez-vous, par rapport à l'encodage en ASCII ?

Q 1.6) Selon la norme ISO-8859-1, quel est le code du caractère « é » en hexadécimale ? Convertir cette valeur en binaire. Que peut-on en conclure sur le nombre de bits de cet encodage ?

2. Les datamatrix

Un datamatrix est une petite image en noir et blanc, qui comporte une information codée. Il s'agit d'un code barre 2D. Le gros avantage de ce type d'image est qu'elle peut être lue en une fraction de seconde par un lecteur industriel (flasher).

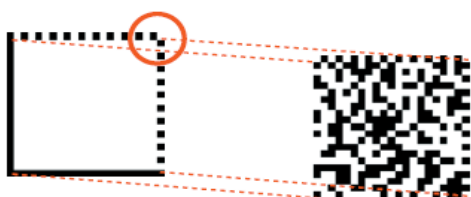


Depuis le 1^{er} janvier 2011, par exemple, tous les médicaments vendus en France comportent sur leur emballage un datamatrix permettant de tracer la provenance, la date de péremption, et le nom du médicament.



Les Datamatrix font l'objet de plusieurs normes. Nous allons ensemble découvrir ceux répondant à la norme ISO/IEC 16022 dite ECC200. Ces datamatrix permettent de coder les caractères alphanumériques en utilisant précisément le code **ASCII « étendu »**.

Composition d'un Datamatrix :

Motifs de repérage
(finder pattern)

Matrice de données

Un datamatrix est composé d'une **marge** pour se repérer, et d'une **matrice** permettant d'y insérer les données sous forme de code. Par exemple, vous pouvez grâce à un datamatrix coder votre carte de visite, en donnant vos noms, adresse, téléphone, e-mail... La taille de la matrice dépend du nombre d'informations que l'on souhaite coder. La capacité peut aller jusqu'à 2335 caractères alphanumériques codés dans une matrice.

Pour lire un datamatrix, il suffit de **connaître le code**, et de savoir interpréter sa signification.

Si vous possédez un smartphone (android ou I-phone), vous pouvez télécharger une application qui vous permettra de lire un datamatrix. Cette application est disponible sur <https://www.kaspersky.com/gr-scanner>, sinon, vous pouvez demander à votre professeur un lecteur de datamatrix



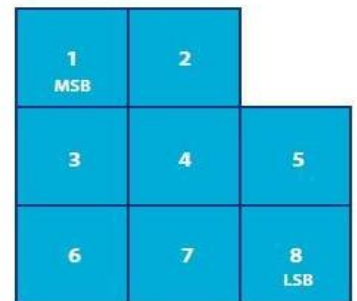
Q 2.1) Lire ce datamatrix et indiquer les informations contenues dedans.

Un datamatrix est composé d'un certain nombre de **codewords**. La traduction du mot codeword vous donne une définition inexacte : « *mot codé* ». Il s'agit en fait d'un **caractère codé**, par exemple une lettre ou un chiffre.

Un codeword = un octet (8 bits) = un caractère

Voici comment sont disposés les codewords dans une matrice 10×10 :

Marge									
2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5		
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8		
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2		
Marge	1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4	Marge
	1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7	
	7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1	
	7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3	
	7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6	
Marge									



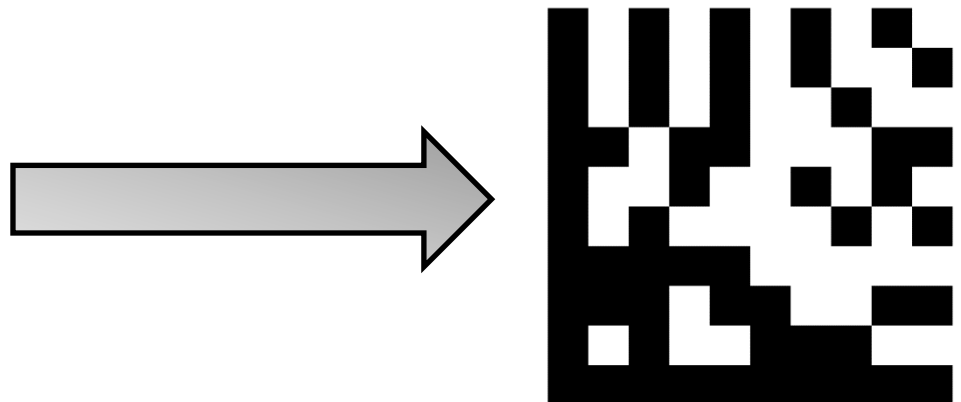
LSB = Least significant bit
MSB = Most significant bit

8 codewords sont ici présents. On peut remarquer que certains codeword sont regroupés dans une seule zone (codeword 2, 5, 6, et 8) alors que d'autres sont fractionnés en plusieurs zones (codeword 1, 3, 4, 7).

Le coin en haut à gauche est le premier bit du deuxième codeword. Chaque bit peut être égal à 0 ou à 1.

Cela peut par exemple donner ceci :

1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	0	1
1	0	1	0	1	0	0	1	0	0
1	1	0	1	1	0	0	0	1	1
1	0	0	1	0	0	1	0	1	0
1	0	1	0	0	0	0	1	0	1
1	1	1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0	1	1
1	0	1	0	0	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1



Le décodage d'un Datamatrix s'effectue ainsi :

1. Lire la valeur numérique en binaire du codeword
2. Calculer la valeur décimale
3. Décoder grâce au code ASCII pour trouver la signification de la valeur décimale.

Q 2.2) Décoder le datamatrix précédent en complétant le tableau suivant avec la valeur en binaire, en hexadécimale puis le caractère associé des trois premiers codewords.

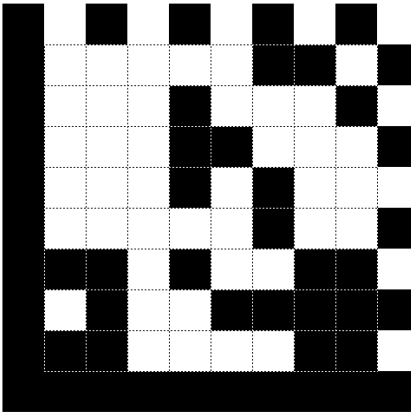
Codeword 1								Codeword 2								Codeword 3							
Binaire								Binaire								Binaire							
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
↓								↓								↓							
Hexadécimale								Hexadécimale								Hexadécimale							
↓								↓								↓							
Caractère								Caractère								Caractère							

Q 2.3) Coder « i2d », en vous aidant des tableaux suivants.

Codeword 1								Codeword 2								Codeword 3							
Caractère								Caractère								Caractère							
i								2								d							
↓								↓								↓							
Hexadécimale								Hexadécimale								Hexadécimale							
↓								↓								↓							
Binaire								Binaire								Binaire							
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8

Q 2.4) Compléter en inscrivant 0 ou 1 (à gauche) et en noirissant (à droite) les cases des 3 codewords « i2d ».

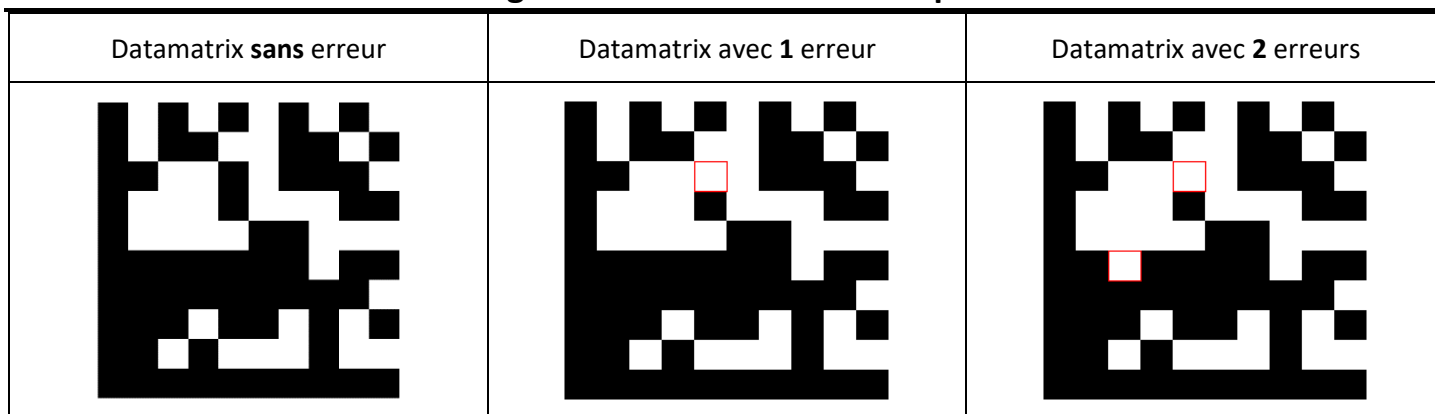
1	0	1	0	1	0	1	0	1	0											
1										1	1	0								1
1				1	0	0	0	1												0
1				1	1	0														1
1		0	0	1	0	1														0
1		0	0	0	0	1														1
1	1	1	0	1	0	0	1	1												0
1	0	1								1	1	1	1							1
1	1	1								0	1	1								0
1	1	1	1	1	1	1	1	1	1											1



Q 2.5) Scanner le datamatrix précédent et indiquer les caractères qu'il renvoie.

Sur les 8 codewords que comporte une matrice 10*10, 5 sont réservés à la vérification de l'information pour **éviter les erreurs de lecture**. Nous avons ignoré cette vérification volontairement. Elle est basée sur un algorithme de REED-SOLOMON.

Q 2.6) Scanner les datamatrix suivants. Que pouvez-vous en conclure.



Q 2.7) À l'aide du générateur de datamatrix <http://datamatrix.kaywa.com/>, compléter le tableau suivant donnant le pourcentage de caractères encodables en fonction du nombre de codewords de la matrice. Que remarquez-vous ?

Taille de la matrice	Nombre de codewords	Nombre maximal de caractères encodables	%
10×10	8	3	37,5 %
12×12			
14×14			
16×16			

3. Conclusion :

Les **datamatrix** sont des images très simples qui permettent de coder des informations (texte, série de chiffres, date...). Des images complètes, avec les couleurs, sont beaucoup plus difficiles à coder, mais le principe est le même.

Le format **BMP**, par exemple, permet de coder une image en utilisant de 8 à 24 bits pour chaque pixel d'écran (comme si un codeword permettait de coder un pixel). La palette de couleur est d'autant plus riche que le nombre de bits est important.

Q 3.1) Quel est le poids informatique (en octets) d'une image de 1024 × 720 pixels codés chacun sur 8 bits ? Et si on codait chaque pixel sur 24 bits ?