

# DOSSIER TECHNIQUE

## SYSTÈMES DE LECTURE DE CODES

### Comporte les documents suivants :

- Les codes DATA Matrix
- Le code ASCII
- Décodage d'un Data Matrix

page 2 à 7

page 8

page 9 à 11

# Les codes DATA MATRIX

## Introduction au marquage direct de Data Matrix

---

### Les données du problème en quelques mots

L'industrie identifie depuis longtemps ses produits à l'aide de codes à barres unidimensionnels (ID) imprimés sur des étiquettes. Cette identification suit le produit tout au long de la chaîne logistique.

Aujourd'hui de nombreux industriels étendent le suivi du produit : désormais la traçabilité suit un produit depuis sa fabrication jusqu'à la fin de son utilisation, et la réglementation impose une identification permanente et fiable sur des produits hauts de gamme sujets aux vols et à la contrefaçon.

Les codes à barres ID marqués sur étiquettes ne répondent plus aux demandes actuelles de traçabilité et sont remplacés par le marquage direct sur pièces (DPMI) de codes 2D (codes en 2 dimensions).

DPMI = Direct Part Mark Identification.

Le marquage direct sur pièce de codes 2D s'impose désormais :

Les étiquettes sont bannies, car elles ne répondent pas aux contraintes de suivi sur le cycle de vie du produit.

Le code à barres ID ne permet pas le stockage de grandes quantités d'informations.

Le code ID n'est pas robuste : un endommagement partiel du code le rend totalement illisible.

La surface nécessaire au marquage d'un code à barres est souvent rédhibitoire.

Le code 2D stocke davantage d'informations sur une petite surface et offre une meilleure robustesse grâce aux algorithmes de corrections d'erreurs. Son marquage direct sur la pièce rend le code indissociable du produit.

### Une solution en 3 points clés

- Le Data Matrix s'impose largement dans le domaine du marquage direct des codes 2D.
- Le gravage par laser constitue une solution satisfaisante aux demandes de DPMI.
- Les solutions de vision industrielle disponibles aujourd'hui permettent la relecture des codes gravés directement sur pièces dans les conditions les plus difficiles : codes minuscules, faibles contrastes, états de surface irréguliers, endommagements partiels du code.

En résumé une solution d'identification d'un produit se résume par :

- Le marquage direct sur pièce au laser d'un code Data Matrix.
- Un système de relecture adapté aux contraintes de marquage.
- Une application de DPMI doit être considérée dans sa globalité : marquage et relecture étant intimement liés.

### Domaines d'application

Le marquage direct de Data Matrix est utilisé dans de nombreux domaines : Automobile, Aéronautique, Aérospatial, Electronique, Médical....

### Qu'est-ce qu'un code Data Matrix ?

Code à 2 dimensions, par opposition aux codes à barres ID (code 39, code 128, etc....).

Exemple de code à barres unidimensionnel EAN128 :



Exemple de code Data Matrix ECC200 :



Ces 2 codes contiennent les mêmes données : Data Matrix ECC200 Laser Cheval

Les informations sont stockées dans une matrice de cellules :

Data Matrix ⇔ Matrice de données.

L'algorithme d'encodage combine les données utiles avec des codes de correction d'erreur, ce qui rend le code plus robuste face à une altération.

Divers formats de codage existent pour le Data Matrix : ECC000, ECC050, ECC080, ECC100, ECC140 et ECC200.

Les normes et directives industrielles privilégient le code ECC200 ( ECC = Error Checking and Correcting).

## Caractéristiques du code Data Matrix ECC200

Les données sont stockées sous forme numérique dans une matrice carrée ou rectangulaire :

- Possibilité de coder jusqu'à 3116 chiffres ou 2335 caractères alphanumériques ou 1555 octets.
- Codes carrés, de 10x10 jusqu'à 144x144 cellules.
- Codes rectangulaires de 8x18 jusqu'à 16x48 cellules.
- Codes de correction d'erreur, jusqu'à 50% d'informations de correction dans le code.

Les dimensions d'un code et son orientation ne sont pas figées. Les seules limites étant la résolution du procédé de marquage, l'état de surface du matériau et les performances du système de lecture.

Par exemple, des dimensions inférieures au mm peuvent être atteintes par un système de marquage par laser.

La structure du code et les propriétés des systèmes de relecture permettent de localiser un code et de déterminer ses caractéristiques quelque soit son orientation.

Un code partiellement endommagé peut être reconstitué grâce aux informations de correction d'erreur.

## 1-1 Caractéristiques des codes 2D

### • Grande capacité de données

Les codes-barres contiennent des données dans une seule direction tandis que les codes 2D contiennent des données dans les deux directions, horizontale et verticale. Ils peuvent donc renfermer beaucoup plus de données que les codes-barres classiques. Les codes-barres contiennent jusqu'à 30 caractères, tandis que les codes 2D peuvent en contenir jusqu'à 3000.



### • Forte densité de données (gain de place)

La taille des codes 2D peut représenter 1/30 de la taille d'un code-barres classique contenant les mêmes données.

Par conséquent, les codes 2D peuvent être imprimés sur du matériel électronique et autres pièces de petite taille où l'espace est limité.

### • Correction d'erreur / restauration des données

Les codes 2D comportent une fonction intégrée de correction d'erreur, leur permettant de restaurer les données si le code est endommagé ou souillé. Une correction d'erreur mathématique (Reed-Solomon) est employée pour la restauration des données.

## ◆ Inconvénient des codes 2D

### • Les codes 2D ne disposent pas de données de secours lorsque les données sont illisibles

Les codes-barres disposent généralement de caractères lisibles en bas au cas où certaines données seraient endommagées/manquantes. Les opérateurs peuvent ainsi lire les caractères et saisir les données à l'aide d'un clavier sans gêner le déroulement des opérations.

Les codes 2D contenant une très grande quantité de données, aucun caractère lisible n'est ajouté.

Lorsque le code 2D est trop endommagé pour être lu, il n'existe aucun moyen de lire les données, ce qui a pour effet de perturber le déroulement des opérations. (Il est possible d'ajouter des caractères lisibles aux codes 2D, toutefois il n'est pas réaliste de demander au personnel de taper plus d'une centaine de caractères.)

Lorsque vous utilisez des codes 2D, vous devez créer un système permettant de fournir des mesures lorsque le code 2D est endommagé.



Caractères lisibles à l'œil nu

Le personnel peut vérifier et saisir les données

## 1-2 Différents types de codes 2D

Les codes 2D sont classés en deux types en fonction de la structure.

### • Type empilé

Les codes-barres classiques sont empilés verticalement.

Exemple



PDF417



Code 49

Le type empilé est composé de plusieurs codes-barres empilés verticalement dans un rectangle.

Les lecteurs laser peuvent lire le type empilé si le laser croise l'ensemble des codes-barres empilés. Toutefois, il ne peut pas tolérer de déplacement supérieur à  $\pm 10^\circ$ .

(Seuls les lecteurs laser dotés d'un logiciel permettant de lire les codes 2D peuvent lire les codes 2D.)

### • Type matriciel

Les données sont composées de modules en noir et blanc formant un motif complexe.

Exemple



Code QR



Data Matrix



Veri Code

Le type matriciel est composé de petits carrés ou points appelés « modules » disposés dans une grille carrée. Ils possèdent un cadre carré, un cadre en forme de L ou des motifs de détection de position permettant une détection fiable de la position. Une caméra ou un lecteur intégré avec dispositifs CCD 2D décode la disposition des modules par traitement de l'image. Le code peut être lu dans n'importe quel sens.

## 1-3 Application des codes 2D

### 1. Contrôle des très petites pièces

#### I Data Matrix, code QR, Veri code (Types)

Les très petites pièces utilisées dans l'industrie des écrans LCD, de l'électronique, des semi-conducteurs et de l'automobile nécessitent plusieurs douzaines de caractères pour le contrôle du suivi de fabrication. Comme ces données doivent être compactes afin de pouvoir être imprimées sur ces petites pièces, des codes 2D matriciels sont souvent utilisés.

Les codes Data Matrix et Veri Code sont utilisés dans l'industrie des écrans LCD et les codes Data Matrix et QR sont généralement utilisés pour les cartes imprimées et les pièces électroniques.

## 2. Notification d'expédition, facturation et étiquetage de produits au moyen de données EDI I Code QR, PDF417

### (Types)

Lorsqu'une base de données ou d'autres informations ne sont pas disponibles pour un article, un code 2D peut fournir des informations très utiles concernant l'identification du produit. L'Association des constructeurs automobiles japonais utilise les codes PDF417 et QR.

## 3. Usage gouvernemental I PDF417

### (Type)

Les codes 2D sont généralement utilisés par les autorités pour éviter la falsification. Au Japon, le code PDF417 a été utilisé pour les billets d'entrée au stade olympique de Nagano. Il est largement utilisé pour les permis de conduire ou les cartes d'identité aux États-Unis car il peut coder un portrait du visage pour garantir une sécurité optimale. Il est également largement employé pour les passeports, cartes d'identité, cartes d'assuré ou formulaires douaniers en Asie du Sud-Est, au Moyen et Proche-Orient, en Afrique et en Amérique du Sud pour éviter la falsification.

## 4. Tri ou suivi des livraisons

### I Code QR, Maxi Code (Types)

Les codes 2D sont utilisés pour le tri automatique à grande vitesse ou le suivi des livraisons dans les systèmes de distribution. Dans ce domaine, ce ne sont pas les informations mais la vitesse qui compte. Les codes Maxi Code et QR répondent à ce besoin.

## 5. Usage médical I PDF417

### (Type)

La directive sur la nouvelle désignation du code des médicaments sur ordonnance spécifie de décrire les codes des produits, dates d'expiration, numéros de production, quantités de bioproduits spécifiques et médicaments injectables. Par exemple, concernant les comprimés et les gélules, ces informations sont imprimées sur une tablette ou une bouteille contenant les emballages individuels. Pour les médicaments injectables, l'impression est réalisée sur chaque emballage de dose (unité de dosage minimale produite et vendue par un fabricant) tel qu'une ampoule ou un flacon ou sur chaque emballage commercial (unité d'emballage minimale vendue par un grossiste à des organismes médicaux, etc.).

En cas d'impression sur un emballage de dose ou un emballage commercial ne présentant qu'un espace limité pour imprimer un code, un symbole composite est utilisé. À l'intérieur du symbole composite, un code de production est imprimé au format GS1 DataBar et les informations variables telles qu'un numéro de production, une date d'expiration ou une quantité sont imprimées au format MicroPDF417.

## 2-1 Structure du code Data Matrix

Le code Data Matrix est un code 2D matriciel développé par ID Matrix en 1987. Il a été enregistré sous la norme ISS de l'AIMI en 1996 et la norme ISO/CEI en 2000.

Spécifications	
Taille minimale	10 x 10 modules
Taille maximale	144 x 144 modules
Capacité de données maximale	Numérique : 3116 caractères
	Alphanumérique : 2335 caractères

Les versions plus anciennes du code Data Matrix comprennent les versions ECC000, ECC050, ECC080, ECC100 et ECC140. ECC200 est la dernière version du code Data Matrix et peut avoir une configuration carrée ou rectangulaire.

### • **ECC000, ECC050, ECC080, ECC100, ECC140**

Ces versions sont composées de nombres impairs de modules allant de 9 x 9 à 49 x 49 en utilisant une correction de convolution.

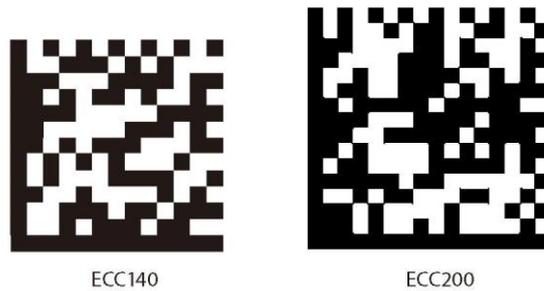
Étant donné qu'une légère distorsion perturbe la lecture lorsque les données sont importantes, ces versions plus anciennes ne sont presque jamais utilisées.

### • **ECC200**

ECC200 possède une plus grande capacité de correction d'erreur pour éliminer les problèmes de distorsion. Le code Reed-Solomon est utilisé pour la correction d'erreur afin de restaurer les données lorsqu'une partie d'un code est endommagée.

La version ECC200 est internationalement normalisée. En règle générale, c'est la version ECC200 qui est utilisée lorsque le code Data Matrix est requis.

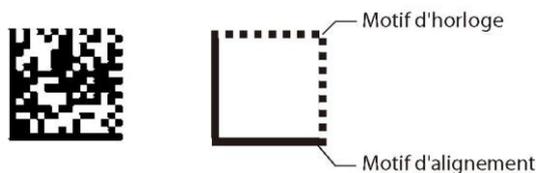
La différence entre les anciennes versions (ECC000, ECC050, ECC080, ECC100, ECC140) et la nouvelle version (ECC200) réside dans le nombre de modules par côté. Les anciennes versions possèdent des nombres impairs de modules tandis que la nouvelle version possède un nombre pair de modules.



## 2-2 Structure du code Data Matrix (ECC200)

• **Motif d'alignement et motif d'horloge**

La zone de données du code Data Matrix est entourée d'un cadre en L appelé motif d'alignement et de lignes pointillées appelées motif d'horloge. Les lecteurs capturent ces motifs pour déterminer la position du code par traitement de l'image. Le code Data Matrix peut ainsi être lu dans n'importe quel sens.



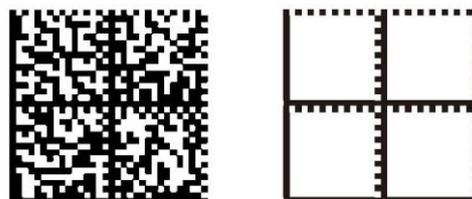
Lorsque le nombre de modules est supérieur à 24 x 24, le code est divisé en blocs qui ne dépassent pas 24 modules par côté. Cette structure empêche la distorsion du code.

• **Taille du code et nombre de blocs**

Il existe vingt-quatre tailles de code, allant de 10 x 10 modules à 144 x 144 modules (dont six tailles pour le type rectangulaire).

Lorsqu'un code possède plus de 26 x 26 modules (plus de 24 x 24 modules pour les données), il est divisé en blocs, comme indiqué ci-dessous, qui ne dépassent pas 24 modules par côté. Cette structure empêche la distorsion du code.

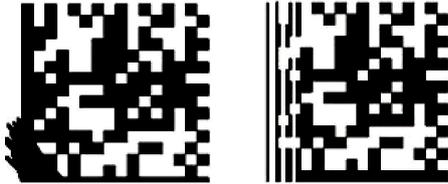
Taille du symbole	Bloc	Cellule de données
10x10 à 26x26	1	8x8 à 24x24
28x28 à 52x52	4	14x14 à 24x24
64x64 à 104x104	16	14x14 à 24x24
120x120 à 144x144	36	18x18 à 22x22



### 3 Pourquoi est-ce que le code 2D n'a pu être lu alors qu'il n'est que partiellement abîmé?

L'alignement du motif est peut-être abîmé.

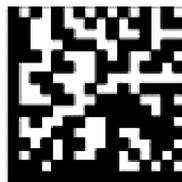
#### EXEMPLES DE PROBLÈME



La lecture des codes DataMatrix commence par la localisation du cadre en L appelé motif d'alignement. Si ce motif d'alignement est abîmé, le code ne peut pas être lu.

#### EXPLICATION

La zone de codage des données des codes DataMatrix est entourée d'un cadre en L appelé motif d'alignement. Le lecteur capture ce motif pour déterminer l'orientation du code et l'endroit où le décodage doit commencer. Par conséquent, si le motif d'alignement est abîmé, le code devient illisible.



DataMatrix



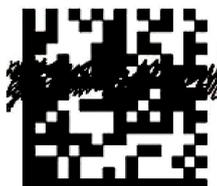
Motif d'alignement

#### LA MEILLEURE SOLUTION !

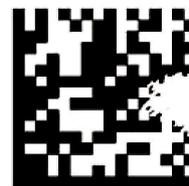
Les codes DataMatrix sont dotés d'une fonction appelée "correction des erreurs". Ceci permet de restaurer et de lire des données si la taille de la partie abîmée le permet. Toutefois, la correction des erreurs est inefficace si le motif d'alignement est abîmé ou usé.



Point



Tache



Données manquantes

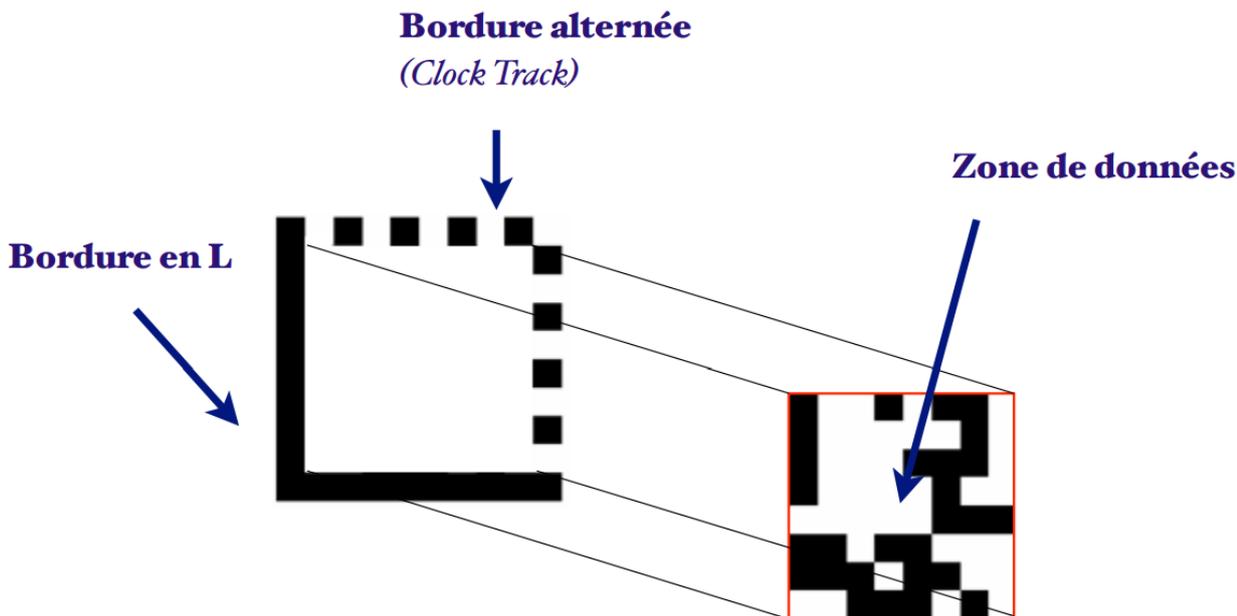
# Le code ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>;</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: [www.LookupTables.com](http://www.LookupTables.com)

## Décodage d'un code Data Matrix

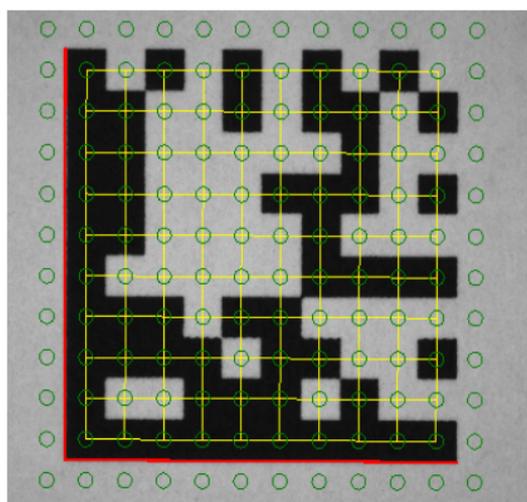
La bordure extérieure du code est immuable, composée de la bordure en L et de la bordure alternée, qu'on retrouvera donc dans chaque Data Matrix. La matrice intérieure correspond à la zone des données et sa taille de matrice est .... N-2 x N-2 bien sûr. 8 x 8 dans cet exemple.



### Les étapes du décodage

Le décodage d'un Data Matrix se fait en plusieurs étapes.

La première consiste à prendre une « photo » du code, ensuite l'algorithme de décodage se met en route. On localise d'abord la bordure en L, deux segments qui se rejoignent à angle droit.



On localise ensuite les modules de la bordure alternée, puis on trace des droites passant par les centres de ces modules et parallèles à la branche de la bordure en L adjacente.

On trace ainsi la grille de lecture du code (en jaune sur ce graphique) dont les points d'intersection sont centrés sur les modules du Data Matrix

Maintenant que la grille de lecture a été tracée, on peut passer au décodage de la zone des données, et le principe est très simple. Il faut déterminer pour chaque module si il est de couleur claire ou sombre.

Notez qu'on n'a pas toujours des Data Matrix ou des QR imprimés en encre noire sur un fond blanc comme dans cet exemple. Donc on fait référence généralement à des modules clairs et sombres. Ce qui est important par contre c'est d'avoir un contraste suffisant entre les deux types de modules.

Découpage et position des codewords d'une matrice 8 x 8.

2.1 = premier module du second codeword

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
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
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

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

10001110 - 10100100 - 10111010 - 01110010 - 00011001 - 00000101 -  
01011000 - 01100110

123456 puis 5 codewords de correction d'erreur

Source : Guide GS1 Data Matrix ECC 200

Prenons le premier codeword de données: 10001110

On convertit la valeur binaire du codeword en valeur décimale.

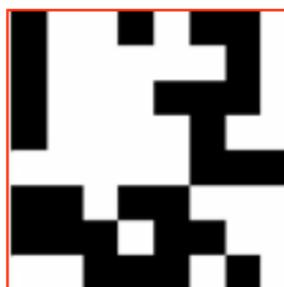
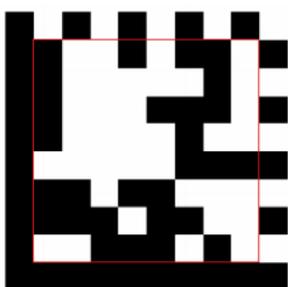
Pour la valeur binaire 10001110 cela donne:  $1x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 0x2^0 = 142$

On se conforme ensuite à la table ci-dessous pour déterminer la valeur de la donnée: Pour un codeword entre 130 et 229, on soustrait 130 à la valeur décimale  $142 - 130 = 12$

Un petit exercice: Faites de même avec les codewords 10100100 et 10111010 pour retrouver les données 34 et 56.

Aux modules sombres on donnera la valeur '1' et au module clairs la valeur '0'. Ces valeurs binaires sont regroupées et ordonnées par groupes de huit (les codewords) selon un découpage particulier spécifié dans le standard ISO du Data Matrix (pour référence ISO/IEC 16022). Regardons ensemble l'exemple du code Data Matrix représentant les données 123456.

Le Data Matrix permet d'encoder les chiffres par paires, donc on utilise les 3 premiers codewords pour encoder successivement 12, 34 et 56. On pourrait aussi bien sûr encoder les 6 chiffres séparément mais cela mobiliserait six codewords au lieu de trois et nous ferait perdre un peu de place. Autant optimiser l'encodage ! Ils nous reste encore 5 octets qui vont être utilisés pour ajouter des jokers... plus précisément des codewords de correction d'erreur. On y reviendra.



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

Codeword	Data or function
1-128	ASCII data (ASCII value + 1)
129	Pad
130-229	2-digit data 00-99 (Numeric Value + 130)
230	Latch to C40 encodation
231	Latch to Base 256 encodation
232	FNC1
233	Structured Append
234	Reader Programming
235	Upper Shrit (shift to Extended ASCII)
236	05 Macro
237	06 Macro
238	Latch to ANSI X12 encodation
239	Latch to Text encodation
240	Latch to EDIFACT encodation
241	ECI Character
242-255	Not to be used in ASCII encodation

Les codewords de 1 à 128 permettent d'encoder les caractères de la table ASCII, on utilise les codewords de 130 à 229 pour encoder les 100 paires de chiffres de 00 à 99.

Données provenant de **MATHÉMATIQUEMENT VÔTRE N°13 / AVRIL-JUIN 2013**