Code QR

Table des matières

Introduction

Le code QR (en anglais QR Code) est un type de code-barres en deux dimensions (ou code matriciel) constitué de modules noirs disposés dans un carré à fond blanc
L'agencement de ces modules définit l'information que contient le code QR
QR est l'abréviation de l'anglais Quick Response ce qui signifie que le contenu du code peut être décodé rapidement après avoir été lu par un lecteur de code-barres, un téléphone mobile, un smartphone, ou encore une webcam
Son avantage est de pouvoir stocker plus d'informations qu'un code à barres, et surtout des données directement reconnues par des applications *Wikipédia*

Le code QR a été créé par l'entreprise japonaise Denso-Wave en 1994 pour suivre le chemin des pièces détachées dans les usines de Toyota. Il est rendu public en 1999 sous licence libre : cela a contribué à la diffusion du code au Japon. Par la suite, il prend un réel essor avec l'avènement des smartphones. À la fin des années 2000, il devient l'un des codes bidimensionnels les plus populaires dans le monde, et les applications de lecture de codes QR sont souvent déjà installées par les fabricants dans les téléphones mobiles *Wikipédia*

L'étude qui suit a été réalisée à partir de la Norme *NF ISO/IEC 18004 du 11 avril 2015* intitulée :
Technologies de l'information — Technologie d'identification automatique et de capture des données — Spécification de la symbologie de code à barres Code QR

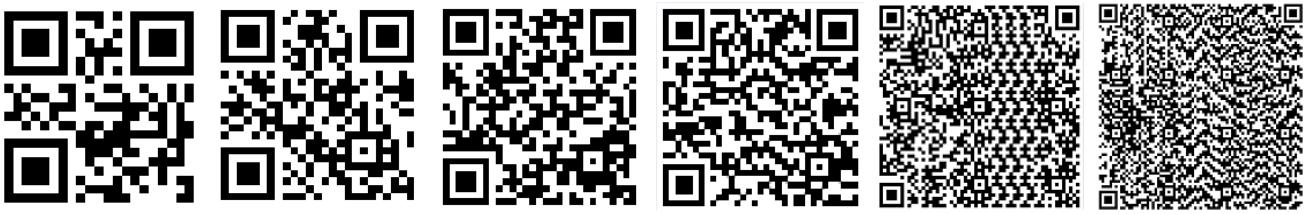
Représentation des données

Dans un symbole Code QR, un module sombre est un « 1 » logique et un module clair est un « 0 » logique

Taille du symbole

La taille d'un symbole de Code QR va de 21 × 21 modules à 177 × 177 modules (zone de silence de largeur 4 modules non comprise), correspondant aux versions 1 à 40, augmentant par incréments de quatre modules par côté : 21-25-29-33...173-177

Différentes versions du Code QR

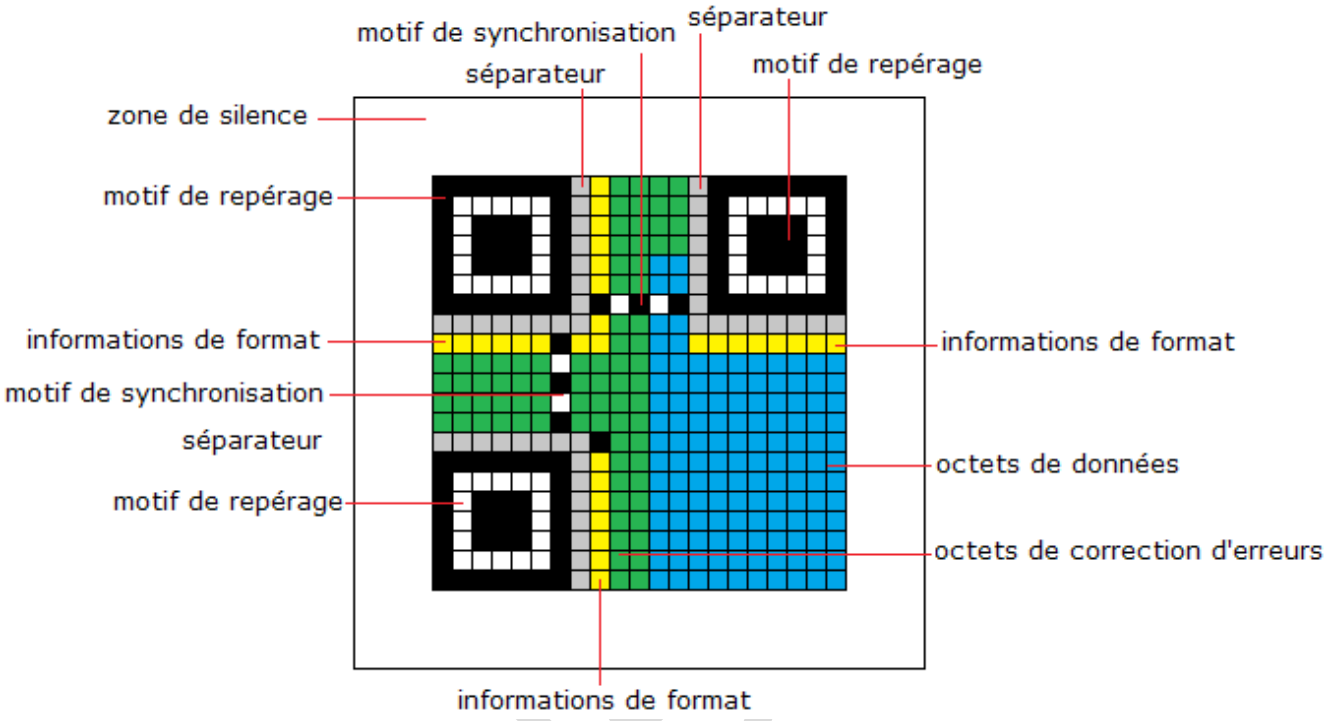


Version 1	Version 2	Version 3	Version 4	Version 10	Version 15
21×21	25×25	29×29	33×33	57×57	77×77
10-25 caractères	20-47 caractères	35-77 caractères	50-114 caractères	174 à 395 caractères	321 à 758 caractères

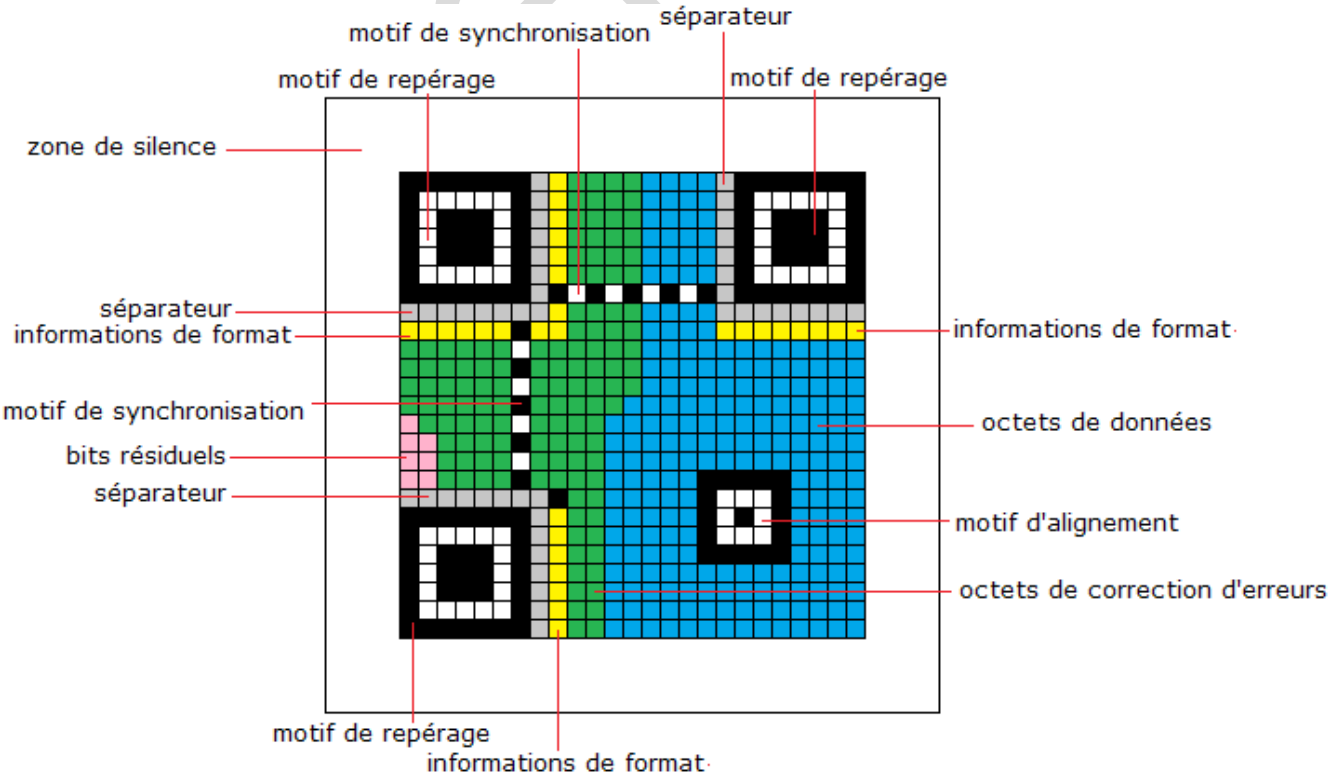
Lecture possible des QR Codes précédents avec l'application Android QR Code Reader

Structure d'un symbole

Symbole de version 1 et de niveau de correction d'erreurs M



Symbole de version 2 et de niveau de correction d'erreurs M



Correction d'erreurs

Codage Reed-Solomon

Les codes QR utilisent le système Reed-Solomon pour la correction d'erreur sur les données

Le code contient jusqu'à 30% de redondance, ce qui signifie qu'il peut être décodé même si 30% des données sont illisibles

Capacité à corriger les erreurs :

- Niveau L (Low) : environ 7 % de redondance
- Niveau M (Medium) : environ 15 %
- Niveau Q (Quartile) : environ 25 %
- Niveau H (High) : environ 30 %

Irving Reed et Gustave Solomon

version	octets de données	niveau de correction d'erreurs	octets de correction d'erreurs
1	19 16 13 9	L M Q H	7 10 13 17
2	34 28 22 16	L M Q H	10 16 22 28

Remarques:

- la version 1 comprend 26 octets (19+7 ou 16+10 ou 13+13 ou 9+17)
- la version 2 comprend 44 octets (34+10 ou 28+16 ou 22+22 ou 16+28)

Utilitaire de calcul du code correcteur RS

Codage BCH

C'est un code correcteur utilisé pour corriger des erreurs aléatoires
BCH sont les initiales des inventeurs : Bose, Chaudhuri, Hocquenghem

Raj Chandra Bose

Dijen Kumar Ray-Chaudhuri

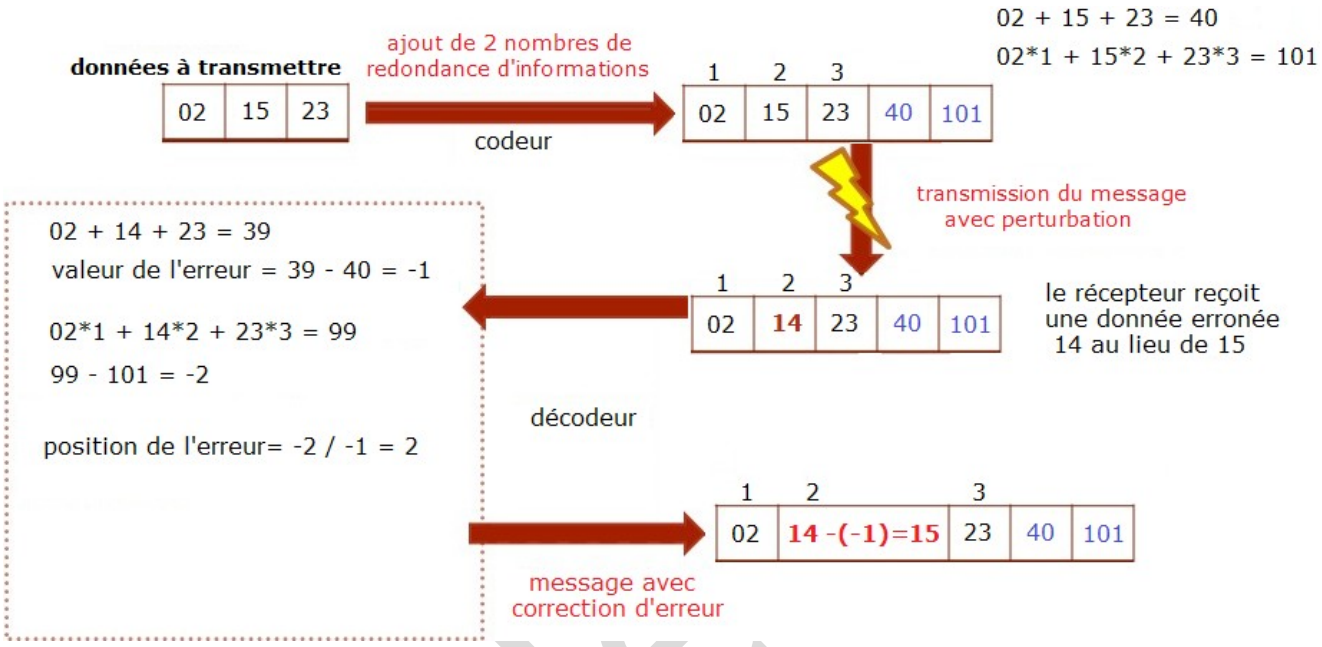
Alexis Hocquenghem
(1908-1990)
est un mathématicien français
connu pour sa découverte
en 1959
des codes correcteurs d'erreurs

photo non disponible

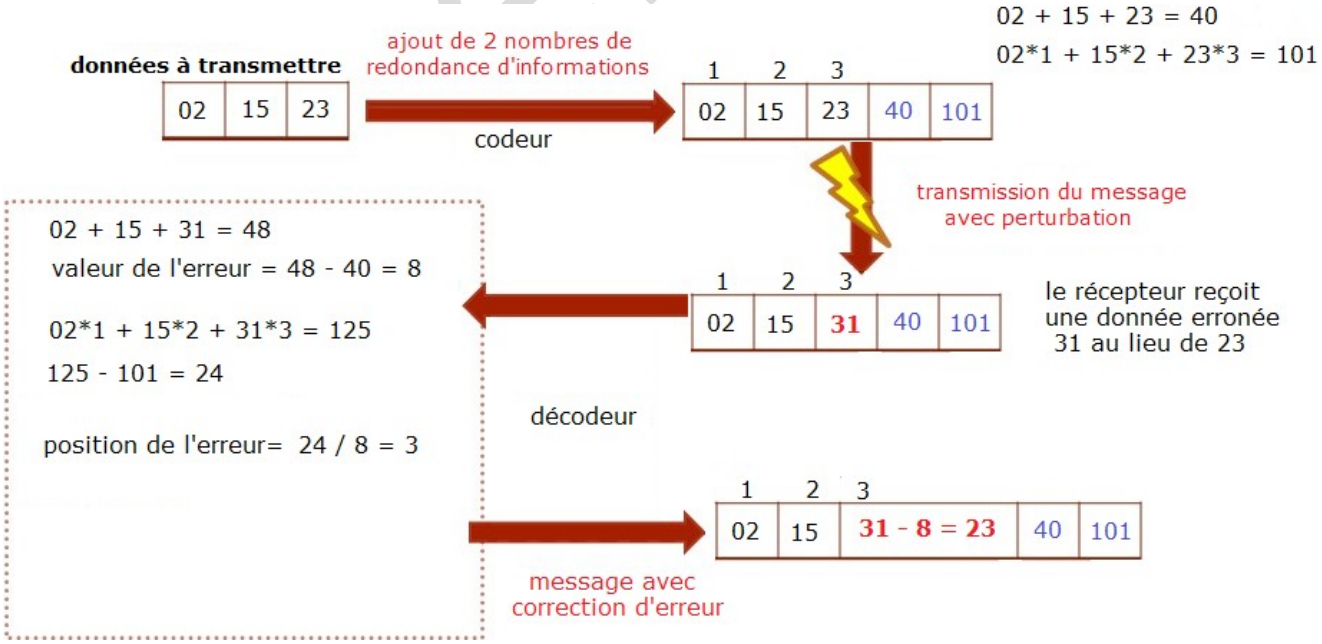
Ce code est utilisé pour la correction des données de format du symbole
Utilitaire de calcul du code correcteur BCH

Principe d'une correction d'erreur

Exemple 1



Exemple 2



Différents modes de codage

ECI, numérique, alphanumérique, octet, Kanji, mélange, adjonction structurée, FNC1

Le mode numérique code les données à partir du jeu de chiffres décimaux (0 à 9) (valeurs d'octet 30_{HEX} à 39_{HEX})

Le mode alphanumérique code les données à partir d'un jeu de 45 caractères:

- 10 chiffres numériques (0 à 9) (valeurs d'octet 30HEX à 39HEX)
- 26 caractères alphabétiques (A à Z) (valeurs d'octet 41 à 5A)
- 9 symboles (SP, $, %, *, +, -, ., /, :) (valeurs d'octet 20_{HEX}, 24_{HEX}, 25_{HEX} , $2A_{HEX}$, $2B_{HEX}$, $2D_{HEX}$ à $2F_{HEX}$,$3A_{HEX}$)

4 bits qui définissent le mode :

ECI	numérique	alpha numérique	octet	Kanji	adjonction structurée	FNC1	motif de terminaison
0111	0001	0010	0100	1000	0011	1) 0101 2) 1001	0000

Capacité de quelques versions du Code QR

version	modules /côté	modules de motifs de fonctions	modules d'informations de format et de version	modules de données	octets de données	bits résiduels
1	21	202	31	208	26	0
2	25	235	31	359	44	7
20	97	659	67	8683	1085	3
30	137	1219	67	17483	2185	3
40	177	1614	67	29648	3706	0

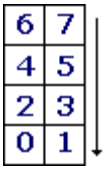
Table de codage pour le mode alphanumérique

Car	Val	Car	Val	Car	Val	Car	Val	Car	Val	Car	Val	Car	Val	Car	Val
0	0	6	6	C	12	I	18	O	24	U	30	SP	36	.	42
1	1	7	7	D	13	J	19	P	25	V	31	$	37	/	43
2	2	8	8	E	14	K	20	Q	26	W	32	%	38	:	44
3	3	9	9	F	15	L	21	R	27	X	33	*	39		
4	4	A	10	G	16	M	22	S	28	Y	34	+	40		
5	5	B	11	H	17	N	23	T	29	Z	35	-	41		

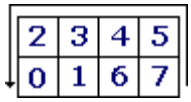
Mise en place de bits dans des caractères de symbole

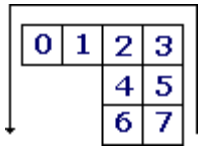
ascendant

descendant

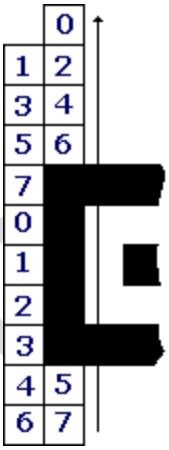


ascendant vers descendant



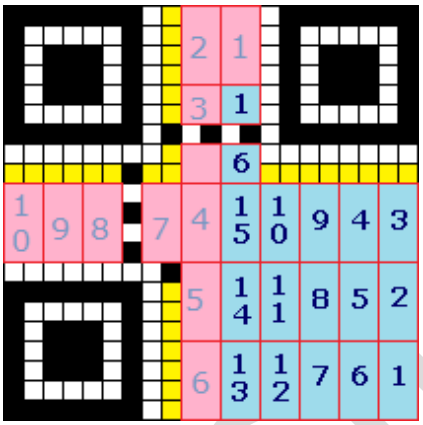


avec motif



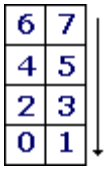
Disposition des caractères

Pour la version 1 avec niveau de correction d'erreurs M

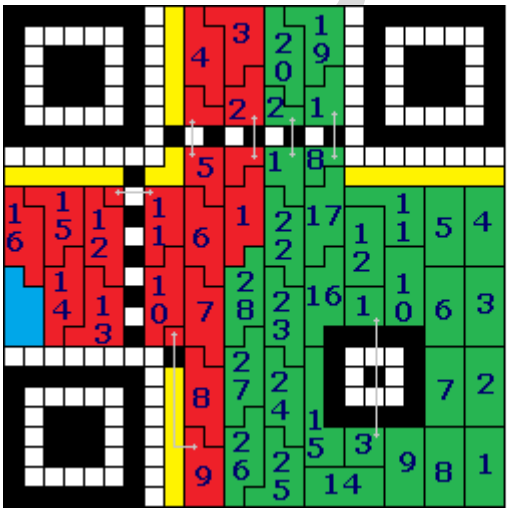


Le schéma ci-contre montre pour un niveau de correction M
l'ordre de placement des octets de données (bleu 1 à 16)
et de correction d'erreurs (rose 1 à 10), ainsi que les zones
réservées aux informations de format (jaune)
Il n'y a pas d'informations de version
Il n'y a pas de bits résiduels

remplissage des octets

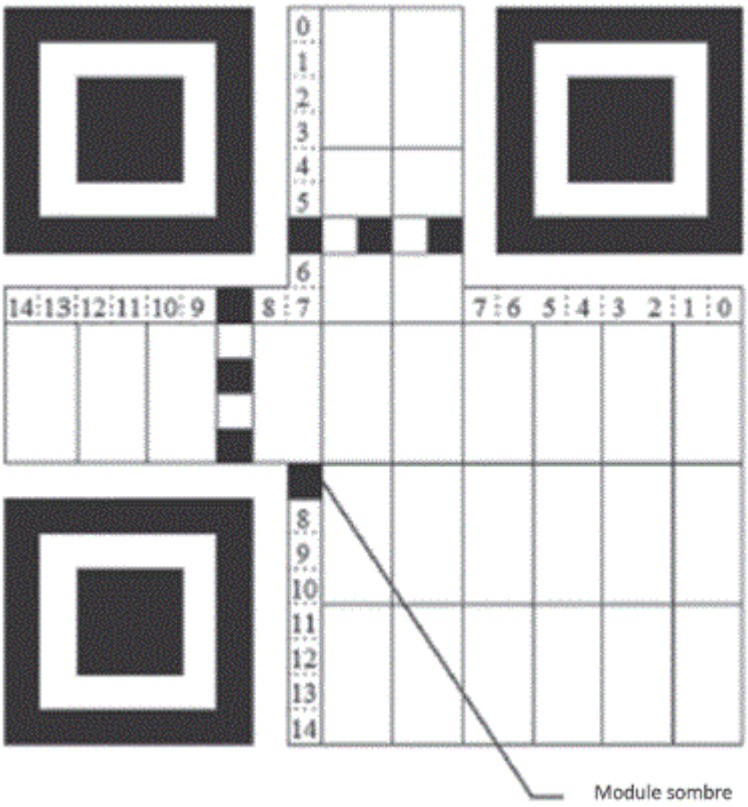


Pour la version 2 avec niveau de correction d'erreurs M



le schéma ci-contre montre
pour un niveau de correction M
l'ordre de placement
des octets de données
de 1 à 28 (vert)
et des octets de correction
d'erreurs de 1 à 16 (rouge)
ainsi que les zones réservées
aux informations de format
et de version (jaune)
et aux bits résiduels (bleu)

Position des informations de format



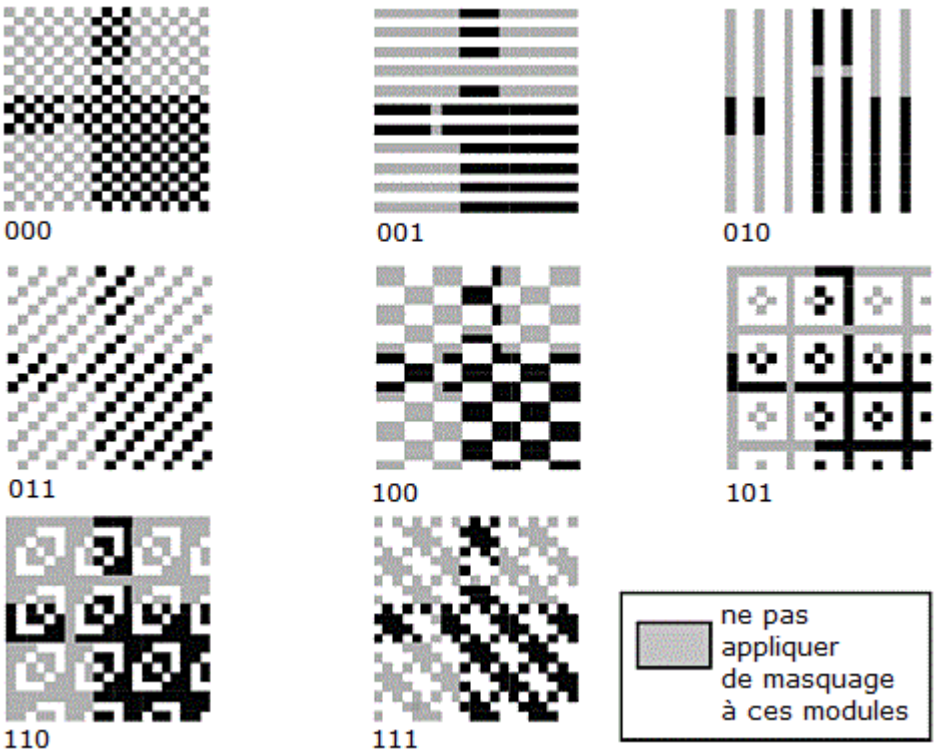
Les informations de format sont constituées d'une séquence de 15 bits comprenant
5 bits de données
et 10 bits de correction d'erreurs calculés en utilisant le code BCH

Les deux premiers bits indiquent le niveau de correction d'erreurs suivant le tableau:

niveau de correction d'erreurs	L	M	Q	H
indicateur binaire	01	00	11	10

Du troisième au cinquième bit les informations de format contiennent la référence du motif de masquage de données

Motifs de masquage de données



Losqu'un motif de masquage est appliqué aux données, sa partie claire ne change pas les données correspondantes mais sa partie sombre complémente les données correspondantes (fonction ou exclusif)

Exemple 1: chaîne 01234567

Codage de la chaîne de données 01234567 en un symbole de Code QR
La chaîne de données sera codée en une version 1-M de symbole, en utilisant le mode numérique

Diviser en groupes de trois chiffres → 012 345 67
012 → 0000001100
345 → 0101011001
67 → 1000011

Concaténer ces données → 0000001100 0101011001 1000011

Convertir l'indicateur de nombre de caractères en binaire sur 10 bits : 8 → 0000001000

Ajouter l'indicateur de mode numérique → 0001
et l'indicateur de nombre de caractères au début de la concaténation précédente
→ 0001 0000001000 0000001100 0101011001 1000011

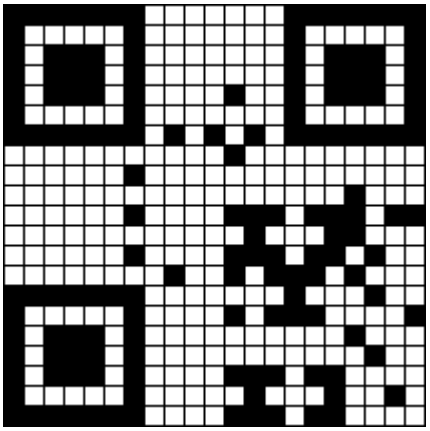
Ajouter le motif de terminaison 0000
→ 0001 0000001000 0000001100 0101011001 1000011 0000

Diviser en octets de données de 8 bits, en ajoutant des bits de remplissage à la fin en fonction des besoins (ici 3 x 0)
→ 00010000 00100000 00001100 01010110 01100001 10000000

Ajouter des octets de données de remplissage 11101100 et 00010001 alternativement, pour atteindre la capacité du symbole pour la version 1-M (16 octets). Il y a 6 octets de données, par conséquent 10 octets de données de remplissage sont nécessaires
11101100 00010001

→ 00010000 00100000 00001100 01010110 01100001 10000000
  11101100 00010001 11101100 00010001 11101100 00010001
  11101100 00010001 11101100 00010001

Ceci donne la grille suivante :



Le niveau de correction d'erreurs est M → 00
Sélectionner le motif de masquage de données → 010

Bits de données des informations de format → 00010

Le calcul de la correction d'erreurs BCH donne 1001101110 comme séquence à ajouter aux données

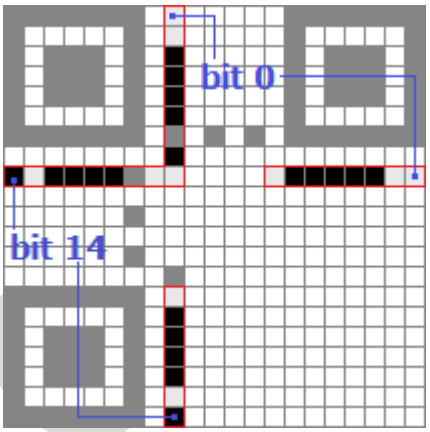
Information de format → 000101001101110

Appliquer l'opération de disjonction à ce train binaire avec le masque 101010000010010

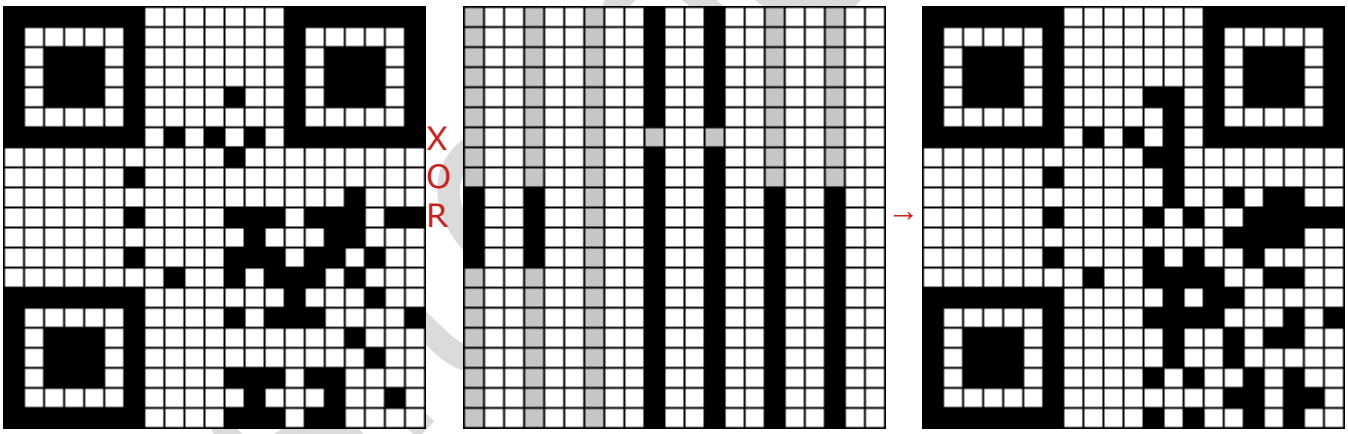
```
      000101001101110
xor   101010000010010
=     101111001111100
```

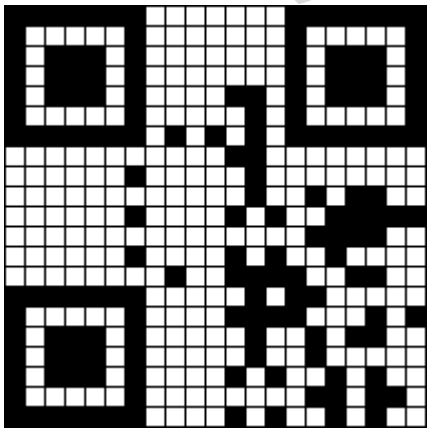
Ce résultat peut être vérifié avec :
Utilitaire de calcul du code correcteur BCH

niveau de correction d'erreurs	L	M	Q	H
indicateur binaire	01	00	11	10

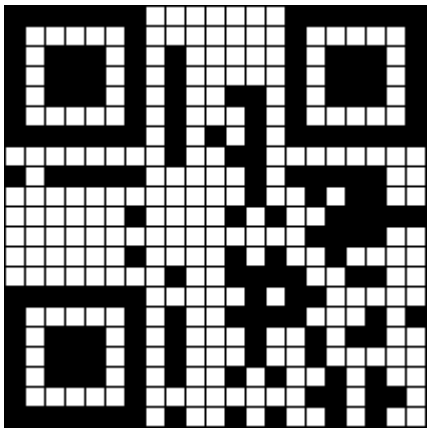


Les informations de format seront à placer sur le symbole, comme indiqué ci-dessous, après l'application du masque 010 sur les octets de données



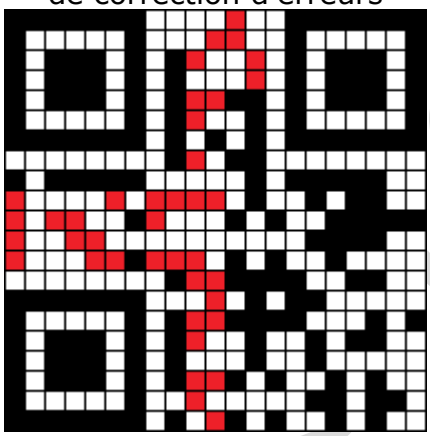


+ informations de format →

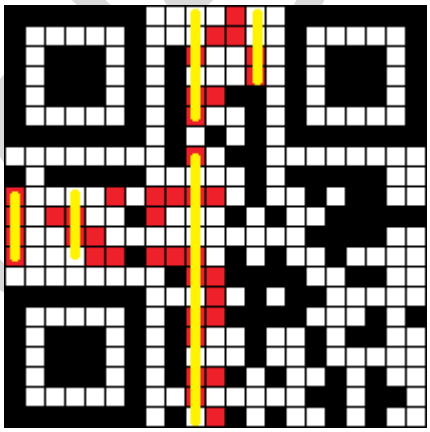


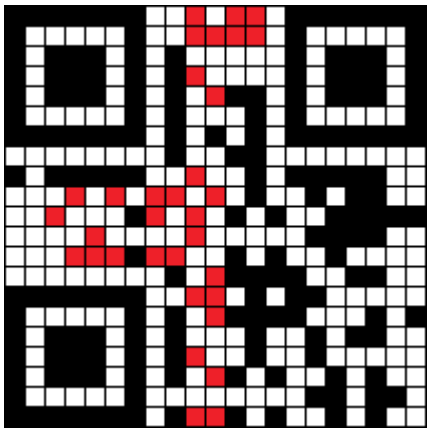
données (binaire)	hexadécimal	décimal	correction d'erreurs (décimal)	hexadécimal	binaire
00010000	10	16	165	A5	10100101
00100000	20	32	36	24	00100100
00001100	0C	12	212	D4	11010100
01010110	56	86	193	C1	11000001
01100001	61	97	237	ED	11101101
10000000	80	128	54	36	00110110
11101100	EC	236	199	C7	11000111
00010001	11	17	135	87	10000111
11101100	EC	236	44	2C	00101100
00010001	11	17	85	55	01010101
11101100	EC	236			
00010001	11	17	*pour le niveau de correction M*		
11101100	EC	236	*il y a 10 octets de correction*		
00010001	11	17	*d'erreurs qui ont été obtenus avec*		
11101100	EC	236	Utilitaire de calcul du code correcteur RS		
00010001	11	17			

avec les octets
de correction d'erreurs

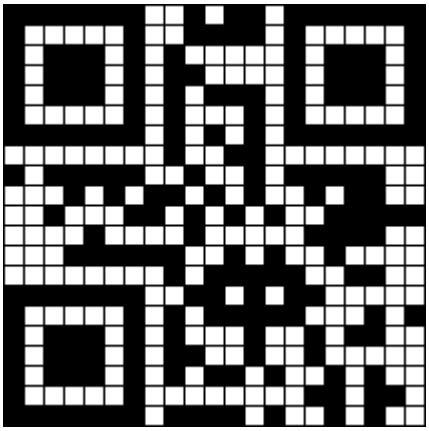


application du masque (en jaune)
sur les octets de correction d'erreurs





Code QR définitif



Lecture possible
de ce code QR
avec l'application :
Android QR Code Reader

Exemple 2: URL

HTTPS://SITELEC.ORG

H	T	T	P	S	:	/	/	S	I	T	E	L	E	C	.	O	R	G
17	29	29	25	28	44	43	43	28	18	29	14	21	14	12	42	24	27	16

Dans le mode alphanumérique, 2 caractères sont regroupés et codés sur 11 bits. Si le nombre de caractères à coder n'est pas un multiple de 2, le dernier caractère est codé sur 6 bits

(17, 29) (29, 25) (28, 44) (43, 43) (28, 18) (29, 14) (21, 14) (12, 42) (24, 27) (16)

Ensuite, il faut prendre séparément chaque paire de valeurs de caractères, multiplier le premier chiffre par 45 puis lui ajouter le second chiffre. Si ce n'est pas un multiple de 2 et que la valeur du caractère est seule, on ne fait aucune opération

	1024	512	256	128	64	32	16	8	4	2	1
17x45+29=794	0	1	1	0	0	0	1	1	0	1	0
29x45+25=1330	1	0	1	0	0	1	1	0	0	1	0
28x45+44=1304	1	0	1	0	0	0	1	1	0	0	0
43x45+43=1978	1	1	1	1	0	1	1	1	0	1	0
28x45+18=1278	1	0	0	1	1	1	1	1	1	1	0
29x45+14=1319	1	0	1	0	0	1	0	0	1	1	1
21x45+14=959	0	1	1	1	0	1	1	1	1	1	1
12x45+42=582	0	1	0	0	1	0	0	0	1	1	0
24x45+27=1107	1	0	0	0	1	0	1	0	0	1	1
16	X	X	X	X	X	0	1	0	0	0	0

Concaténation des données

Cette concaténation se fait dans l'ordre du codage:

|0 1 1 0 0 0 1 1 0 1 0|1 0 1 0 0 1 1 0 0 1 0|…..|0 1 0 0 0 0|

Il faut ensuite ajouter au début de la chaîne, les 4 bits qui définissent le mode:

ECI	numérique	alpha numérique	octet	Kanji	adjonction structurée	FNC1	motif de terminaison
0111	0001	0010	0100	1000	0011	0101 ou 1001	0000

(alphanumérique → 0010)

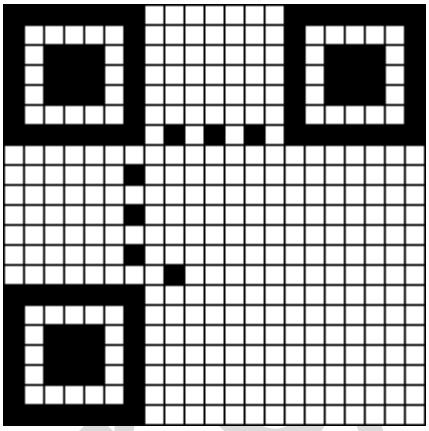
|0 0 1 0|0 1 1 0 0 0 1 1 0 1 0|1 0 1 0 0 1 1 0 0 1 0|…..|0 1 0 0 0 0|

Il faut aussi indiquer le nombre de caractères: HTTPS://SITELEC.ORG → 19 caractères

Choix de la version

- 11 x 9 + 6
  soit 105 bits pour coder les données
- 4 bits pour l'indicateur de mode
- 9, 11 ou 13 bits pour l'indicateur
  du nombre de caractères
  → 9 pour la version 1

soit 118 bits :
on prend un symbole de version 1 (21x21)



avec le type de correction d'erreur L ou M
avec le type L → 19 x 8 = 152 données >118
avec le type M → 16 x 8 = 128 données >118

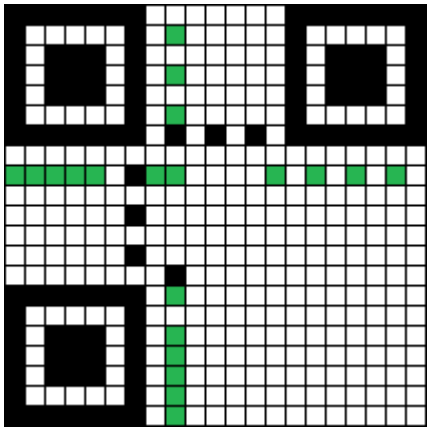
Nombre de bits dans l'indicateur de nombre de caractères

numéro de version	numérique	alphanumérique	octet
1→ 9	10	9	8
10→ 26	12	11	16
27→ 40	14	13	16

L'indicateur du nombre de caractères vaut 19 = 16+2+1 sur 9 bits → 000010011
Ces données sont ajoutées après l'indicateur de mode
|0 0 1 0|0 0 0 0 1 0 0 1 1|0 1 1 0 0 0 1 1 0 1 0|1 0 1 0 0 1 1 0 0 1 0|…..|0 1 0 0 0 0|

Les données binaires doivent être découpées par octets. Ces chaînes de 8 bits sont appelées codewords. On ajoute des 0 à la dernière chaîne si nécessaire afin de compléter l'octet

données avant découpage	octets		hexa	déc	commentaire
0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0	1	0 0 1 0 0 0 0 0	20	32	
	2	1 0 0 1 1 0 1 1	9B	155	
	3	0 0 0 1 1 0 1 0	1A	26	
	4	1 0 1 0 0 1 1 0	A6	166	
	5	0 1 0 1 0 1 0 0	54	84	
	6	0 1 1 0 0 0 1 1	63	99	
	7	1 1 0 1 1 1 0 1	DD	221	
	8	0 1 0 0 1 1 1 1	4F	79	
	9	1 1 1 0 1 0 1 0	EA	234	
	10	0 1 0 0 1 1 1 0	4E	78	
	11	1 1 1 0 1 1 1 1	EF	239	
	12	1 1 0 1 0 0 1 0	D2	210	
	13	0 0 1 1 0 1 0 0	34	52	
	14	0 1 0 1 0 0 1 1	53	83	
	15	0 1 0 0 0 0 0 0	40	64	ajout de 2 x 0
	16	0 0 0 0 0 0 0 0	00	0	ajout octet de 0
	17	1 1 1 0 1 1 0 0	EC	236	ajout de
	18	0 0 0 1 0 0 0 1	11	17	3 octets
	19	1 1 1 0 1 1 0 0	EC	236	de remplissage



Le niveau de correction d'erreurs est L → 01
Le motif de masquage de données est 2 → 010
Bits de données des informations de format → 01010
Le calcul de la correction d'erreurs BCH donne 0110111000 comme séquence à ajouter aux données
Information de format → 010100110111000
Appliquer l'opération de disjonction à ce train binaire avec le masque 101010000010010

```
        010100110111000
xor     101010000010010
=       111110110101010
```

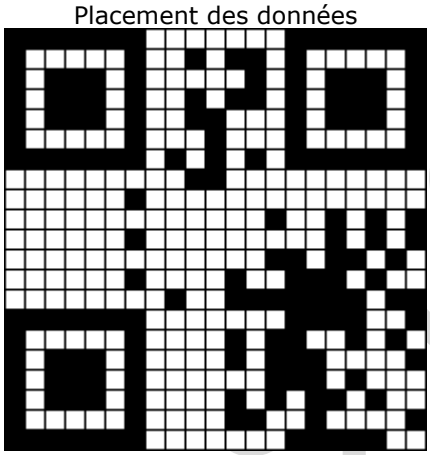
Ce résultat peut être vérifié avec : Utilitaire de calcul du code correcteur BCH

Le calcul de la correction d'erreurs pour un niveau L a donné 7 octets de correction d'erreurs:
102 246 148 162 190 56 45

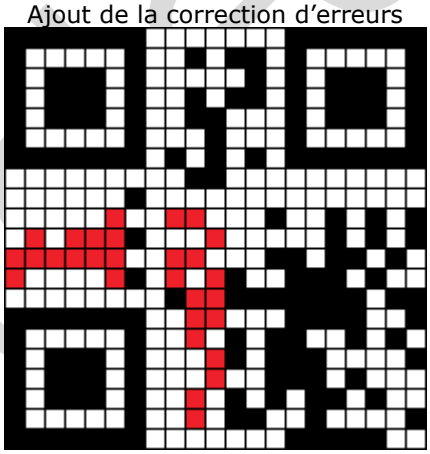
Ce résultat peut être vérifié avec :
Utilitaire de calcul du code correcteur RS

décimal	hexadécimal	binaire
102	66	01100110
246	F6	11110110
148	94	10010100
162	A2	10100010
190	BE	10111110
56	38	00111000
45	2D	00101101

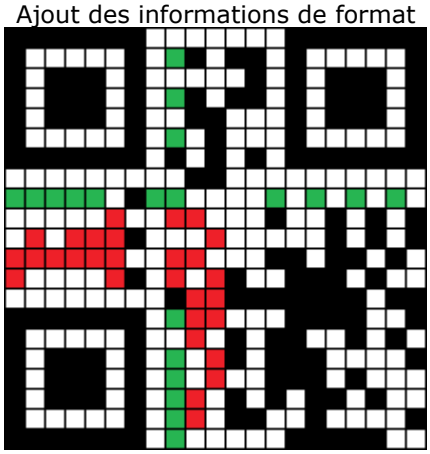
Placement des données



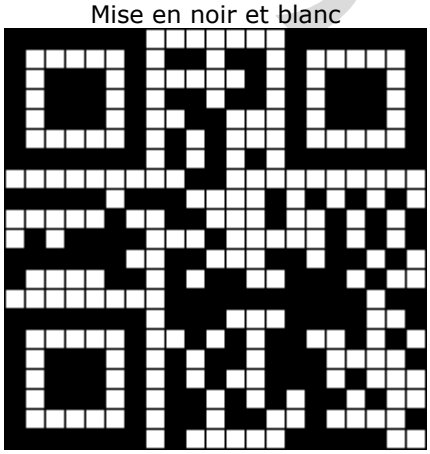
Ajout de la correction d'erreurs



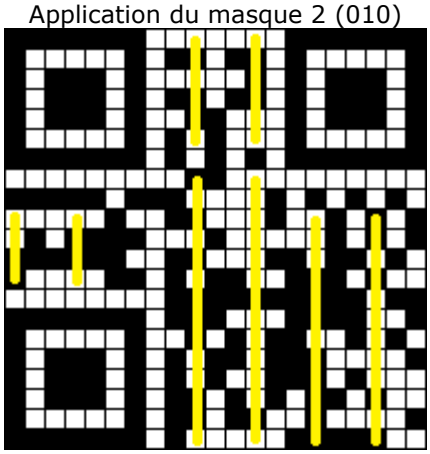
Ajout des informations de format



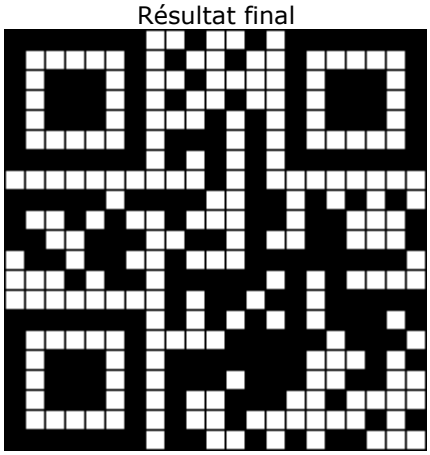
Mise en noir et blanc



Application du masque 2 (010)

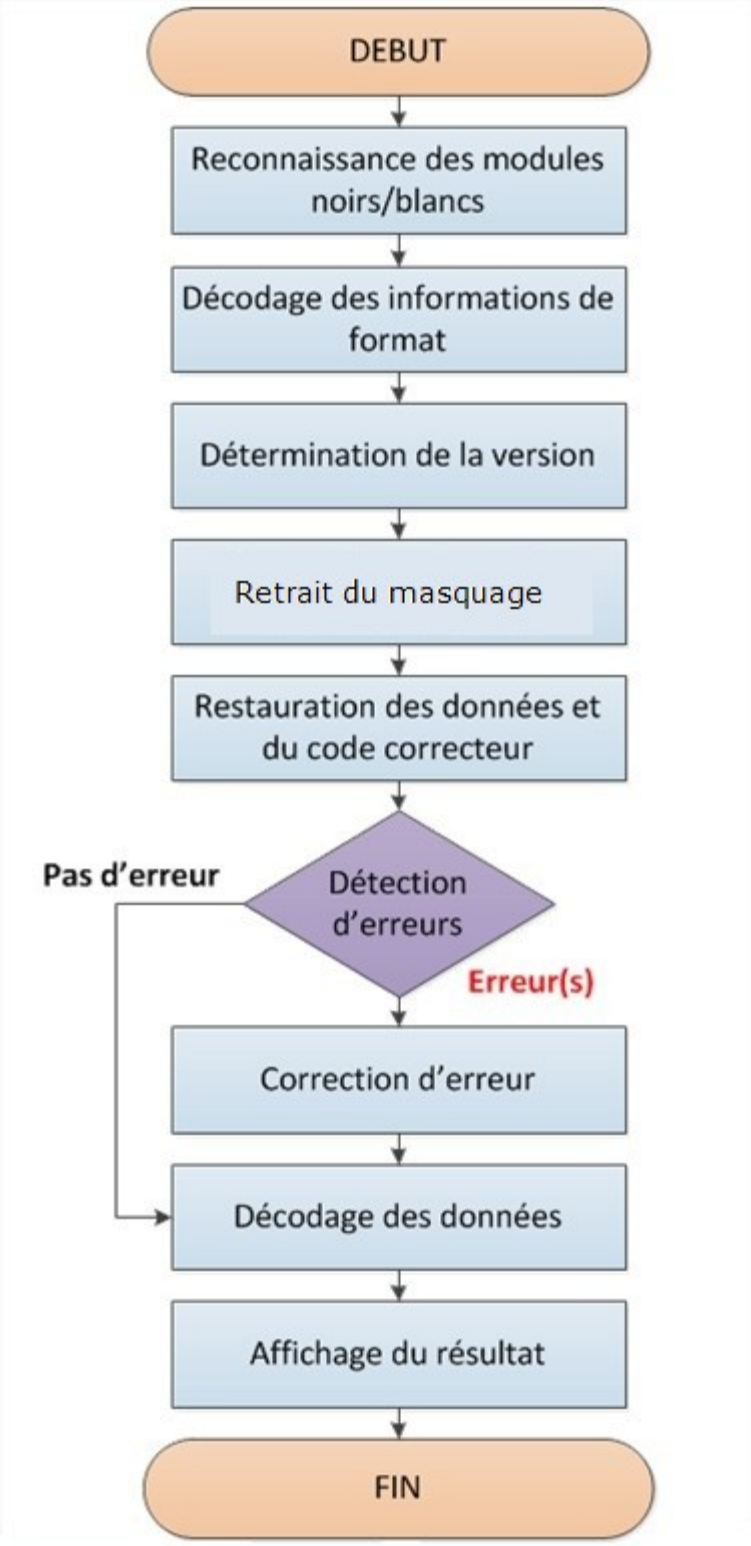


Résultat final



Lecture possible du QR Code précédent avec l'application : Android QR Code Reader

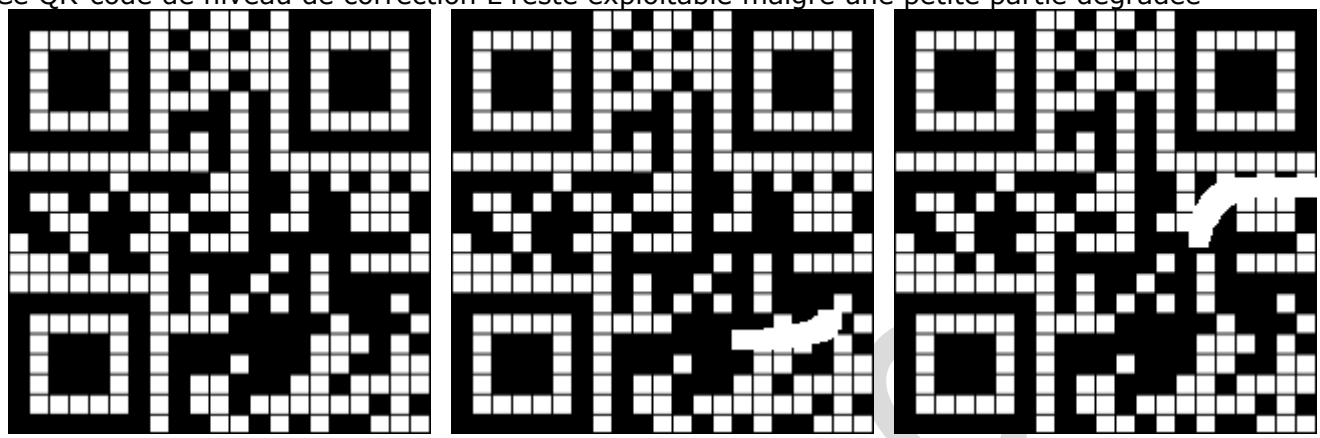
Étapes de décodage du Code QR



Altération du motif

Niveau de correction L

Ce QR-code de niveau de correction L reste exploitable malgré une petite partie dégradée



Lecture possible des QR Codes précédents avec l'application : Android QR Code Reader

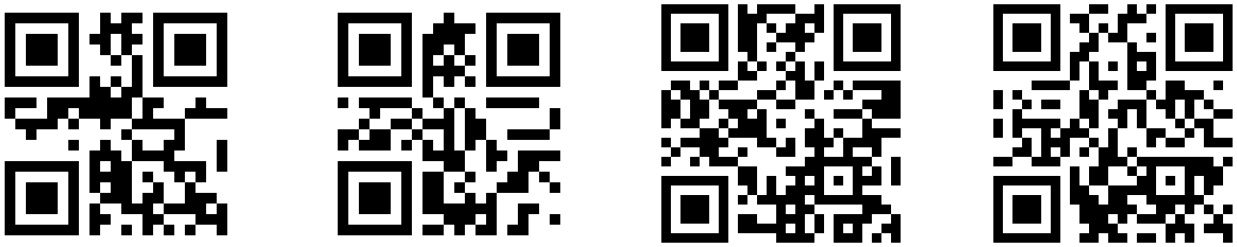
Niveau de correction H

Ce QR-code de niveau de correction H reste exploitable malgré une grande partie dégradée

Lecture possible des QR Codes précédents avec l'application : Android QR Code Reader

Codes QR, avec différents niveaux de correction d'erreurs, renvoyant la même URL

L M Q H



Lecture possible des QR Codes précédents avec l'application : Android QR Code Reader

Voir aussi la page https://sitelec.org/cours/abati/qrcode/qrcode.html

11 août 2019