Barcodes – principle

Identification systems (IDFS)

Department of Control and Telematics
Faculty of Transportation Sciences, CTU in Prague

Contents

How does it work?

- **Bulls eye code**
- PostNet
- 1D Bar code
- 2D Bar code

Bulls eye code

PostNet

1D Bar code

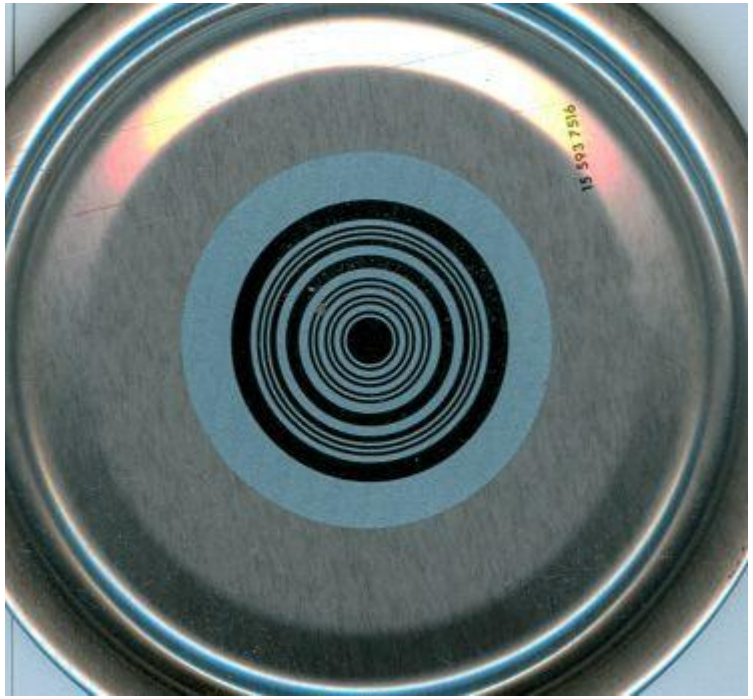
2D Bar code

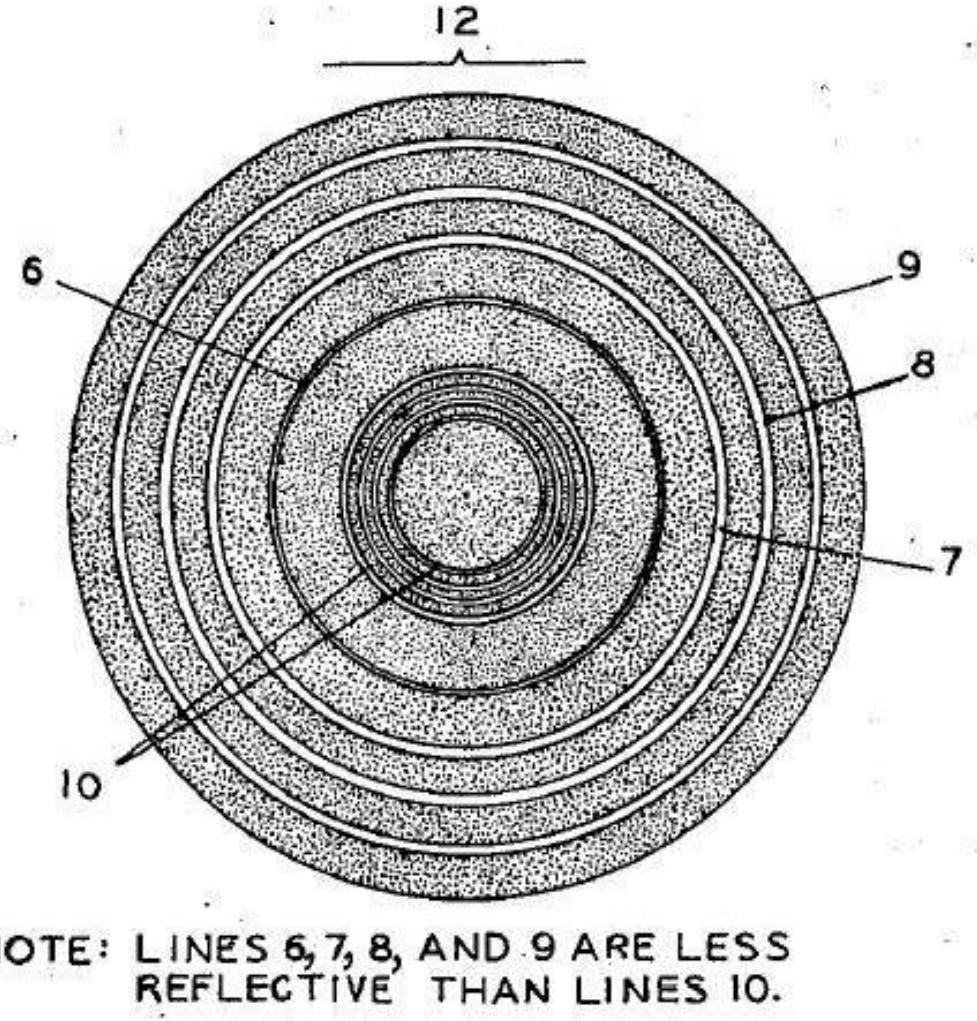
HOW DOES IT WORK?

2. How does it work

Bulls eye code

- Bulls eye code





2. How does it work

Bulls eye code

FIG. 1

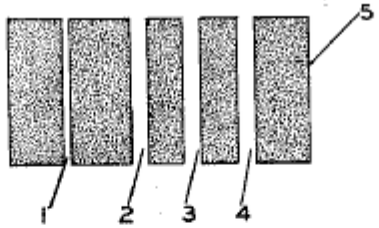


FIG. 2

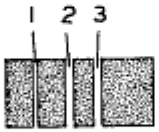


FIG. 3

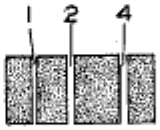


FIG. 4

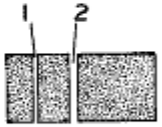


FIG. 5

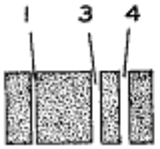


FIG. 6

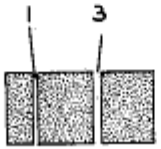


FIG. 7

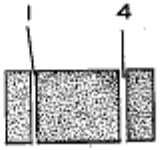


FIG. 8

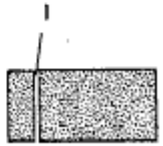


FIG. 9

2. How does it work

Bulls eye code

- Reader output

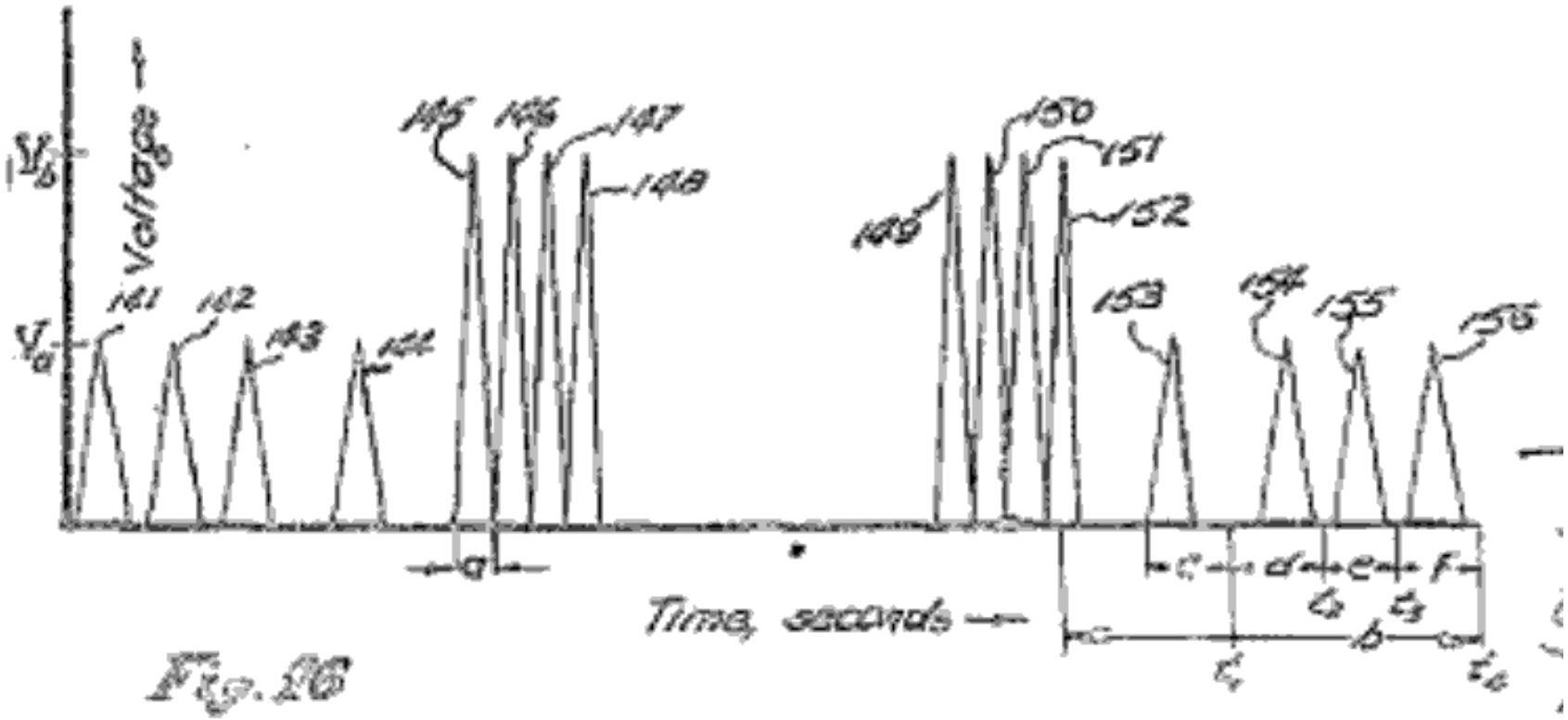


Fig. 26

Bulls eye code

PostNet

1D Bar code

2D Bar code

HOW DOES IT WORK?

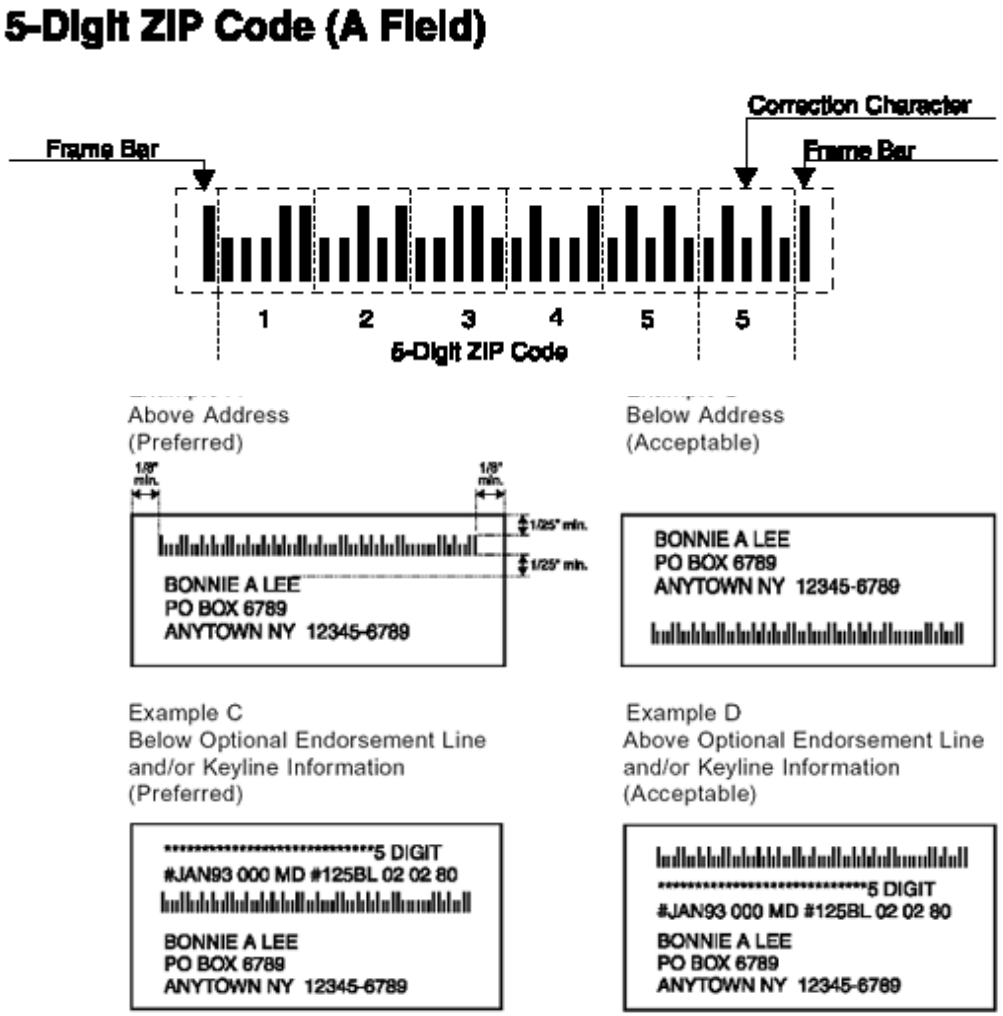
2. How does it work

PostNet

- PostNet code

Code Elements

Numeric Value	Binary Code Value 7 4 2 1 0	Barcode Value 74210
1	00011	
2	00101	
3	00110	
4	01001	
5	01010	
6	01100	
7	10001	
8	10010	
9	10100	
0	11000	



Bulls eye code

PostNet

1D Bar code (UPC/EAN/GS1 DataBar/…)

2D Bar code

HOW DOES IT WORK?

2. How does it work?

CODE 2 of 5

Composition

- Every character of this code, excluding start and stop character, is formed by 5 bars (2 wide + 3 narrow),
- every character is represented by same width in the barcode.
- Parallel spaces between bars have same width (do not cary information)

Character	Bar 1	Bar 2	Bar 3	Bar 4	Bar 5
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
Start	1	1	0		
Stop	1	0	1		

2. How does it work?

UPC code

Composition

- The scan able area of every UPC-A barcode follows the pattern **SLLLLLLMRRRRRRE**, where the S (start), M (middle), and E (end). The L (left) and R (right) sections collectively represent the 12 numerical digits that make each UPC unique.
- The first digit **L** is the prefix. The last digit **R** is an error correcting check digit,
- the guard bars, separate the groups of six digits
- L/R = 7 modules, S/E = 3 modules, M = 5 modules total 95 modules of the same width

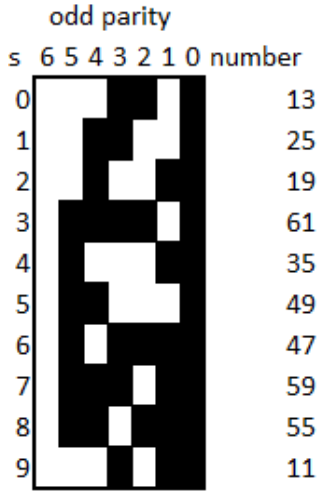
http://en.wikipedia.org/wiki/Universal_Product_Code

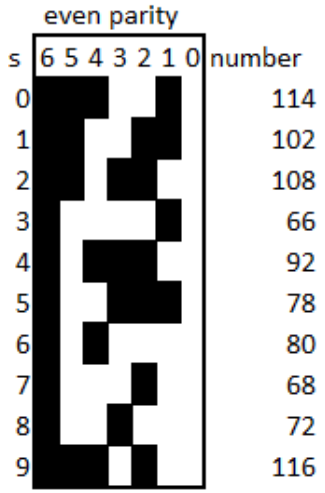
2. How does it work

UPC Code

How to read bars?

- Each digit: **four vertical lines**, two black and two white. (7 modules), L and R have **reversed** values (color)





Bulls eye code

1D Bar code

2D matrix code (PDF417/DataMatrix/QRCode/…)

HOW DOES IT WORK?

2. How does it work

PDF-417

Composition:

- size of the symbol can be modified
- multiple linear bar-codes stacked above
- Symbol = ratio of the widths of the bars and spaces to each other
- maximum of 90 rows and 30 columns
- **capable of storing up to 2710 digits (1850 aflanumeric chars, 1108 bytes)**

Compaction mode	*Datas to encode*	*Rate compaction*
"Byte"	ASCII 0 to 255	1.2 byte per CW
"Text"	ASCII 9, 10, 13 & 32 a 127	2 characters per CW
"Numeric"	Only digits 0 to 9	2.9 digits per CW

2D Barcode PDF-417 with 6 characters

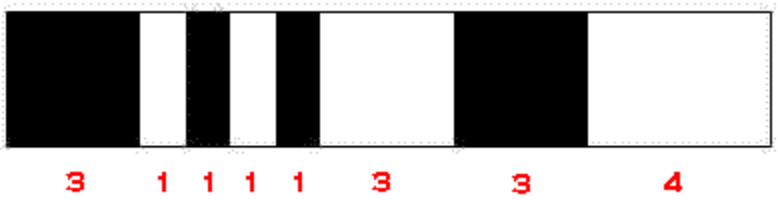
2D Barcode PDF-417 with 200 characters

2. How does it work

PDF-417

Code word:

- 4 bars and 4 spaces which totals 17 modules in width.
- Each bar and space can be from 1 to 6 modules in length.
- In theory it has 9*929 patterns. Each set of 929 patterns is called a cluster (character set). PDF417 only uses cluster number 0, 3 and 6.
- Adjacent rows use different clusters in the sequence 0, 3, 6, 0, 3, 6



http://grandzebu.net/informatique/codbar-en/pdf417.htm

2. How does it work

PDF-417

- The CW number 900 to 928 have special meaning, some enable to switch between modes in order to optimise the code.

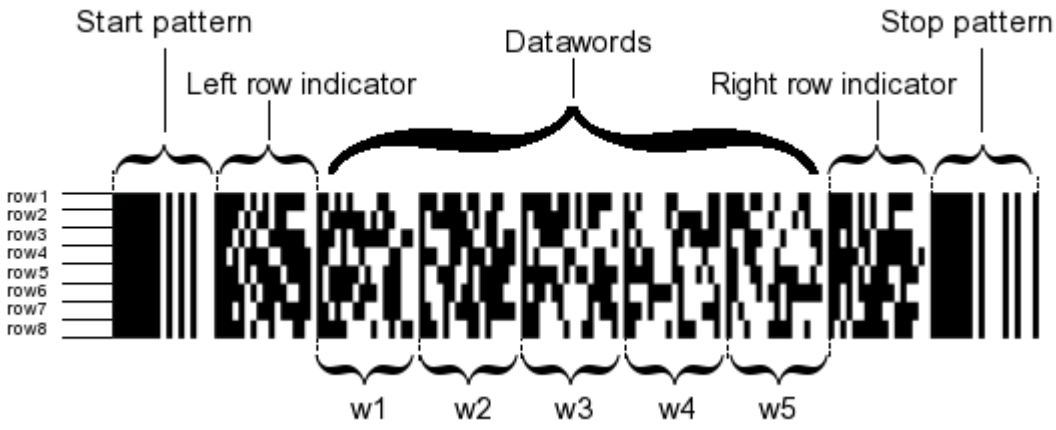
CW number :	*Function*
900	Switch to "Text" mode
901	Switch to "Byte" mode
902	Switch to "Numeric" mode
903 a 912	Reserved
913	Switch to "Octet" only for the next CW
914 a 920	Reserved
921	Initialization
922	Terminator codeword for Macro PDF control block
923	Sequence tag to identify the beginning of optional fields in the Macro PDF control block
924	Switch to "Byte" mode (If the total number of byte is multiple of 6)
925	Identifier for a user defined Extended Channel Interpretation (ECI)
926	Identifier for a general purpose ECI format
927	Identifier for an ECI of a character set or code page
928	Macro marker CW to indicate the beginning of a Macro PDF Control Block

2. How does it work

PDF-417

- Start and stop pattern (static and are the same for all barcodes)
- Left and right row indicators (chosen to achieve maximum contrast, also bear row number and error correction level)
- Data and data count (unique for each barcode and repre-sents the encoded data, numeric, alpha, …)
- Error correction code-words (2 min, 510 max)

Start pattern	Left row indicator	Data count	Datawords	Right row indicator	Stop pattern
Start pattern	Left row indicator	Datawords		Right row indicator	Stop pattern
...	...	...		...	...
Start pattern	Left row indicator	Datawords		Right row indicator	Stop pattern
Start pattern	Left row indicator	Datawords	Error correction	Right row indicator	Stop pattern



2. How does it work

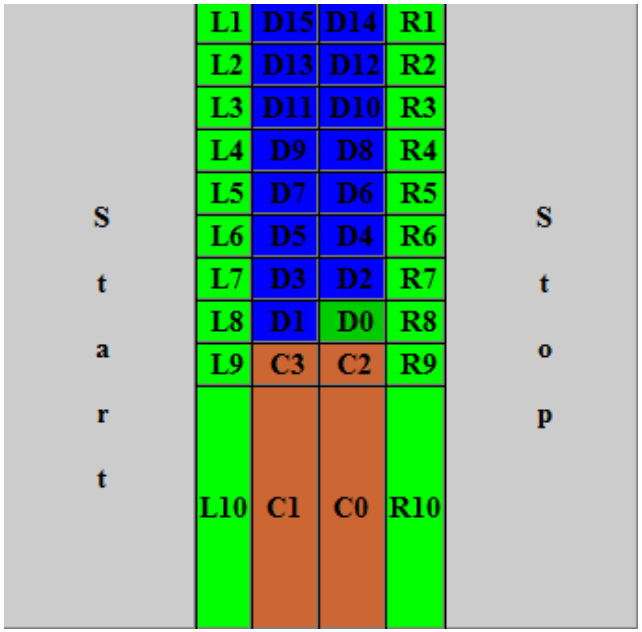
PDF-417

Example:

- First CW indicates CW total number of the code including: data, CW of stuffing and itself but excluding CW correction.
- Sample of code with 14 data CW, a 15th CW indicate CW number, one padding CW and 4 correction CW. (Level 1)

Structure

- D15 = length descriptor (16 in this sample)
- D0 = padding
- D1 a D14 = data
- L1 a L10 = left side CW
- R1 a R10 = right side CW
- C0 a C3 = error correction, level 1



2. How does it work

PDF-417

Example: **4 different character sets:**

- Each CW encode 2 characters ;
- if C1 and C2 are the values of the two characters, CW value is : C1 x 30 + C2
- If it remains an alone character, we add to it a padding switch, for instance T_PUN.

Sample, sequence to encode : Super !

S : 18, LOW : 27, u : 20, p : 15, e : 4, r : 17, SPACE : 26, T_PUN : 29, ! : 10
that is 9 characters, we'll add a T_PUN for the padding.

$CW_1 = 18 \times 30 + 27 = 567$

$CW_2 = 20 \times 30 + 15 = 615$

$CW_3 = 4 \times 30 + 17 = 137$

$CW_4 = 26 * 30 + 29 = 809$

$CW_5 = 10 \times 30 + 29 = 329$

The sequence is consequently : 567, 615, 137, 809, 329

Value	*Uppercase*	*Lowercase*	*Mixed*	*Punctuation*
0	A	a	0	;
1	B	b	1	<
2	C	c	2	>
3	D	d	3	@
4	E	e	4	[
5	F	f	5	\
6	G	g	6	]
7	H	h	7	_
8	I	I	8	` (Quote)
9	J	j	9	~
10	K	k	&	!
11	L	l	CR	CR
12	M	m	HT	HT
13	N	n	,	,
14	O	o	:	:
15	P	p	#	LF
16	Q	q	-	-
17	R	r	.	.
18	S	s	$	$
19	T	t	/	/
20	U	u	+	g
21	V	v	%	|
22	W	w	*	*
23	X	x	=	(
24	Y	y	^	)
25	Z	z	PUN	?
26	SP	SP	SP	{
27	LOW	T_UPP	LOW	}
28	MIX	MIX	UPP	' (Apostrophe)
29	T_PUN	T_PUN	T_PUN	UPP

2. How does it work

PDF-417

- The "Byte" mode allow to encode 256 different bytes, that is the entire extended ASCII table.

Sample 1 : word to encode : alcool

The sequence of bytes (in ASCII) is : 97, 108, 99, 111, 111, 108

$S = 97 \times 256^5 + 108 \times 256^4 + 99 \times 256^3 + 111 \times 256^2 + 111 \times 256 + 108 = 107\ 118\ 152\ 609\ 644$

$CW_0 = 107\ 118\ 152\ 609\ 644 \text{ MOD } 900 = 244$

$S = 107\ 118\ 152\ 609\ 644 \setminus 900 = 119\ 020\ 169\ 566$

$CW_1 = 119\ 020\ 169\ 566 \text{ MOD } 900 = 766$

$S = 119\ 020\ 169\ 566 \setminus 900 = 132\ 244\ 632$

$CW_2 = 132\ 244\ 632 \text{ MOD } 900 = 432$

$S = 132\ 244\ 632 \setminus 900 = 146\ 938$

$CW3 = 146\ 938 \text{ MOD } 900 = 238$

$S = 146\ 938 \setminus 900 = 163$

$CW4 = 163 \text{ MOD } 900 = 163$

The sequence including the switch is consequently : 924, 163, 238, 432, 766, 244

Sample 2 : word to encode : alcoolique

The sequence of bytes (in ASCII) is : 97, 108, 99, 111, 111, 108, 105, 113, 117, 101

The first 6 bytes are coded like above and we add 105, 113, 117 and 101

The sequence including the switch is consequently : 901, 163, 238, 432, 766, 244, 105, 113, 117, 101

2. How does it work

PDF-417

- Left and right side CWs are computed according to the table used for the actual row.
- To obtain the CW value, make the following calculation : (Row Number \ 3) x 30 + X with X taken in the following table.
- (First row is row number 0)

Table used to encode the CWs of this row	*X for the left side CW*	*X for the right side CW*
1	(Number of rows -1) \ 3	Number of data columns - 1
2	(Security level x 3) + (Number of rows -1) MOD 3	(Number of rows -1) \ 3
3	Number of data columns - 1	(Security level x 3) + (Number of rows -1) MOD 3

Bulls eye code

1D Bar code

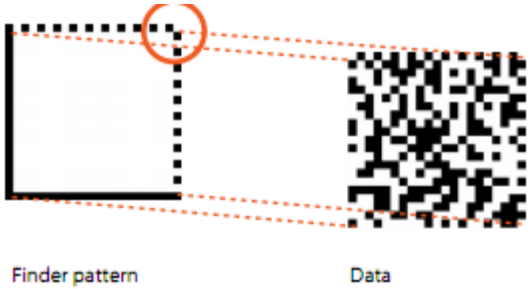
2D matrix code (PDF417/**DataMatrix**/QRCode/…)

HOW DOES IT WORK?

2. How does it work

GS1 - DataMatrix

- Composed of two separate parts the finder pattern (to locate the symbol), and the encoded data itself



Finder Pattern

- defines the shape, the size, X-dimension and the number of rows and columns in the symbol.
- has a function similar to the Auxiliary Pattern in an EAN-13
- The solid dark: "**L finder pattern**" is used to determine the size, orientation and distortion of the symbol.
- Dashed lines: "**Clock Track**" defines the basic structure of the symbol and can also help determine its size and distortion.

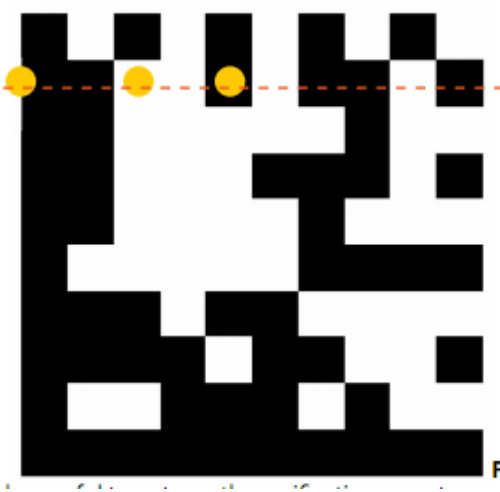
2. How does it work

GS1 - DataMatrix

Symbol structure

- Number of rows and columns – variable from 10 to 144 lines

Symbol Size*		Data Region		Mapping Matrix Size	Total Codewords		Maximum Data Capacity		% of codewords used for Error Correction	Max. Correctable Codewords
							Num.	Alphanum.		Error/Erasure
Row	Col	Size	No.		Data	Error	Cap.	Cap.		
10	10	8x8	1	8x8	3	5	6	3	62.5	2/0
12	12	10x10	1	10x10	5	7	10	6	58.3	3/0
14	14	12x12	1	12x12	8	10	16	10	55.6	5/7
16	16	14x14	1	14x14	12	12	24	16	50	6/9
18	18	16x16	1	16x16	18	14	36	25	43.8	7/11
20	20	18x18	1	18x18	22	18	44	31	45	9/15
22	22	20x20	1	20x20	30	20	60	43	40	10/17



- Example:
  - Symbol size 10x10 + quiet zone 2 = 12 lines/collumns
  - Data part: 8x8 = 8 code words (3 data / 5 error correction)

2. How does it work

GS1 - DataMatrix

Symbol structure

- Divided into data regions, matrix 32x32 into 4 14x14 regions
- Data unit 8 bits = code word

Error correction

- Variable, Reed-Solomon error correction
- Calculates complementary codes and add-ins
- Reconstitutes the original encoded data by recalculating the data from the complementary codes and add-ins.
- The recalculation regenerates the original data by locating errors at the time of scanning.

2. How does it work

GS1 - DataMatrix

Encoding example: char: “123456”

- Data encoding:
  - The ASCII encoding converts the 6 characters into 3 bytes.
  - 12, 34 and 56 (x+130) = 142 164 186 = 3 data code words

- Error correction: (RS algorithm) 5 error correction code words:

Codeword:	1	2	3	4	5	6	7	8
Decimal:	142	164	186	114	25	5	88	102
Hex:	8E	A4	BA	72	19	05	58	66

10001110 10100100 10111010 01110010 00011001 00000101 01011000 01100110

2. How does it work

GS1 - DataMatrix

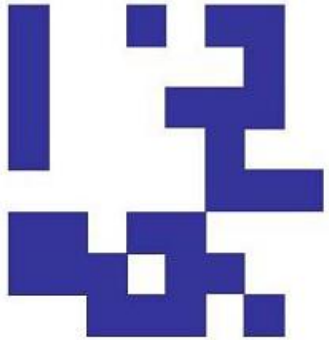
10001110 10100100 10111010 01110010 00011001 00000101 01011000 01100110

The final matrix would be:

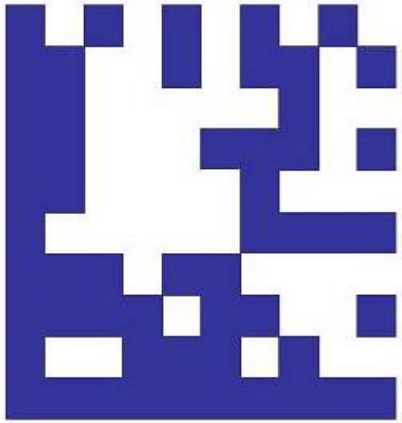
2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6

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

After colouring the patterns which are numbered 1:

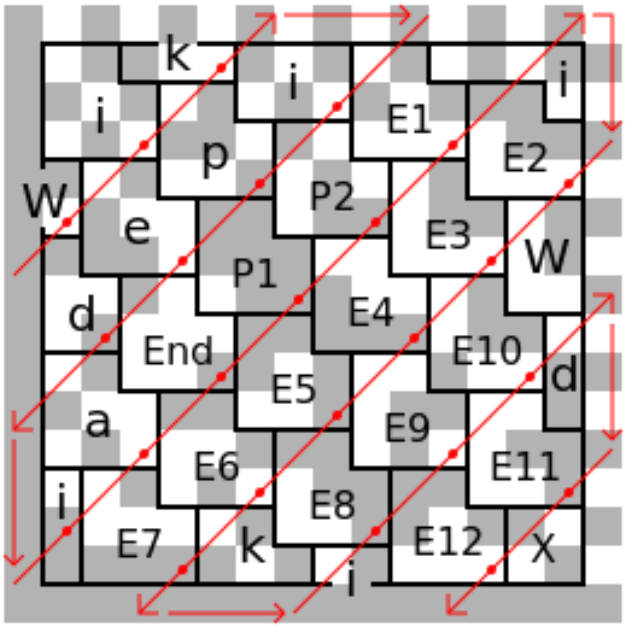


Finally we add the finder pattern to cover the symbol above :



2. How does it work

GS1 - DataMatrix

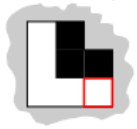


Convert data from ASCII to L-shaped tiles by adding 1 and convert to binary:

Eg Uppercase 'W' = ASCII 87
87 + 1 = 88
= 58 (base 16) = 01011000 (base 2)

0	1	
0	1	1
0	0	0

=



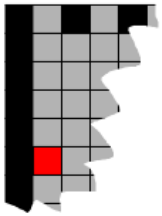
Lowercase 'i' = ASCII 105
105 + 1 = 106
= 6A (base 16) = 01101010 (base 2)

0	1	
1	0	1
0	1	0

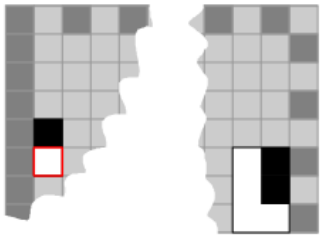
=

Start filling grid from 5th row, 1st column

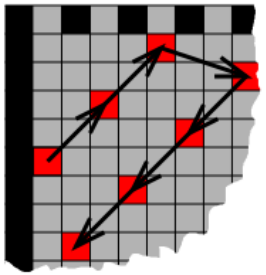
If the tile falls off the edge,
put remainder on the opposite side



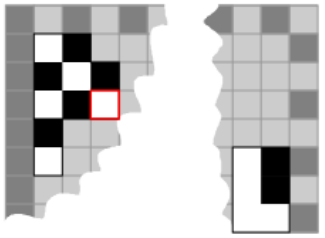
This shows the first W in position



Confinue placing tiles in zig-zag



This shows the W and the i



Bulls eye code

1D Bar code

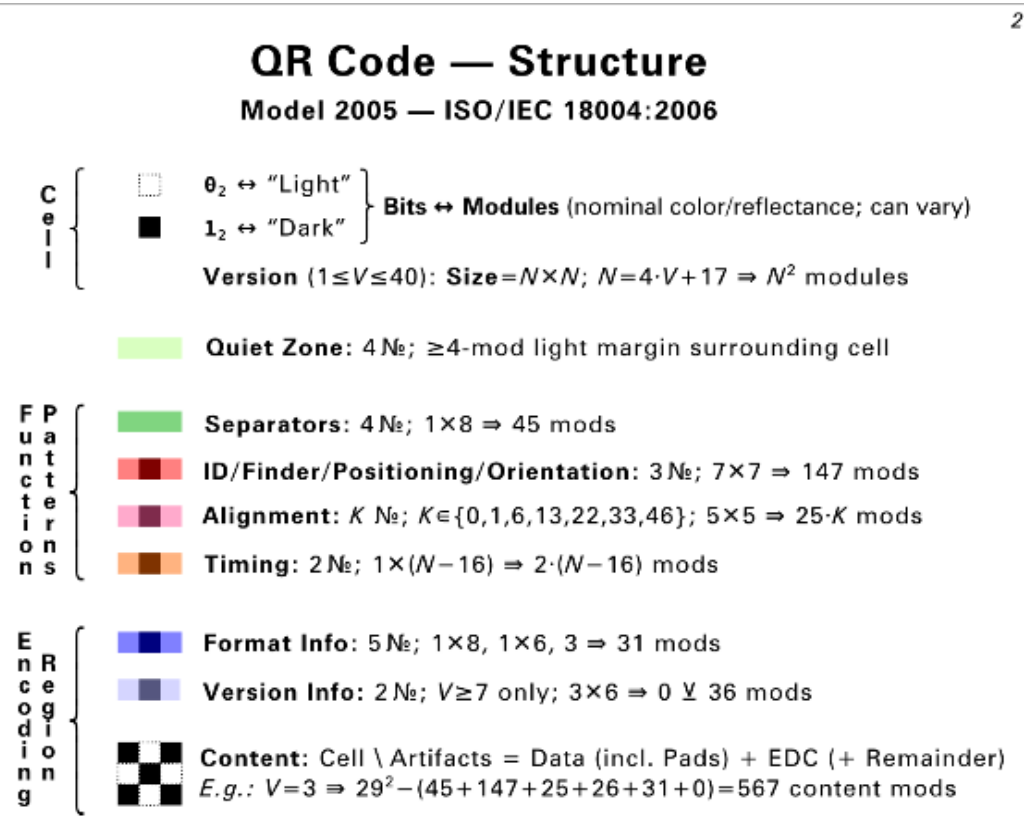
2D matrix code (PDF417/DataMatrix/**QRCode/…)**

HOW DOES IT WORK?

2. How does it work

QR code

© 2012 Walter Tuvell

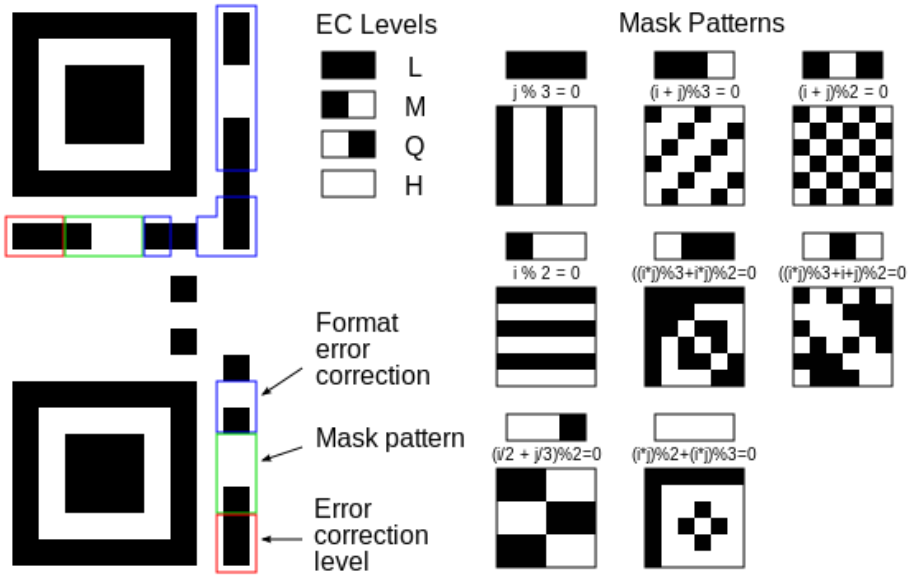


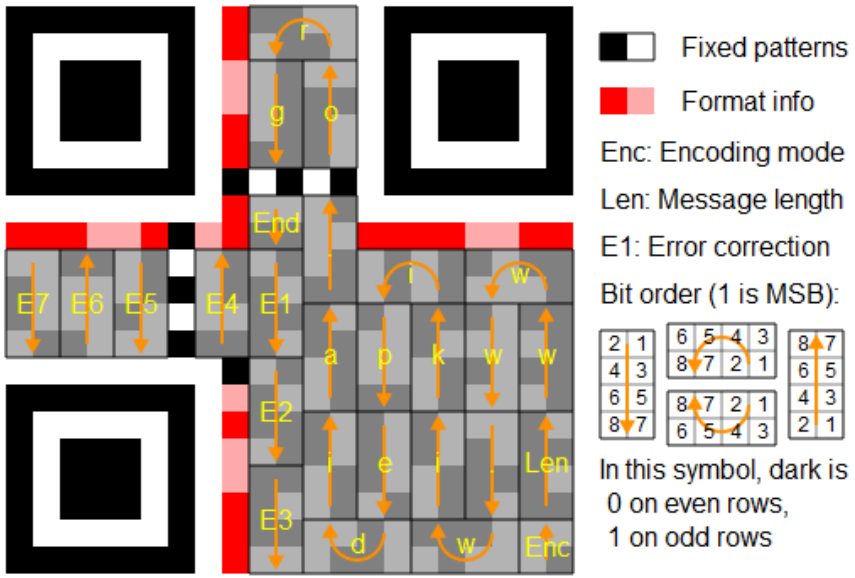
2. How does it work

QR code

Symbol structure

- Number of rows and columns – variable from 21 to 177 lines



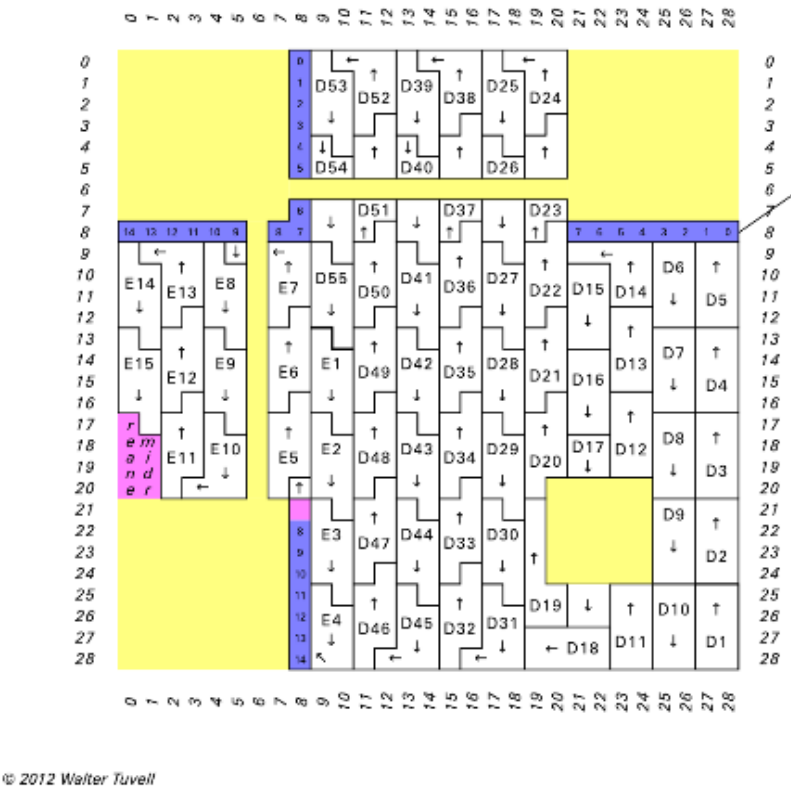


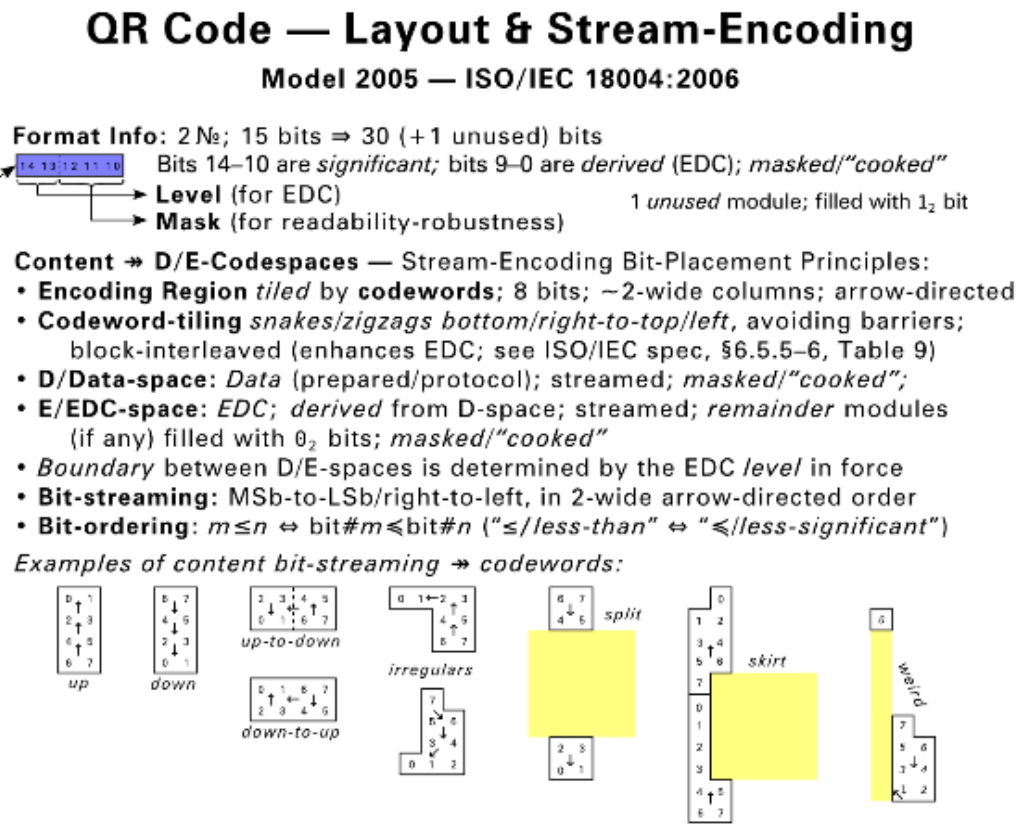
- Data unit 8 bits = code word
- Format info 2x (encoded BCH)

2. How does it work

QR code

- Structure

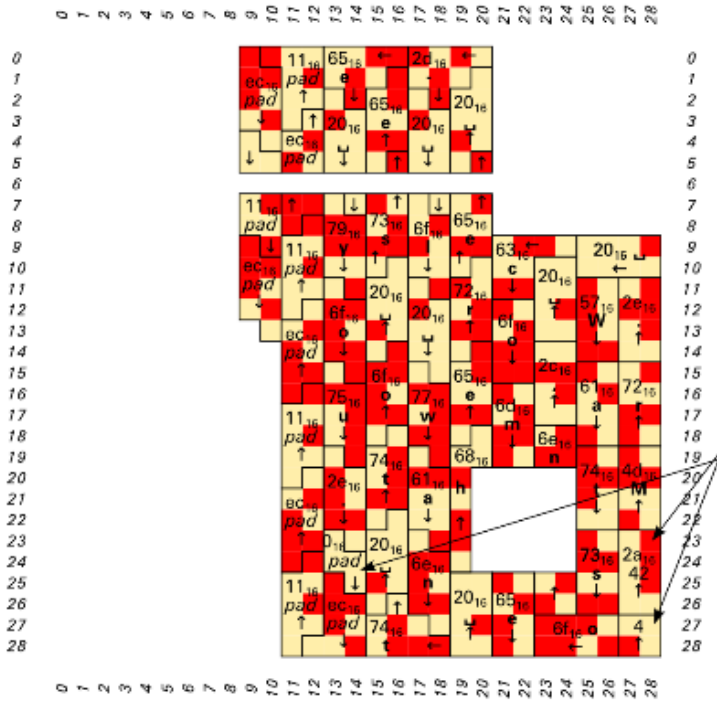




2. How does it work

QR code

Encoding Data



5

QR Code — Protocol(s)

Model 2005 — ISO/IEC 18004:2006

D-Space Content ***(raw/unmasked)***: Sequence of **SDD (Self-Describing Data)** *segments*
Mode Indicator ≙ First 4 bits of SDD segment

Native Modes: SDD is TLV (Type/Length/Value)
$0001_2=1_{16}$ ≙ **N**[umeric] — 0–9 *[3 chars/digits ↠ 10 bits]*
$0010_2=2_{16}$ ≙ **A**[lphanumeric] — 0–9 A–Z ␣ $ % * + - . / : *[2 chars ↠ 11 bits]*
$0100_2=4_{16}$ ≙ **B**[yte|inary] — 00_{16}–ff_{16} *["default" ~ ISO/IEC 8859-1="Latin-1"; 1 char ↠ 8 bits]*
$1000_2=8_{16}$ ≙ **K**[anji] — Shift JIS X 0208 *[see ISO/IEC spec for encoding]*
Type: Character-set (as just indicated, above)
Length: Count of N/A/B/K chars, base-2 encoded in 8–16 bits:

Versions 1– 9	—	N:10	A:9	B:8	K:8
Versions 10–26	—	N:12	A:11	B:16	K:10
Versions 27–40	—	N:14	A:13	B:16	K:12

Value: Standardized, efficient, per-charset encoded bit-stream (as just indicated, above)
Pad-out partial/final (8-bit) D-codeword with 0_2 bits (if necessary)
Pad-out D-space with alternating $11101100_2=ec_{16}$ & $00010001_2=11_{16}$ bytes (if necessary)

FNC1 (Function Code 1) Modes: Pre-defined semantics
$0101_2=5_{16}$ ≙ FNC1, 1st position — See ISO/IEC spec
$1001_2=9_{16}$ ≙ FNC1, 2nd position — See ISO/IEC spec

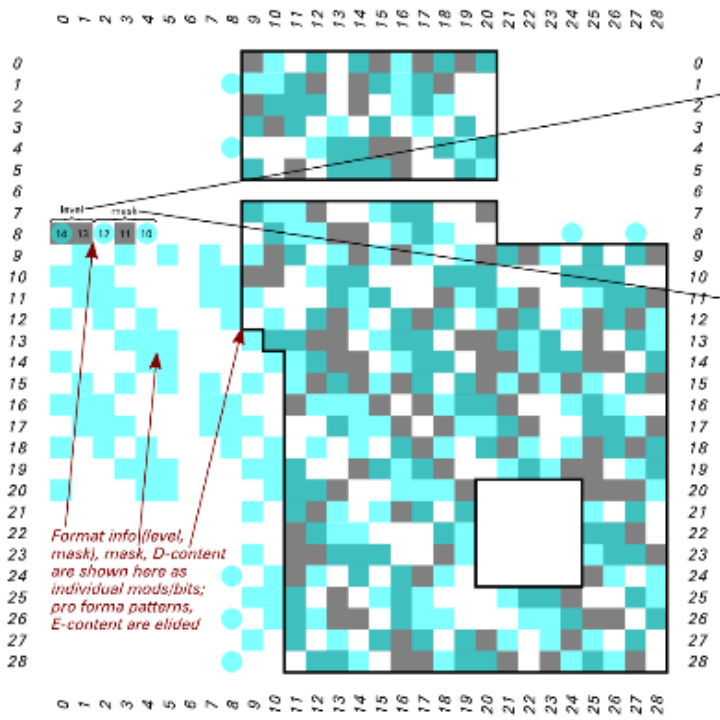
ECI (Extended Channel Interpretation): General escape hatch (e.g., compression, encryption)
$0111_2=7_{16}$ — See ISO/IEC spec

***Faux* Modes:** Structural constructs; not "true" modes
$0011_2=3_{16}$ ≙ **Structured-Append** — Link ≤16 QR code symbols (see ISO/IEC spec)
$0000_2=0_{16}$ ≙ **Terminator/EOM** — Potentially truncated/omitted

2. How does it work

QR code

Masking data:



4

QR Code — Levels & Masks

Model 2005 — ISO/IEC 18004:2006

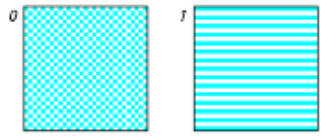
Level (EDC): Determines count/distribution of codewords in D-space & E-space
~7½% EDC levels: $01_2=1\triangleq$**L[ow]**; $00_2=0\triangleq$**M[edium]**; $11_2=3\triangleq$**Q[uality]**; $10_2=2\triangleq$**H[igh]**
Detailed version/level definitions at ISO/IEC spec, §6.4.10, Table 7
E.g.: Version=3 ⇒ 567 content mods/bits ⇒ 70 codewords (+ 7 remainder bits)

VersionLevel=3-L ⇒ 55 D-words + 15 E-words
VersionLevel=3-M ⇒ 44 D-words + 26 E-words
VersionLevel=3-Q ⇒ 34 D-words + 36 E-words
VersionLevel=3-H ⇒ 26 D-words + 44 E-words

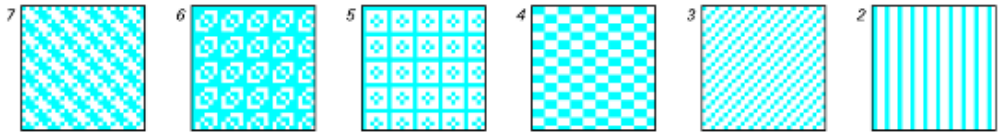
Raw format info (level, mask) also masked ("cooked") — not by the D/E mask (below), but by this special 15-bit ***Format Mask****:*
$1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0_2$

Mask: *Raw* D/E (*not metadata*) bits XOR-*masked ("cooked")* with **mask template**
Goal: Robustness for reading by scanning devices; balance light/dark modules; minimize occurrence of *pro forma* patterns in D/E spaces
Robustness evaluation rubric defined at ISO/IEC spec, §6.8.1
D/E mask templates ('/' ≙ integer divison; '%' ≙ integer remainder):

Mask $000_2=0$ ≙ `(row + col)%2 = 0`
Mask $001_2=1$ ≙ `row%2 = 0`
Mask $010_2=2$ ≙ `col%3 = 0`
Mask $011_2=3$ ≙ `(row + col)%3 = 0`
Mask $100_2=4$ ≙ `(row/2 + col/3) % 2 = 0`
Mask $101_2=5$ ≙ `(row•col)%2 + (row•col)%3 = 0`
Mask $110_2=6$ ≙ `((row•col)%2 + (row•col)%3)%2 = 0`
Mask $111_2=7$ ≙ `((row+col)%2 + (row•col)%3)%2 = 0`



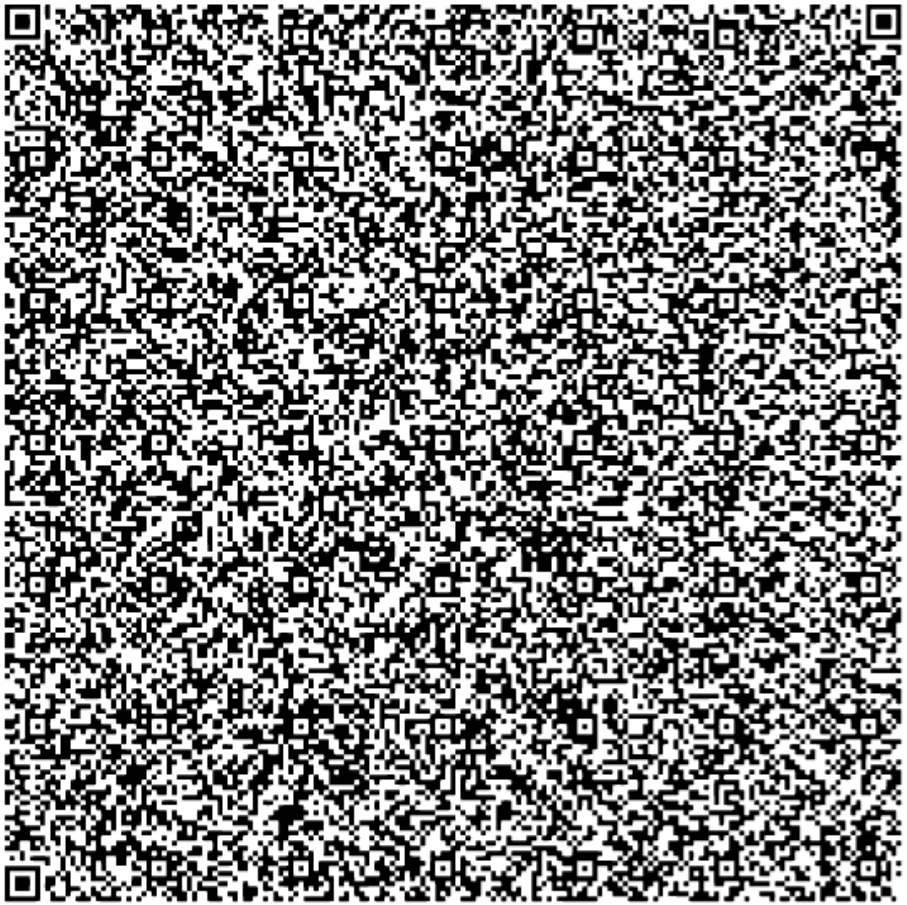
Mask templates (Version 3, 29×29)



2. How does it work

QR code

- Try it yourself



READERS

5. Readers

- A **barcode reader** (or **barcode scanner**) is an electronic device for reading printed barcodes. It consists of a light source, a lens and a light sensor translating **optical** impulses **into electrical** ones.
- Additionally, nearly all readers contain decoder analyzing the barcode's image data provided by the sensor and sending the barcode's content to the scanner's output port.

5. Readers

Types of barcode readers:

- Pen-type readers
- Laser scanners
- CCD readers
- Camera-based readers
- Omni-directional barcode scanners
- Cell phone cameras
- 3D scanners

BENEFITS

6. Benefits

- Can provide detailed up-to-date information on the business, accelerating decisions and with more confidence. For example:
  - Fast-selling items can be identified quickly and automatically reordered.
  - Slow-selling items can be identified, preventing inventory build-up.
  - The effects of merchandising changes can be monitored, allowing fast-moving, more profitable items to occupy the best space,
  - Historical data can be used to predict seasonal fluctuations
  - Items may be repriced on the shelf to reflect price increases.
  - This technology also enables the profiling of individual consumers, typically through a voluntary registration of discount cards.
- Besides sales and inventory tracking, barcodes are very useful in logistics.


00011491980436
00011491980436

5. References

- History of development of barcode http://www.barcoding.com/information/barcode_history.shtml
- Interviews with inventors http://idhistory.com/videodirectory.html
- Barcodes specification http://mdn.morovia.com/kb/20/, http://www.tec-it.com/en/support/knowbase/symbologies/Default.aspx
- Summary of barcodes http://en.wikipedia.org/wiki/Barcode
- Collection of information about barcodes http://www.adams1.com/newspage.html
- Changing color barcode http://2d-code.co.uk/4d-barcodes/
- All about QR codes http://www.denso-wave.com/qrcode/, en.wikipedia.org/wiki/QR_code