

# QR 码介绍

## 一、初识 QR 码

Qr 码属于矩阵式二维码中的一个种类，由 DENSO(日本电装)公司开发，由 JIS 和 ISO 将其标准化。QR 码的样子其实在很多场合已经能够被看到了，贴个图展示一下：



## 二、QR 码的特点

### 1、高速读取

QR 就是取自“Quick Response”的首字母，对读取速度的体验源自于我手机上的一个软件，象上面贴出的码图，通过摄像头从拍摄到解码到显示内容也就三秒左右，对摄像的角度也没有什么要求。

### 2、高容量、高密度

理论上内容经过压缩处理后可以存 7089 个数字，4296 个字母和数字混合字符，2953 个 8 位字节数据，1817 个汉字。

### 3、支持纠错处理

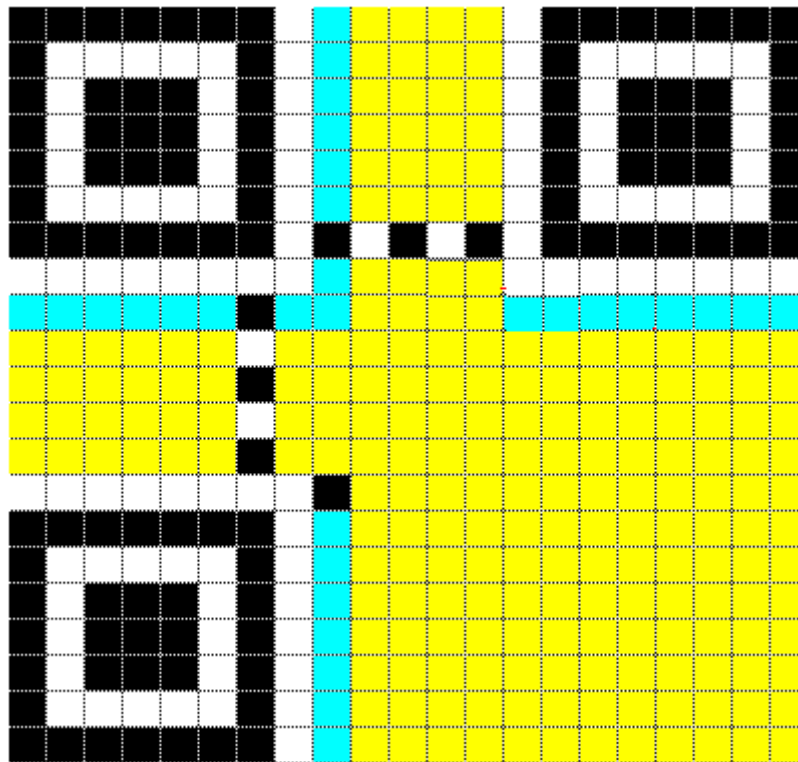
纠错处理相对复杂，目前我还没有深入了解，按照 QR 码的标准文档说明，QR 码的纠错分为 4 个级别，分别是：

- level L :最大 7% 的错误能够被纠正；
- level M :最大 15% 的错误能够被纠正；
- level Q :最大 25% 的错误能够被纠正；
- level H :最大 30% 的错误能够被纠正；

### 4、结构化数据

看似无规则的图形，其实对区域有严格的定义，下图就是一个模式 2、版本 1 的 QR 图结构(关于

QR 码的"模式"、"版本"将在后面进行介绍)：



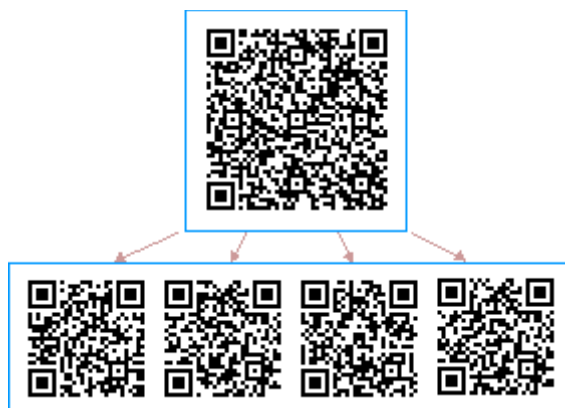
在上图 21\*21 的矩阵中，黑白的区域在 QR 码规范中被指定为固定的位置，称为寻像图形 (finder pattern) 和 定位图形(timing pattern)。寻像图形和定位图形用来帮助解码程序确定图形中具体符号的坐标。

黄色的区域用来保存被编码的数据内容以及纠错信息码。

蓝色的区域，用来标识纠错的级别（也就是 Level L 到 Level H)和所谓的"Mask pattern",这个区域被称为“格式化信息” (format information) 。

## 5、扩展能力

QR 码的 Structure Append 特点，使一个 QR 码可以分解成多个 QR 码，反之，也可以将多个 QR 码的数据组合到一个 QR 码中来。



三、QR 码的模式和版本

QR 码分为 Model1 和 Model2 两种模式，Model1 是对 QR 的初始定义，Model2 是对 Model1 的扩展，目前使用较为普遍的是 Model2，本文的所有说明也仅用于 Model2。QR 图的大小(size)被定义为版本 (Version)，版本号从 1 到 40。版本 1 就是一个 21\*21 的矩阵，每增加一个版本号，矩阵的大小就增加 4 个模块(Module)，因此，版本 40 就是一个 177\*177 的矩阵。（版本越高，意味着存储的内容越多，纠错能力也越强）。

四、QR 码支持的编码内容

QR 码支持编码的内容包括纯数字、数字和字符混合编码、8 位字节码和包含汉字在内的多字节字符。其中：

- 数字：每三个为一组压缩成 10bit。
- 字母数字混合:每两个为一组，压缩成 11bit。
- 8bit 字节数据：无压缩直接保存。
- 多字节字符：每一个字符被压缩成 13bit。

五、最大容量

QR 码的最大容量取决于选择的版本、纠错级别和编码模式 (Mode:数字、字符、多字节字符等)。以版本 1、纠错级别为 Level Q 的 QR 码为例，可以存储 27 个纯数字，或 17 个字母数字混合字符或 11 个 8bit 字节数据。如果要存储同样多的内容同时提高纠错级别，则需要采用更高的 版本。版本 1~9 数据容量、纠错码容量对照如下表：

版本号	容错级别编码后数据量 (字节)	编码后数据量 (字节)	容错数据 (字节)	(numeric)	(alphanumeric)	8bit
1	L	19	7	41	25	17
	M	16	10	34	20	14
	Q	13	13	27	16	11
	H	9	17	17	10	7
2	L	34	10	77	47	32
	M	28	16	63	38	26
	Q	22	22	48	29	20
	H	16	28	34	20	14
3	L	55	15	127	77	53
	M	44	26	101	61	42
	Q	34	36	77	47	32
	H	26	44	58	35	24
4	L	80	20	187	114	78
	M	64	36	149	90	62

	Q	48	52	111	67	46
	H	36	64	82	50	34
5	L	108	26	255	154	106
	M	86	48	202	122	84
	Q	62	72	144	87	60
	H	46	88	106	64	44
6	L	136	36	322	195	134
	M	108	64	255	154	106
	Q	76	96	175	108	74
	H	60	112	139	84	58
7	L	156	40	370	224	154
	M	124	72	293	178	122
	Q	88	108	207	125	86
	H	66	130	154	93	64
8	L	194	48	461	279	192
	M	154	88	365	221	152
	Q	110	132	259	157	108
	H	86	156	202	122	84
9	L	232	60	552	335	230
	M	182	110	432	262	180
	Q	132	160	312	189	130
	H	100	192	235	143	98

如果要了解更详细的 QR 码容量信息，可以到电装的网站去看看 <http://www.denso-wave.com/qrcode/vertable1-e.html>。

下面，就举例说明将“ABCDE123”转换成为版本 1、Level H 的 QR 码转换方法。

#### 一、确认模式标识符

QR 码的模式 (Mode)就是前文提到的数字、字符、8bit 字节码、多字节码等。对于不同的模式，都有对应的模式标识符 (Mode Indicator)来帮助解码程序进行匹配,模式标识符是 4bit 的二进制数：

- 1、数字模式 (numeric mode ) : 0001
- 2、混合字符模式 (alphanumeric mode) : 0010
- 3、8bit byte 模式: 0100
- 4、日本汉字 (KANJI mode) : 1000
- 5、中国汉字(GB2312):1101

由于示例文本串是混合字符，因此将选择混合字符模式，其标识码为：0010

#### 二、确认文本串计数标识符 (Character count indicator)

文本串计数标识符用来存储源内容字符串的长度，在版本 1-9 的 QR 码中，文本串长度标识符自

身的长度被定义为：

数字: 10bit

混合字符 : 9bit

8bit 字节码 : 8bit

多字节码 : 8bit

在本例中，源文本串的长度为 8 个字符，混合字符的长度为 9bit，因此将字符个数 8 编码为 9 位二进制表示：000001000。加上混合字符模式标识码，总的编码为 0010 000001000

三、数据内容编码

1、数字模式下的编码

在数字模式下，数据被限制为 3 个数字一段，分成若干段。如："123456" 将分成"123"和"456"，分别被编码成 10bit 的二进制数。“123”的 10bit 二进制表示法为：0001111011，实际上就是二进制的 123。当数据的长度不足 3 个数字时，如果只有 1 个数字则用 4bit，如果有 2 个数字就用 7 个 bit 来表示。如："9876"被分成"987"和"6"两段，因此被表示为"1111011011 0110"。

2、混合字符模式下的编码

混合字符模式编码，其字符对照表如下：

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

编码方式为：

源码被分成两个字符一段，如下所示，每段的第一个字符乘上 45，再用第二个数字相加。因此每段变成了 11bit 的 2 进制码，如果字符个数只有 1 个，则用 6bit 表示。

示例：

模式	长度	"AB"	"CD"	"E1"	"23"
		45*10+11	45*12+13	45*14+1	45*2+3
		461	553	631	93
0010	000001000	00111001101	01000101001	01001110111	00001011101

3、8bit 字节数据不经编码转换直接保存。

4、中文汉字转换方法。

对于第一字节为 0xA1~0xAA 之间,第二字节在 0xA1~0xFE 之间字符：

- a)第一字节减去 0xA1；
- b)上一步结果乘以 0x60;
- c) 第二字节减去 0xA1;
- d)将 b)步骤的结果加上 c 步骤的结果;
- e)将结果转换为 13 位二进制串。

对于第一字节为 0xB0~0xFA 之间,第二字节在 0xA1~0xFE 之间字符：

- a)第一字节减去 0xA6；
- b)上一步结果乘以 0x60;
- c) 第二字节减去 0xA1;
- d)将 b)步骤的结果加上 c 步骤的结果;
- e)将结果转换为 13 位二进制串。

#### 四、添加编码终止符(Terminator)

如果编码后的字符长度不足当前版本和纠错级别所存储的容量，则在后续补"0000",如果容量已满则无需添加终止符。此时得到的编码串为：

0010 000001000 00111001101 01000101001 01001110111 00001011101 **0000**

#### 五、编成 8bit 码字(Code words)

将以上的编码再按 8bit 一组，形成码字(code words):

00100000 01000001 11001101 01000101 00101001 11011100 00101110 10000

如果尾部数据不足 8bit,则在尾部充 0:

00100000 01000001 11001101 01000101 00101001 11011100 00101110 10000**000**

如果编码后的数据不足版本及纠错级别的最大容量，则在尾部补充 "11101100"和 "00010001",直到全部填满。最后，版本 1、Level H 下的"ABCDE123" 的 QR 码是：

00100000 01000001 11001101 01000101 00101001 11011100 00101110 10000000 **11101100**

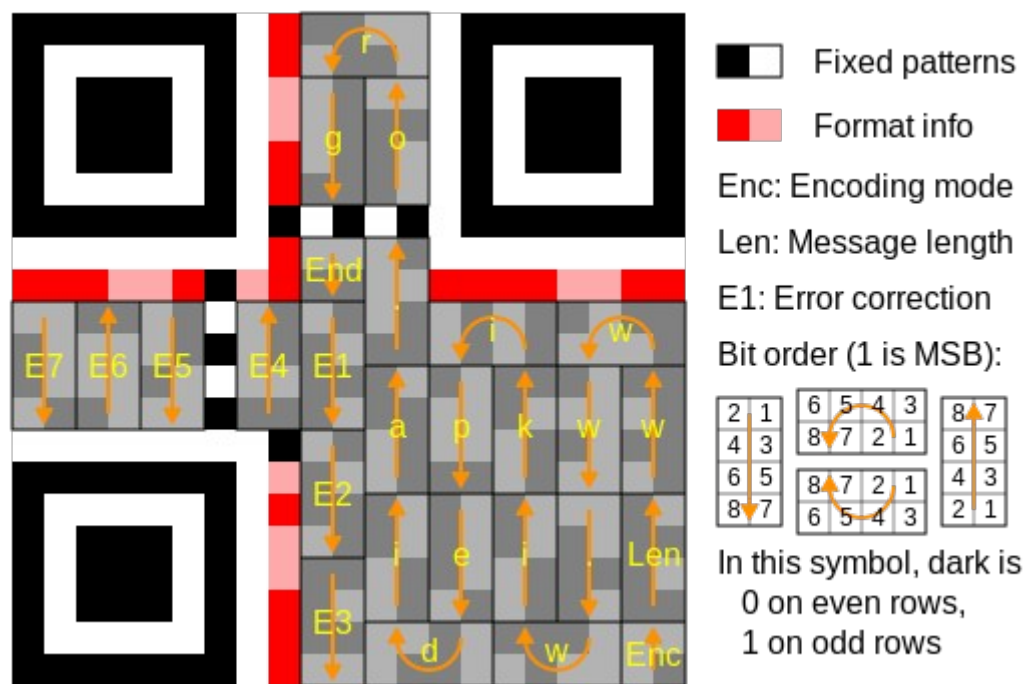
十进制表示法为：

32 65 205 69 41 220 46 128 236

六、添加纠错码

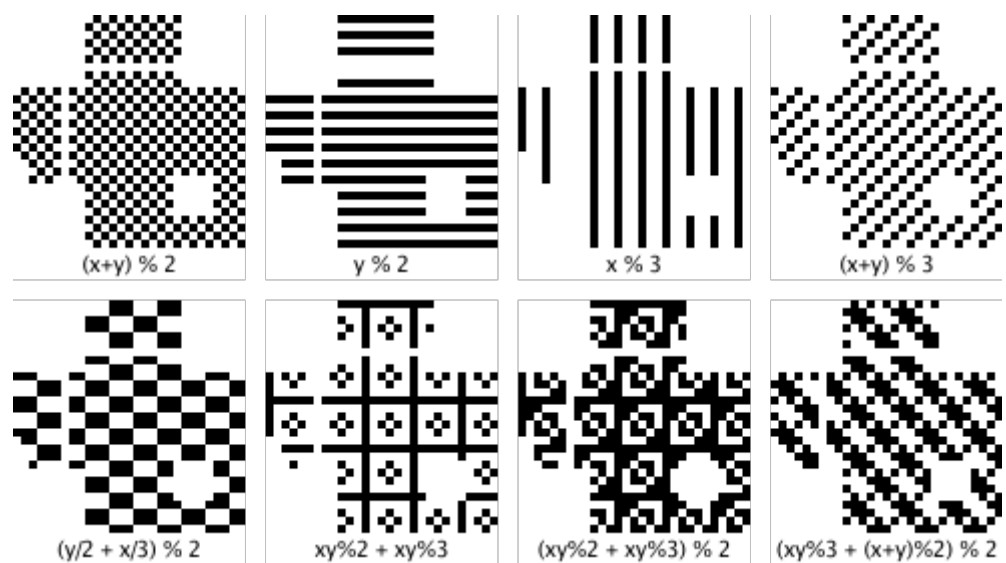
略

七、写入原始数据



八、和掩码模版进行异或运算生成最终的数据

掩码模版



最后结果：

