

入学年度	学部	学 科	組	番 号	検	フリガナ	
2	3	B		1		氏名	

学籍番号を実際に QR コードにしてみよう。

● QR コードの型

中大の学籍番号は大文字のみのアルファベットと数字あわせて 11 文字からなるので「英数字モード」を用いることにする。この文字数なら誤り訂正レベルが高めの「レベル Q」を選択しても 1 型 (21 × 21 セル) に収まるので、1-Q 型を選ぶことにする。また、マスクパターンは本来出来上がったデータの上に掛けてみて一番良いものを選ぶべきなのだが、ここではそれを省略し予め市松模様のパターン (000 型) をかけることに決めておく。

- 型番 (バージョン) : 1 (21 × 21 セル)
- 誤り訂正レベル : Q (2 進指示子 11, 復元能力 25%)
- マスクパターン : 000 (市松模様)
- データモード : 0010 (英数字モード)

● 形式情報

誤り訂正レベルとマスクパターンは「形式情報」としてタイミングパターンのすぐそばに格納される。今の場合、11000 を BCH(15, 5) で符号化し、さらに 101010000010010 というマスクを掛けてつくられる。結果は次のようになる。

011010101011111

● データの bit 列化

さて、次にデータ領域に配置するデータを準備する。まず、最初の 4bit はモードを指示するためのもので、学籍番号では英数字モードを用いるので 0010 とする。次に文字数を指示する必要があるが、英数字モードの場合の最大格納文字数の関係から、これを 9bit で表す。中大の学籍番号は 11 文字なので、これを 2 進法で表示すると 1011 であるが、これに 0 を加えて 9bit 化し 000001011 とする。

そして、いよいよ実際のデータを bit 列になおす。英数字モードではまず下表の通りに各文字を数字化する。なぜ 45 文字が使用可能かという点、 $45^2 = 2025 \ni 2048 = 2^{11}$ なので、2 文字の組を 11bit で表すことができ、効率よく符号化できるからである。そこで、データを 2 文字ずつに区切り、1 つ目の文字の下の表の値を 45 倍したものと 2 つ目の文字の表の値を足す。(2 文字の並びは「45 進法」で表されていると考え、これを 10 進法に直すことに相当する。)

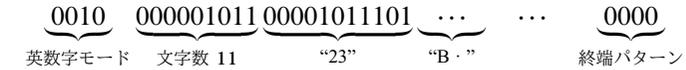
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		\$	%	*
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
+	-	.	/	:															
40	41	42	43	44															

たとえば、「23」という 2 文字は $2 \times 45 + 3 = 93$ という 10 進数で表され、「B2」は $11 \times 45 + 2 = 497$ となる。なお文字数が奇数の場合は最後に残った 1 文字は対応する値をそのままの値とする。さらに、こうして計算された数を 11bit の 2 進数で表す。たとえば、「93」は 2 進法で 1011101 だが、これを 11bit に

するために最初に 0 をいくつか加え、00001011101 とする。以下、これをこれを続け、文字数が奇数の場合の最後に残った 1 文字は対応する値を 2 進法で表し、6 bit で表記する。

学生証番号	2	3	B								
「10 進法化」	93										
11bit 化	00001011101										

すべてを bit 化したら、最後に終端パターンとして 0000 を付加する。



こうして得られたデータを 8bit ごと (1byte ごと) に区切り直す。最後のビット列が 8bit 未満の場合は 0 で埋める。また、1-Q 型では RS(26, 13, 6) 符号を用いるので、得られた byte 数が情報 byte 数である 13 に満たない場合は "11101100" および "00010001" という「埋め草パターン」を交互に付加する。

英数字モード	文字数 11						"23"															
0	0	1	0	0	0	0	0	0	1	0	1	1										
"B."																						
終端パターン				0 fill		埋め草パターン 1				埋め草パターン 2												
0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	1		
埋め草パターン 1																						
1	1	1	0	1	1	0	0															

このようにして、13byte からなる情報語を得る。

	8bit データ							
1.	0	0	1	0	0	0	0	0
2.	0	1	0	1	1	0	0	0
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								
11.								
12.								
13.	1	1	1	0	1	1	0	0