

いよいよ学籍番号を示す QR コードを実際につくる。

● QR コードの型

前回説明したとおり，中大の学籍番号は「英数字モード」で 11 文字からなるので，誤り訂正レベルが高めの「レベル Q」を選択しても，最も小さい 1 型（21 × 21 セル）に収まる．従って，1-Q 型の QR コードを作成することにする．

● マスク

出来上がった QR コードの黒色の部分と白色の部分の極端に偏っていると，読み取りエラーが起こる恐れがあるので，マスクパターンと呼ばれる 8 種類のパターンのいずれかを適用してデータ領域の白黒比率を均一化し，特定のパターン（黒一色など）の発生を抑制する．本来，出来上がったデータの上に 8 つあるパターンを逐次重ね合わせ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．

● 形式情報

QR コード読み取り機器は，2 次元コードを読み取ったとき，それが QR コードであることを認識し，その型番（大きさ）を把握した後，その中に格納された内容を解読する．そのためには，予めマスクパターンと誤り訂正レベルを知らなければならない．そのような「形式情報」と呼ばれる情報はタイミングパターンのすぐそばに BCH 符号を用いて格納される．ここでは，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

● RS 符号による符号化

1Q 型には 13 byte（8 bit の組 13 個）の情報からその倍の長さの 26 byte の符号語が格納されている．QR コードでは RS 符号と呼ばれる符号を用いて，符号語を  $\mathbb{F}_{256}$  を係数とする 25 次式として扱う．前回，各自の学籍番号から 26 byte の符号語を作った．

● マスク処理

マスク処理はマスクパターンに対応した 8 bit データを送信語の bit データに各 bit ごとに足し合わせていくことによってなされる．ここで「足す」とは， $\mathbb{F}_2$  における加法を用いて足す，すなわち  $1 + 1 = 0$  として計算する．このような演算は「排他的論理和」と呼ばれる．

前回，Mathematica を用いて作成された 26 byte の送信語を右の上の表の一番上の行に書き込み，すぐ下のマスクとの排他的論理和をとる，すなわち 0 と 1 を  $\mathbb{F}_2$  のルールに従って計算する．

入学年度	学部	学 科	組	番 号	検	フリガナ	
	B	1				氏 名	

位置	1										2										3										4										5																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
送信語																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

ここまでの結果をもとに下の図のマス目を黒く塗っていく．

																18-2	18-1	17-8	17-7																
																18-4	18-3	17-6	17-5																
																18-6	18-5	17-4	17-3																
																18-8	18-7	17-2	17-1																
																19-2	19-1	16-8	16-7																
																19-4	19-3	16-6	16-5																
																19-6	19-5	16-4	16-3																
																19-8	19-7	16-2	16-1																
26-2	26-1	25-8	25-7	24-2	24-1			23-8	23-7	20-2	20-1	15-8	15-7	10-2	10-1	9-8	9-7	4-2	4-1	3-8	3-7														
26-4	26-3	25-6	25-5	24-4	24-3			23-6	23-5	20-4	20-3	15-6	15-5	10-4	10-3	9-6	9-5	4-4	4-3	3-6	3-5														
26-6	26-5	25-4	25-3	24-6	24-5			23-4	23-3	20-6	20-5	15-4	15-3	10-6	10-5	9-4	9-3	4-6	4-5	3-4	3-3														
26-8	26-7	25-2	25-1	24-8	24-7			23-2	23-1	20-8	20-7	15-2	15-1	10-8	10-7	9-2	9-1	4-8	4-7	3-2	3-1														
																21-2	21-1	14-8	14-7	11-2	11-1	8-8	8-7	5-2	5-1	2-8	2-7								
																21-4	21-3	14-6	14-5	11-4	11-3	8-6	8-5	5-4	5-3	2-6	2-5								
																21-6	21-5	14-4	14-3	11-6	11-5	8-4	8-3	5-6	5-5	2-4	2-3								
																21-8	21-7	14-2	14-1	11-8	11-7	8-2	8-1	5-8	5-7	2-2	2-1								
																22-2	22-1	13-8	13-7	12-2	12-1	7-8	7-7	6-2	6-1	1-8	1-7								
																22-4	22-3	13-6	13-5	12-4	12-3	7-6	7-5	6-4	6-3	1-6	1-5								
																22-6	22-5	13-4	13-3	12-6	12-5	7-4	7-3	6-6	6-5	1-4	1-3								
																22-8	22-7	13-2	13-1	12-8	12-7	7-2	7-1	6-8	6-7	1-2	1-1								

漢字モードのQRコードを作ろう

QRコードで漢字をコード化する仕組みを詳しく知るために、自分の名前をQRコードにしてみよう。

● Shift\_JISコードの概略

0と1のみを扱うコンピュータ内部では、数は10進法ではなく2進法で表現した方が都合がよい。しかし、プログラミングの際に数を2進法で表すと桁数が大きくなりすぎて人間にとっては理解しづらくなってしまう。そこで、プログラミングなどの際には16進法を用いて数を表すことが一般的である。1 byte = 2<sup>8</sup> = 16<sup>2</sup> なので、16進法で2桁までの数を1 byte で表すことができ、2 byte を用いれば16進法で4桁までの数を表現できる。すなわち、2<sup>16</sup> = 65,535 字の文字を2 byte を用いて表すことができる。一方、常用漢字は約2,000字程度であり、さらに使用頻度の低いJIS第4水準漢字までをすべて含めても約11,000字程度であって、かなりの空き領域ができる。Shift\_JISコードは、コンピュータ上で日本語を含む文字列を表現するために用いられる文字コードの一つであるが、JIS第2水準漢字までの漢字など6,879文字が含まれる。これは2<sup>13</sup> = 8,192より少ないので、すべて表すには2 byte = 16 bit も必要はなく、13 bit あれば十分である。そこで、QRコードでは無駄をなくすために、漢字1文字を2 byte = 16 bit ではなく13 bit の2進数で表示することにより、情報を圧縮して無駄をなくしている。

Mathematicaで“草”のShift\_JISコードを求めるには“ToCharacterCode”というコマンドを用いるとよい。しかし、ToCharacterCode["草","ShiftJIS"]とすると、結果が10進法で表示されるので、16進法で表示するために次のようにする。

```
BaseForm[ToCharacterCode["草","ShiftJIS"],16]
{9116,9016}
```

Shift\_JISコードの漢字は第1 byteを縦軸に、第2 byteを横軸にとった仮想の正方形の中で、2つの長方形の領域内に含まれている。第1の長方形は第1 byteが81<sub>16</sub>から9F<sub>16</sub>、第2 byteが40<sub>16</sub>からFC<sub>16</sub>、第2の長方形は第1 byteがE0<sub>16</sub>からEB<sub>16</sub>、第2 byteが40<sub>16</sub>からFC<sub>16</sub>の長方形である。

		0番	1番	2番	...	188番
	第2 byte	40 <sub>16</sub>	41 <sub>16</sub>	42 <sub>16</sub>	...	FC <sub>16</sub>
第1 byte						
0番	81 <sub>16</sub>		、	。	...	○
⋮	⋮	⋮				
30番	9F <sub>16</sub>	檪	藥	檻	...	滌
		⋮				
31番	E0 <sub>16</sub>	漾	漓	滷	...	琰
⋮	⋮	⋮				
42番	EB <sub>16</sub>	*	*	*	...	*

圧縮の原理は以下の通り。第1 byteは81<sub>16</sub>を0番とし、9F<sub>16</sub>までを連番で表すと、9F<sub>16</sub>−81<sub>16</sub> = 1E<sub>16</sub> = 30なので、9F<sub>16</sub>は30番となる。漢字コードがE040<sub>16</sub>を超える場合、E0<sub>16</sub>からC1<sub>16</sub>を引くと、E0<sub>16</sub>−C1<sub>16</sub> = 1F<sub>16</sub> = 31となるので、E0<sub>16</sub>が31番、E1<sub>16</sub>が32番、...、EB<sub>16</sub>が42番となる。一方、第2 byteは40<sub>16</sub>を0番とすると、FC<sub>16</sub>−40<sub>16</sub> = BC<sub>16</sub> = 188だから、FC<sub>16</sub>は188番となる。そこで、漢字コードを次のようにして「192進法化」する。すなわち、

(第1 byteの通し番号 (0~43)) × 192 + (第2 byteの通し番号 (0~188))

を漢字コードの番号とする。この計算はすべて16進法のまま行う。例えば、“草”なら“9190<sub>16</sub>”から“8140<sub>16</sub>”を引くと“1050<sub>16</sub>”となるが、この最初の1 byteにC0<sub>16</sub> = 192を掛け、つぎの byte を足す。すなわち、

10<sub>16</sub> × C0<sub>16</sub> + 50<sub>16</sub> = 0C50<sub>16</sub>

こうして“草”は次のように変換される。

“草” ⇒ 9190<sub>16</sub>  
⇒ 9190<sub>16</sub> − 8140<sub>16</sub> = 1050<sub>16</sub>  
⇒ 10<sub>16</sub> × C0<sub>16</sub> + 50<sub>16</sub> = 0C50<sub>16</sub>  
⇒ 0110001010000<sub>2</sub>

では、“草”、“野”、“美”、“登”、“莉”を「192進法化」してみよう。まず、漢字を対応するShift\_JIS漢字コードに直と、それぞれ、“9190<sub>16</sub>”、“96EC<sub>16</sub>”、“94FC<sub>16</sub>”、“936F<sub>16</sub>”、“E4BB<sub>16</sub>”となる。

氏 名	Shift_JIS コード		差	「192 進法化」	13bit 化	
草	9190 <sub>16</sub>	−	8140 <sub>16</sub>	= 1050 <sub>16</sub>	0C50 <sub>16</sub>	0 1 1 0 0 0 1 0 1 0 0 0 0
野	96EC <sub>16</sub>	−	8140 <sub>16</sub>	= 15AC <sub>16</sub>	106C <sub>16</sub>	1 0 0 0 0 0 1 1 0 1 1 0 0
美	94FC <sub>16</sub>	−	8140 <sub>16</sub>	= 13BC <sub>16</sub>	0EFC <sub>16</sub>	0 1 1 1 0 1 1 1 1 1 1 0 0
登	936F <sub>16</sub>	−	8140 <sub>16</sub>	= 122F <sub>16</sub>	0DAF <sub>16</sub>	0 1 1 0 1 1 0 1 0 1 1 1 1
莉	E4BB <sub>16</sub>	−	C140 <sub>16</sub>	= 237B <sub>16</sub>	1ABB <sub>16</sub>	1 1 0 1 0 1 0 1 1 1 0 1 1

これを自分の名前でやってみよう。

氏名	Shift_JISコード		差	「192進法化」	13bit化