

入学年度	学部	学 科	組	番 号	検	フリガナ	
	B	1				氏名	

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が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。

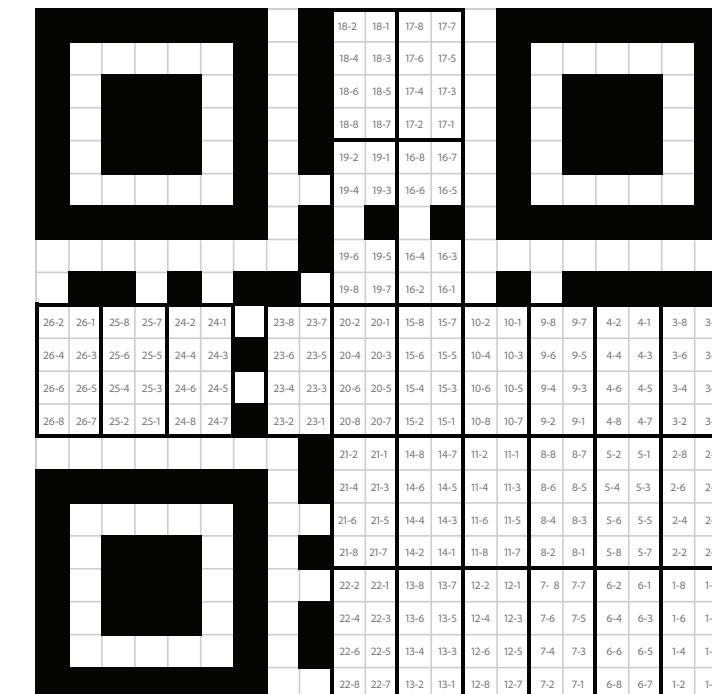
・マスク処理

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。

位置	1	2	3	4	5
送信語					
マスク	1	0	0	1	1
排他的論理和	0	0	1	1	0
6	7	8	9	10	11
0	1	1	0	0	1
1	0	1	0	0	1
0	1	1	0	0	1
12	13	14	15	16	17
0	1	1	0	0	1
1	0	1	0	0	1
0	1	1	0	0	1
18	19	20	21	22	23
1	0	0	1	1	0
0	0	1	0	0	1
1	0	0	1	1	0
0	1	0	0	1	1
24	25	26			
1	0	0	1	1	0
0	0	1	0	1	1
1	0	0	1	1	0
0	1	0	0	1	1

ここまで得た結果をもとに下の図のマス目を黒く塗っていく。



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 = $2^8 = 16^2$ 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、 $2^{16} = 65,535$ 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 $2^{13} = 8,192$ 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 を横軸にし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 81₁₆ から 9F₁₆、第 2 byte が 40₁₆ から FC₁₆、第 2 の長方形は第 1 byte が E0₁₆ から EB₁₆、第 2 byte が 40₁₆ から FC₁₆ の長方形である。

		0 番	1 番	2 番	...	188 番
第 1 byte	第 2 byte	40 ₁₆	41 ₁₆	42 ₁₆	...	FC ₁₆
		81 ₁₆	、	。	...	○
0 番	⋮	⋮	⋮	⋮	⋮	⋮
30 番	9F ₁₆	槩	蘂	檻	...	滌
31 番	E0 ₁₆	漾	漓	滷	...	琰
⋮	⋮	⋮	⋮	⋮	⋮	⋮
42 番	EB ₁₆	*	*	*	...	*

圧縮の原理は以下の通り。第 1 byte は 81₁₆ を 0 番とし、9F₁₆ までを連番で表すと、9F₁₆ - 81₁₆ = 1E₁₆ = 30 なので、9F₁₆ は 30 番となる。漢字コードが E040₁₆ を超える場合、E0₁₆ から C1₁₆ を引くと、E0₁₆ - C1₁₆ = 1F₁₆ = 31 となるので、E0₁₆ が 31 番、E1₁₆ が 32 番、…、EB₁₆ が 42 番となる。一方、第 2 byte は 40₁₆ を 0 番とすると、FC₁₆ - 40₁₆ = BC₁₆ = 188 だから、FC₁₆ 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、

$$(\text{第 1 byte の通し番号 (0~43)}) \times 192 + (\text{第 2 byte の通し番号 (0~188)})$$

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 “9190₁₆” から “8140₁₆” を引くと “1050₁₆” となるが、この最初の 1byte に C0₁₆ = 192 を掛け、つぎの byte を足す。すなわち、

$$10_{16} \times C0_{16} + 50_{16} = 0C50_{16}$$

こうして“草”は次のように変換される。

$$\begin{aligned} \text{“草”} &\Rightarrow 9190_{16} \\ &\Rightarrow 9190_{16} - 8140_{16} = 1050_{16} \\ &\Rightarrow 10_{16} \times C0_{16} + 50_{16} = 0C50_{16} \\ &\Rightarrow 0110001010000_2 \end{aligned}$$

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 漢字コードに直す、それぞれ、“9190₁₆”, “96EC₁₆”, “94FC₁₆”, “936F₁₆”, “E4BB₁₆” となる。

氏名	Shift_JIS コード	差	「192 進法化」	13bit 化
草	9190 ₁₆ - 8140 ₁₆ = 1050 ₁₆	0C50 ₁₆	0110001010000_0	1100010100000000
野	96EC ₁₆ - 8140 ₁₆ = 15AC ₁₆	106C ₁₆	10000001101100_0	1000000110110000
美	94FC ₁₆ - 8140 ₁₆ = 13BC ₁₆	0EFC ₁₆	011101111110_0	01110111111000
登	936F ₁₆ - 8140 ₁₆ = 122F ₁₆	0DAF ₁₆	01101101011111_1	0110110101111111
莉	E4BB ₁₆ - C140 ₁₆ = 237B ₁₆	1ABB ₁₆	1101010111011_1	110101011101111

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