計算して作る QR コード

M.S.

2025 年 4 月 16 日

1　QR コードは株式会社デンソーウェーブの登録商標です

> QR コード（キューアールコード）は、1994 年（平成 6 年）に日本・愛知県の自動車部品メーカーであるデンソーの開発部門（現在は分社化してデンソーウェーブ）が発明したマトリックス型二次元コードである [1][2]。データ読み取りや店頭決済用コードとして世界中で多用されている。「QR」は Quick Response の頭字語であり、高速読み取りを目的の一つとしている名称である。「QR コード」はデンソーウェーブの登録商標（日本第 4075066 号 [3]）である [注 1][注 2]。
>
> ——Wikipedia

みなさんは QR コードを「計算して（作成サイトやアプリを使わずに）」作ったことはあるでしょうか。ただの黒と白の並びに見える QR コードにも、実は様々な工夫や奥の深い数学が隠されています。今回は QR コードの白黒を自分で計算することによってその深淵の一部を覗いてみたいと思います。果てしないモジュロ 100011101 演算を乗り越えたとき、QR コードはその三つ目でにっこりと覗き返してくれることでしょう。

私はコロナ禍が始まってすぐの頃、中 3 〜高 1 の春休みに手計算で QR コードが作れることを知り、実際にやってみました。その時は計算を間違えて読み取れませんでしたが、翌年に再挑戦して読み取りに成功しました。また、仕組みを知れば紙と鉛筆で読み取るのは作るより遥かに簡単です。そのときも身の回りで読みやすそうな QR コードを見つけては、方眼ノートに写して人力で読み取って遊んでいました。これはかなり楽しいのでおすすめです。

作り方までは理解しなくとも、この記事を読んで身の回りの QR コードの模様を気にしてみたり、紙と鉛筆での読み取りに挑戦したりしてもらえるとうれしいです。

用語の使い方には不正確なところがあるかもしれません。それに限らず気になった点がありましたら気軽に言ってください。

2　QR コードの基本

QR コードの規格は JIS X 0510「情報技術－自動認識及びデータ取得技術－ＱＲコード　バーコードシンボル体系仕様」によって定められています。購入しようとすると 6000 円ぐらいかかりますが、日本工業標準調査会（`https://www.jisc.go.jp/`）のサイトで利用者登録すれば無料で閲覧することができます。

身の回りの QR コードを見てみると、様々な細かさのレベルがあることに気が付きます。QR コードには 1 から 40 までの型番（バージョン）があり、バージョン 1 がもっともシンプルでバージョン 40 が最も複雑で

す。型番の番号が大きいほど多くのデータを格納することができます。私はバージョン 30 ぐらいの QR コードが使われているのを見たことがあります。今回は基本的にバージョン 1 と 2 の QR コードについて説明します。

図 1　何かに見えそうで見えないバージョン 40

QR コードには、数字モード（数字のみ)、英数字モード（大文字アルファベット、数字、一部の半角の記号)、8 ビットバイトモード（大小アルファベット、数字、半角の記号)、漢字モード（ひらがな、カタカナ、漢字など）の（主に）4 種類のモードがあります。読み取るとウェブサイトに飛ぶ QR コードは 8 ビットバイトモードです。そういった QR コードも検索機能があるわけではなく、URL を文字列として格納しています。デバイス側に読み取ったものが URL なら検索するという仕組みがあるのでしょう。

QR コードの特長の一つが誤り訂正能力です。例えば、ある 0 ～ 9 の数字の列を読んだとき「1, 2, * , 4, 6, 2」(*は読み取れなかった）で、最後の数字 2 はそれまでの数字を 10 で割ったあまりを表しているとします。すると最後の「2」のおかげで読み取れなかった*は 9 だとわかりますよね。QR コードではリード・ソロモン（RS）符号というものが使われており、誤りがどこにあるのか、どう直せば正しくなるのかがわかるようになっています。QR コードの一部が隠れていたり真ん中が SNS のアイコンになっていたりしても読み取れる場合があるのはこのためです。とはいえもちろん限度はあり、4 段階のレベルがあります（表??)。型番、モード、誤り訂正レベルによって、最大入力文字数は表??のように変わります。

QR コードにはいくつか特別な模様があります。3 つある位置検出パターンは目を引きますが、ほかにも位置合わせパターン、タイミングパターン、分離パターン、QR コード全体を囲むクワイエットゾーンがあり、これらを合わせて機能パターンといいます。特にタイミングパターンの点線は意識して見てみるとおもしろいかもしれません。

表 1　誤り訂正レベル

誤り訂正レベル	復元能力
L	約 7%
M	約 15%
Q	約 25%
H	約 30%

表 2　最大入力文字数

型番	モード	L	M	Q	H
1	数字	41	34	27	17
	英数字	25	20	16	10
	8 ビットバイト	17	14	11	7
	漢字	10	8	7	4
2	数字	77	63	48	34
	英数字	47	38	29	20
	8 ビットバイト	32	26	20	14
	漢字	20	16	12	8

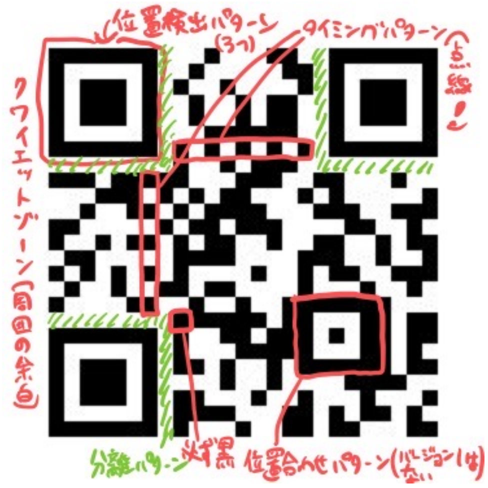

図 2　機能パターン

3　データの符号化

今回はシンプルな例として、英数字モードで「KUALA 2025 /*QR CODE*/」という文字列を英数字モードでバージョン 1、誤り訂正レベル L の QR コードにしてみたいと思います。

まず、表**??**を見てモード指示子を確認します。今回は英数字モードなので 0010 です。

次に文字数を 2 進数で表します。桁数は表**??**に従います。

今回は 22 文字（スペースも 1 文字と数えます）なので、9 桁の 2 進数にして 000010110 です。これを文字

表 3　モード指示子

モード	モード指示子
数字	0001
英数字	0010
8 ビットバイト	0100
漢字	1000

表 4　文字数指示子の桁数

バージョン	数字	英数字	8 ビットバイト	漢字
1 ∼ 9	10	9	8	8
10 ∼ 26	12	11	16	10
27 ∼ 40	14	13	16	12

数指示子と呼びます。

次はいよいよデータの符号化です。英数字モードの場合は次の手順で変換します。まず、データを 2 文字ずつのブロックに分けます。「KU」「AL」「A 」「20」「25」「 /」「*Q」「R 」「CO」「DE」「*/」となります。それぞれの文字を表??にしたがって整数に変換したあと、各ブロックの 1 文字目に対応する整数を 45 倍して 2 文字目に対応する整数と足し算し、11 桁の 2 進数に変換します。最後のブロックに 1 文字しかない場合は対応する数字を 6 桁の 2 進数に変換します。

表 5　英数字モードでの符号化

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

今回は次ページ上部の式のようになります。手計算で 2 進数に変換する方法として、もとの整数を 2 で繰り返し割り（端数は切り捨てる）、出てきた数が偶数か奇数かを逆から（1 から）読む方法があります。偶数なら 0、奇数なら 1 です。

8 ビットバイトモードのときはそれぞれの文字を 8 桁の文字コードで表します。符号化には ISO/IEC 8859-1 を用います。ASCII だと思ってほぼ問題ありません。

モード指示子、文字数指示子、データの各ブロックから作った 2 進数たちをつなげ、8 桁ずつに区切ってデータコード語列を作ります。

00100000 10110011 10100010 00111010 11100111 10011000 00101101 00000101 11111100 11111111
10111101 01100111 00011010 00110100 01001010 11111100 000110

ここで、表??で最終的なデータコード語数を確認しておきましょう。

”KU”	→	$20 \times 45 + 30 = 930$	→	01110100010
”AL”	→	$10 \times 45 + 21 = 471$	→	00111010111
”A ”	→	$10 \times 45 + 36 = 486$	→	00111100110
”20”	→	$2 \times 45 + 0 = 90$	→	00001011010
”25”	→	$2 \times 45 + 5 = 95$	→	00001011111
” /”	→	$36 \times 45 + 43 = 1663$	→	11001111111
”*Q”	→	$39 \times 45 + 26 = 1781$	→	11011110101
”R ”	→	$27 \times 45 + 36 = 1251$	→	10011100011
”CO”	→	$12 \times 45 + 24 = 564$	→	01000110100
”DE”	→	$13 \times 45 + 14 = 599$	→	01001010111
”*/”	→	$43 \times 45 + 43 = 1798$	→	11100000110

表 6　データコード語数

型番	L	M	Q	H
1	19	16	13	9
2	34	28	22	16

今回のデータコード語数は 19 です。ここからさらにデータコード語列を伸ばしていきますが、数字を 8 桁ごとのブロックに区切ったときのブロック（コード語）の数がデータコード語数を超えないように注意しなければいけません。言い換えれば、最終的なメッセージビット列（データコード語列）の桁数はデータコード語数のちょうど 8 倍にする必要があります。

まずは終端パターン 0000 を付け加えます。もし付け加えてデータコード語数を超えるようであれば付け加える 0 の数を減らしてデータコード語数ぴったりになるようにします。

00100000 10110011 10100010 00111010 11100111 10011000 00101101 00000101 11111100 11111111
10111101 01100111 00011010 00110100 01001010 11111100 00011000 00

次に、データコードの桁数が 8 の倍数になるように 0 を付け加えます。すでに 8 の倍数になっている場合は不要です。

00100000 10110011 10100010 00111010 11100111 10011000 00101101 00000101 11111100 11111111
10111101 01100111 00011010 00110100 01001010 11111100 00011000 00000000

これでデータコード語数を満たせばデータコード語列は完成ですが、足りない場合は足りるまで 11101100 と 00010001 を交互に付け加えていきます。これを埋め草コードといいます。

00100000 10110011 10100010 00111010 11100111 10011000 00101101 00000101 11111100 11111111
10111101 01100111 00011010 00110100 01001010 11111100 00011000 00000000 11101100

はい、データコード語列の完成です。ここまでの計算が間違っているとあとで痛い目を見ます。

4　誤り訂正

次にリード・ソロモン（RS）符号を作ります。

唐突ですが、8 桁の数字は α の 7 次多項式の係数とみることができます。例えば 11001001 という数の並びがあった場合、これは係数が 0 か 1 の 7 次式 $\alpha^7+\alpha^6+\alpha^3+1$ と考えます。ここで次のような計算をする世界を考えてみましょう。まず、この世界には 0 か 1 しかありません。足し算は $0+0=0, 0+1=1, 1+0=1, 1+1=0$ となります。常に 2 で割ってあまりのみを見ると考えてもよいでしょう。さらに、$\alpha^8+\alpha^4+\alpha^3+\alpha^2+1=0$ と決めることにします。この式の両辺に α^8 を足すと $\alpha^8=\alpha^4+\alpha^3+\alpha^2+1$ です。ここで $1+1=0$ より $\alpha^8+\alpha^8=0$ となることを用いました。つまり α^8 は α の 7 次以下の多項式に分解することができます。そして $\alpha^8=\alpha^4+\alpha^3+\alpha^2+1$ を使って α の 8 次以上を分解しながら α の累乗の計算を進めていくと、$\alpha^0(=1)$ から α^{254} までの間に 00000001 から 11111111 までの 255 通りの係数が一度ずつ登場し、$\alpha^{255}=1$ になります。つまり 00000000 から 11111111 までの 256 通りの数の並びと「$0, \alpha^0(=1), \cdots, \alpha^{254}$」は 1 対 1 に対応するのです！ 具体的には付録にある表**??**、表**??**のようになります。○○表現という言い方が正しいのかわかりませんが、ここではそう呼ぶことにします。

この関係を使うと、z の n 次多項式 $g(z)=(z-\alpha^0)(z-\alpha^1)\cdots(z-\alpha^{n-1})$ は表**??**のように展開することができるらしいです。この多項式 $g(z)$ を生成多項式と呼びます。

表 7　生成多項式

n	$g(z)$
7	$z^7+\alpha^{87}z^6+\alpha^{229}z^5+\alpha^{146}z^4+\alpha^{149}z^3+\alpha^{238}z^2+\alpha^{102}z+\alpha^{21}$

RS 符号には行列がわちゃわちゃ出てくるすごそうな理論があるのですがここでは生成に関係ない部分は割愛します。興味がある人は `https://qiita.com/Kta-M/items/6f7049a1e78b1fe7e883` などを読んでみてください。

それではいよいよ計算です。理論と表に圧倒されたかもしれませんが、計算はわかってしまえば単純作業です。計算ミスをしないように気を付けて進めていきましょう。

まず表**??**で誤り訂正コード語数を確認します。今回は誤り訂正コード語数は 7 なので、7 次の生成多項式を使用します。

表 8　誤り訂正コード語数

型番	L	M	Q	H
1	7	10	13	17
2	10	16	22	28

データコード語列の各ブロック（データコード語）を表**??**によって指数表現に変換します。例えば 2 つ目のデータコード語では $10110011(\to 179) \to \alpha^{171}$ です。データコード語は 1 つ目だけ変換すればよいのですが、例示のためデータコード語列全体を変換して指数だけ並べてみます。0 は指数表現できないので $(-)$ と書くことにします。

$$5\ 171\ 209\ 9\ 81\ 17\ 18\ 50\ 168\ 175\ 109\ 110\ 105\ 106\ 37\ 168\ 28\ (-)\ 122$$

ここで指数表現は z の (データコード語数 -1) 次多項式 $f(z)$ の係数と考えることができます。つまり、

$$\begin{aligned} f(z)=&\alpha^5z^{18}+\alpha^{171}z^{17}+\alpha^{209}z^{16}+\alpha^7z^{15}+\alpha^{81}z^{14}+\alpha^{17}z^{13}+\alpha^{18}z^{12}+\alpha^{50}z^{11}+\alpha^{168}z^{10}+\alpha^{175}z^9 \\ &+\alpha^{109}z^8+\alpha^{110}z^7+\alpha^{105}z^6+\alpha^{106}z^5+\alpha^{37}z^4+\alpha^{168}z^3+\alpha^{28}z^2+\alpha^{122} \end{aligned}$$

です。

データコード語列に誤り訂正コード語数の分だけ $00000000 = 0 = (-)$ のブロックを付け加えておきます。これは今回なら $f(z)$ に z^7 を掛けることと同じです。ここからは、この $z^7f(z)$ を生成多項式 $g(z)$ で割った余りを求めていきます。

生成多項式 $g(z)$ に $f(z)$ の最高次の項の係数と z の累乗を掛けて、$z^7f(z)$ と最高次の項が等しくなるようにします。今回は α^5z^{18} を掛けます。このとき $\alpha^{255}=1$ の関係を利用すると

$$g(z) = z^7 + \alpha^{87}z^6 + \alpha^{229}z^5 + \alpha^{146}z^4 + \alpha^{149}z^3 + \alpha^{238}z^2 + \alpha^{102}z + \alpha^{21}$$
$$\alpha^5z^{18}g(z) = \alpha^5z^{25} + \alpha^{92}z^{24} + \alpha^{234}z^{23} + \alpha^{151}z^{22} + \alpha^{154}z^{21} + \alpha^{243}z^{19} + \alpha^{107}z^{18} + \alpha^{26}z^{17}$$

となります。これを $z^7f(z)$ から引くために、$z^7f(z)$ と $\alpha^5z^{18}g(z)$ から係数を取り出し、2 進数表現して上下に並べて書きます。$z^7f(z)$ はデータコード語列そのままで、$\alpha^5z^{18}g(z)$ は表**??**を使って指数表現に変換します。

$$z^7f(z) \to 00100000\ 10110011\ 10100010\ 00111010\ \cdots$$
$$\alpha^5z^{18}g(z) \to 00100000\ 01011011\ 11111011\ 10101010\ \cdots$$

普通の割り算の筆算ならばここで上から下を引きますが、今回は 0 と 1 しかなく $0+0=0, 0+1=1, 1+0=1, 1+1=0$ というルールで計算するのでした。さらに 2 進数表現は多項式の係数を羅列したものであることを思い出すと、ここでは「上下の関係にある 2 つの数のみを見て、2 つが同じならば 0、異なれば 1」というルールにしたがって新しい数列を作っていけばよいことがわかります。すると

$$z^7f(z) \to 00100000\ 10110011\ 10100010\ 00111010\ \cdots$$
$$\alpha^5z^{18}g(z) \to 00100000\ 01011011\ 11111011\ 10101010\ \cdots$$
$$\text{演算の結果} \to 00000000\ 11101000\ 01011001\ 10010000\ \cdots$$

となって、最高次の項が消えました。このように

1. 多項式 A（最初は $z^7f(z)$）の係数を 2 進数表示して 8 桁ごとに（項ごとに）区切りながら並べる。
2. 多項式 A（最初は $z^7f(z)$）の最高次の項の係数を表**??**を使って指数表示する。
3. 2 の多項式と最高次の項が等しくなるように、係数が指数表示の生成多項式に α の累乗と z の累乗を掛ける。$\alpha^{255}=1$ の関係を使って指数が 254 以下になるようにする。
4. 3 の多項式の係数を 2 進数表示して 1 の下に並べる。
5. 1 と 4 の 2 つの数を見て、2 つが同じならば 0、異なれば 1 というルールで新しい数列を作り、これらを係数とする多項式を改めて A とする。最初のブロックは 00000000 になり、多項式 A の次数は 1 以上小さくなったはずである。
6. 新しい多項式 A を使って 2、3、4、5 の操作を繰り返す。多項式 A の次数が (誤り訂正コード語数-1) と等しくなったら終了する。

という操作によって計算を進めていきます。0 と 1 だけの計算ですが非常に計算量が多いです。今回と同じ次数の場合は A3 用紙の横を縦に 20 分割して各列に次数を揃えて係数の 2 進数表示を書いていきながら 3 の多項式の係数の指数は別の紙に書き、2 進数表示が右にはみ出たら紙の左下対角にできる余白で続けていくようにするといいかもしれません。計算は非常に非常に大変ですががんばってください。

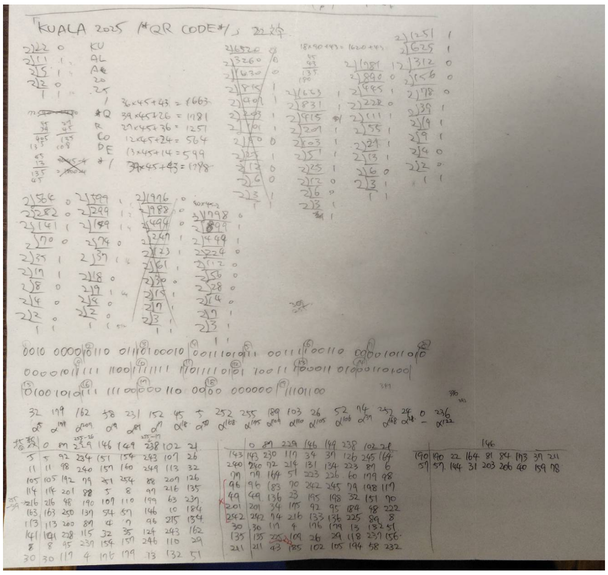
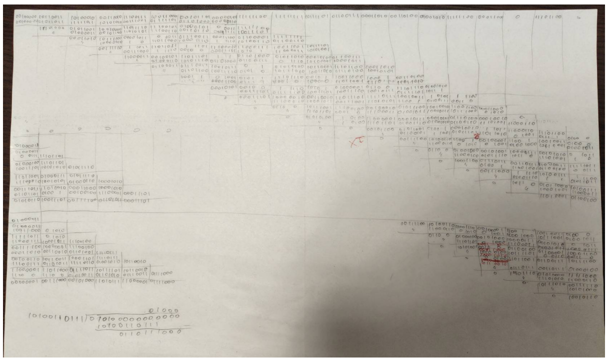

図 3　計算がんばった

この解説を書くにあたって自分でも計算してみました。注意してもいくつも計算間違いをしてしまうので、プログラムを書いて確かめながらやったことを白状しておきます。

この計算を終えてできる多項式の係数の 2 進数表示を並べたものが誤り訂正コード語列です。これを前の章で作ったデータコード語列のあとに続けて書くことで QR コードのデータ部分が完成します。バージョン 2 の場合は余白ができてしまうので、残余ビット 0000000 を付け加えます。

今回のデータコード語列に誤り訂正コード語列を続けて書くと

00100000 10110011 10100010 00111010 11100111 10011000 00101101 00000101 11111100 11111111 10111101 01100111 00011010 00110100 01001010 11111100 00011000 00000000 11101100 10010110 00000001 00111000 00101000 11010111 11000001 01111000

です。

5　マスク処理

それでは作ったデータを空っぽの QR コードに並べていきます。バージョン 1 の場合は図??の順番で赤枠内のモジュールを白 (0) か黒 (1) で埋めます。

こうして並べただけでは、白や黒が広い範囲に連続したり、機能パターンと同じ模様が機能パターン以外のところに現れる可能性があります。こうしたことが起きると読み取り精度が悪くなるので、マスク処理というものを施します。マスク処理で使うマスクパターンには表??の 8 種類があります。条件式の i は行番号（上から $0, 1, \cdots$）、j は列番号（左から同様に）を表し、a div b と a mod b はそれぞれ a を b で割った商（整数）と余りを表します。

マスク処理では、さっきデータを並べた部分のみについて、選んだマスクパターンの条件に合うモジュールの白黒を反転させます。本来は全パターンを試して評価し一番いいのを選ぶのですが、面倒なので今回は一番楽な 010 のマスクパターンを使います。

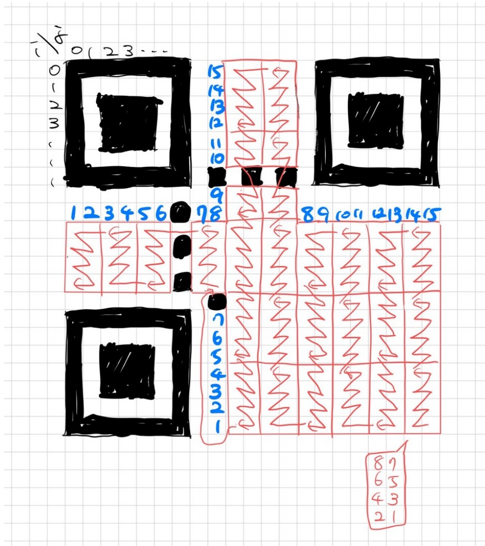

図 4　並べる順番

表 9　マスクパターン

マスクパターン参照子	条件
000	(i+j) mod 2 = 0
001	i mod 2 = 0
010	j mod 3 = 0
011	(i+j) mod 3 = 0
100	((i div 2) + (j div 3)) mod 2 = 0
101	(ij) mod 2 + (ij) mod 3 = 0
110	((ij) mod 2 + (ij) mod 3) mod 2 = 0
111	((i+j) mod 2 + (ij) mod 3) mod 2 = 0

6　形式情報

最後にもう少し計算をします。誤り訂正指示子を表??で確認し、マスクパターン参照子をあとに繋げて 5 桁の数字にします。

表 10　誤り訂正指示子

誤り訂正レベル	誤り訂正指示子
L	01
M	00
Q	11
H	10

今回なら 01010 ですね。これは誤り訂正コード語を作ったときと同じようにして x の 4 次以下の多項式

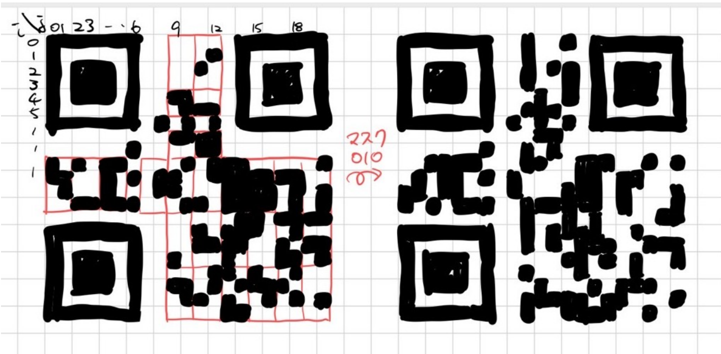

図 5　マスクパターン 010

の係数と考えることができます。今回は x^3+x です。これに x^{10} を掛け、0 と 1 だけの世界で生成多項式 $G(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$ で割った余りを求めます。これは誤り訂正コード語のときほどめんどくさくはなく、係数だけ取り出して普通の割り算の筆算（より正確には多項式の割り算）のように計算できます（図**??**）。

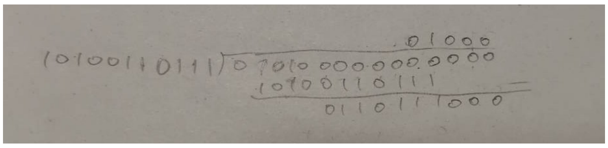

図 6　形式情報の計算

今回の場合、余りの係数を取り出したものは 0110111000 です。これを 00010 にくっつけると 000100110111000 です。次にこれと 101010000010010 を 0 と 1 の世界で足し算し、101110110101010 を得ます。この数字の並びの 0 を白に、1 を黒に対応させ、図**??**の青で 1 から 15 の数字が書いてある 2 箇所を塗ります。$1,\cdots,15$ の順に $1,0,1,1,\cdots,1,0$ と対応させてください。

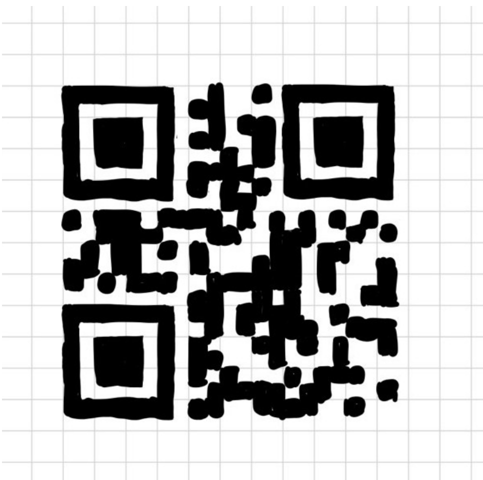

図 7　完成!!

これで読み取れたら完成です！！！ お疲れさまでした！！！

7 Python で計算

到底合っている自信が持てない計算なので、Python でこの計算過程を実装しました。バージョンは 1 と 2 のみ、マスクパターンも上の 4 つのみで、自分の検算用に作ったので変数名とか適当ですが、よかったら使ってください。8 ビットバイトモードには対応しています。URL は予告なく変更する場合があります。
`https://github.com/Peteworden/articles/blob/main/qrcode/qr.py`

Excel でもほぼ同じものを作ることができます。まだ Python を知らない、最初に QR コードの計算をした頃に作りました。Excel で簡単に QR コードを作れる方法もありますが、計算を自分で書いてみるとツールを使うのとは違う楽しさがあります。

8 紙と鉛筆で QR コードを読み取る

以上のことを踏まえると、スマホなどのカメラを使うことなく、紙と鉛筆と文字コードの変換表だけで QR コードを読み取ることができます。まず、バージョンが小さくて左上の位置検出パターンの 1 つ下の行の左から 3 ～ 5 マス目が黒白白（001 のマスク）または黒黒黒（010 のマスク）になっているものを探します。それ以外はマスクを外すのが大変です。紙に QR コードの右半分を写しとったら、マスク処理を逆算したものをその隣に書きます。そして右下からモード、文字数、データの順で読んでいけば、例えば https://... という文字列が浮かび上がるでしょう。誤り訂正符号は計算せずに済むどころか読む必要もありません。こうしていろいろな QR コードを人力で読んでみるのも、たまにはいいのではないでしょうか。

9 おわりに

最後までお読みいただきありがとうございました。QR コードを自分で作ってみたい、または手で読み取ってみたい気分になりましたか？ この解説の執筆中に QuizKnock が似た内容の動画を出していたので、こちらもぜひ見てみてください（`https://youtu.be/PTLLwlL9zF8?si=lakf6x5BfR_ti-tf`）。

QR コードの作り方を最初に学んだときには岡崎高校の高校生が書いたサイト（`https://sites.google.com/a/osshc.co.cc/web/studies/it/qr`）に大変お世話になったのですが、現在は残念ながら見れなくなっています。この PDF ファイルが、岡崎高校のサイトに代わるわかりやすい解説書となればうれしいです。

最後に、今後はこうした PDF を `https://peteworden.github.io/articles/articles.html` にまとめようと思っています。これまでに書いたものも入れておくのでよかったらご覧ください。

付録 A 付録（でかい表）

これらの表は Python で LaTeX を書いて作成しました。

表 11　指数表現から 2 進数表現・整数表現

指数	2 進数	整数	指数	2 進数	整数	指数	2 進数	整数	指数	2 進数	整数
0	00000000	0	α^{63}	10100001	161	α^{127}	11001100	204	α^{191}	01000001	65
α^{0}	00000001	1	α^{64}	01011111	95	α^{128}	10000101	133	α^{192}	10000010	130
α^{1}	00000010	2	α^{65}	10111110	190	α^{129}	00010111	23	α^{193}	00011001	25
α^{2}	00000100	4	α^{66}	01100001	97	α^{130}	00101110	46	α^{194}	00110010	50
α^{3}	00001000	8	α^{67}	11000010	194	α^{131}	01011100	92	α^{195}	01100100	100
α^{4}	00010000	16	α^{68}	10011001	153	α^{132}	10111000	184	α^{196}	11001000	200
α^{5}	00100000	32	α^{69}	00101111	47	α^{133}	01101101	109	α^{197}	10001101	141
α^{6}	01000000	64	α^{70}	01011110	94	α^{134}	11011010	218	α^{198}	00000111	7
α^{7}	10000000	128	α^{71}	10111100	188	α^{135}	10101001	169	α^{199}	00001110	14
α^{8}	00011101	29	α^{72}	01100101	101	α^{136}	01001111	79	α^{200}	00011100	28
α^{9}	00111010	58	α^{73}	11001010	202	α^{137}	10011110	158	α^{201}	00111000	56
α^{10}	01110100	116	α^{74}	10001001	137	α^{138}	00100001	33	α^{202}	01110000	112
α^{11}	11101000	232	α^{75}	00001111	15	α^{139}	01000010	66	α^{203}	11100000	224
α^{12}	11001101	205	α^{76}	00011110	30	α^{140}	10000100	132	α^{204}	11011101	221
α^{13}	10000111	135	α^{77}	00111100	60	α^{141}	00010101	21	α^{205}	10100111	167
α^{14}	00010011	19	α^{78}	01111000	120	α^{142}	00101010	42	α^{206}	01010011	83
α^{15}	00100110	38	α^{79}	11110000	240	α^{143}	01010100	84	α^{207}	10100110	166
α^{16}	01001100	76	α^{80}	11111101	253	α^{144}	10101000	168	α^{208}	01010001	81
α^{17}	10011000	152	α^{81}	11100111	231	α^{145}	01001101	77	α^{209}	10100010	162
α^{18}	00101101	45	α^{82}	11010011	211	α^{146}	10011010	154	α^{210}	01011001	89
α^{19}	01011010	90	α^{83}	10111011	187	α^{147}	00101001	41	α^{211}	10110010	178
α^{20}	10110100	180	α^{84}	01101011	107	α^{148}	01010010	82	α^{212}	01111001	121
α^{21}	01110101	117	α^{85}	11010110	214	α^{149}	10100100	164	α^{213}	11110010	242
α^{22}	11101010	234	α^{86}	10110001	177	α^{150}	01010101	85	α^{214}	11111001	249
α^{23}	11001001	201	α^{87}	01111111	127	α^{151}	10101010	170	α^{215}	11101111	239
α^{24}	10001111	143	α^{88}	11111110	254	α^{152}	01001001	73	α^{216}	11000011	195
α^{25}	00000011	3	α^{89}	11100001	225	α^{153}	10010010	146	α^{217}	10011011	155
α^{26}	00000110	6	α^{90}	11011111	223	α^{154}	00111001	57	α^{218}	00101011	43
α^{27}	00001100	12	α^{91}	10100011	163	α^{155}	01110010	114	α^{219}	01010110	86
α^{28}	00011000	24	α^{92}	01011011	91	α^{156}	11100100	228	α^{220}	10101100	172
α^{29}	00110000	48	α^{93}	10110110	182	α^{157}	11010101	213	α^{221}	01000101	69
α^{30}	01100000	96	α^{94}	01110001	113	α^{158}	10110111	183	α^{222}	10001010	138
α^{31}	11000000	192	α^{95}	11100010	226	α^{159}	01110011	115	α^{223}	00001001	9
α^{32}	10011101	157	α^{96}	11011001	217	α^{160}	11100110	230	α^{224}	00010010	18
α^{33}	00100111	39	α^{97}	10101111	175	α^{161}	11010001	209	α^{225}	00100100	36
α^{34}	01001110	78	α^{98}	01000011	67	α^{162}	10111111	191	α^{226}	01001000	72
α^{35}	10011100	156	α^{99}	10000110	134	α^{163}	01100011	99	α^{227}	10010000	144
α^{36}	00100101	37	α^{100}	00010001	17	α^{164}	11000110	198	α^{228}	00111101	61
α^{37}	01001010	74	α^{101}	00100010	34	α^{165}	10010001	145	α^{229}	01111010	122
α^{38}	10010100	148	α^{102}	01000100	68	α^{166}	00111111	63	α^{230}	11110100	244
α^{39}	00110101	53	α^{103}	10001000	136	α^{167}	01111110	126	α^{231}	11110101	245
α^{40}	01101010	106	α^{104}	00001101	13	α^{168}	11111100	252	α^{232}	11110111	247
α^{41}	11010100	212	α^{105}	00011010	26	α^{169}	11100101	229	α^{233}	11110011	243
α^{42}	10110101	181	α^{106}	00110100	52	α^{170}	11010111	215	α^{234}	11111011	251
α^{43}	01110111	119	α^{107}	01101000	104	α^{171}	10110011	179	α^{235}	11101011	235
α^{44}	11101110	238	α^{108}	11010000	208	α^{172}	01111011	123	α^{236}	11001011	203
α^{45}	11000001	193	α^{109}	10111101	189	α^{173}	11110110	246	α^{237}	10001011	139
α^{46}	10011111	159	α^{110}	01100111	103	α^{174}	11110001	241	α^{238}	00001011	11
α^{47}	00100011	35	α^{111}	11001110	206	α^{175}	11111111	255	α^{239}	00010110	22
α^{48}	01000110	70	α^{112}	10000001	129	α^{176}	11100011	227	α^{240}	00101100	44
α^{49}	10001100	140	α^{113}	00011111	31	α^{177}	11011011	219	α^{241}	01011000	88
α^{50}	00000101	5	α^{114}	00111110	62	α^{178}	10101011	171	α^{242}	10110000	176
α^{51}	00001010	10	α^{115}	01111100	124	α^{179}	01001011	75	α^{243}	01111101	125
α^{52}	00010100	20	α^{116}	11111000	248	α^{180}	10010110	150	α^{244}	11111010	250
α^{53}	00101000	40	α^{117}	11101101	237	α^{181}	00110001	49	α^{245}	11101001	233
α^{54}	01010000	80	α^{118}	11000111	199	α^{182}	01100010	98	α^{246}	11001111	207
α^{55}	10100000	160	α^{119}	10010011	147	α^{183}	11000100	196	α^{247}	10000011	131
α^{56}	01011101	93	α^{120}	00111011	59	α^{184}	10010101	149	α^{248}	00011011	27
α^{57}	10111010	186	α^{121}	01110110	118	α^{185}	00110111	55	α^{249}	00110110	54
α^{58}	01101001	105	α^{122}	11101100	236	α^{186}	01101110	110	α^{250}	01101100	108
α^{59}	11010010	210	α^{123}	11000101	197	α^{187}	11011100	220	α^{251}	11011000	216
α^{60}	10111001	185	α^{124}	10010111	151	α^{188}	10100101	165	α^{252}	10101101	173
α^{61}	01101111	111	α^{125}	00110011	51	α^{189}	01010111	87	α^{253}	01000111	71
α^{62}	11011110	222	α^{126}	01100110	102	α^{190}	10101110	174	α^{254}	10001110	142

表 12　2 進数表現・整数表現から指数表現

整数	2 進数	指数	整数	2 進数	指数	整数	2 進数	指数	整数	2 進数	指数
0	00000000	0	64	01000000	α^{6}	128	10000000	α^{7}	192	11000000	α^{31}
1	00000001	α^{0}	65	01000001	α^{191}	129	10000001	α^{112}	193	11000001	α^{45}
2	00000010	α^{1}	66	01000010	α^{139}	130	10000010	α^{192}	194	11000010	α^{67}
3	00000011	α^{25}	67	01000011	α^{98}	131	10000011	α^{247}	195	11000011	α^{216}
4	00000100	α^{2}	68	01000100	α^{102}	132	10000100	α^{140}	196	11000100	α^{183}
5	00000101	α^{50}	69	01000101	α^{221}	133	10000101	α^{128}	197	11000101	α^{123}
6	00000110	α^{26}	70	01000110	α^{48}	134	10000110	α^{99}	198	11000110	α^{164}
7	00000111	α^{198}	71	01000111	α^{253}	135	10000111	α^{13}	199	11000111	α^{118}
8	00001000	α^{3}	72	01001000	α^{226}	136	10001000	α^{103}	200	11001000	α^{196}
9	00001001	α^{223}	73	01001001	α^{152}	137	10001001	α^{74}	201	11001001	α^{23}
10	00001010	α^{51}	74	01001010	α^{37}	138	10001010	α^{222}	202	11001010	α^{73}
11	00001011	α^{238}	75	01001011	α^{179}	139	10001011	α^{237}	203	11001011	α^{236}
12	00001100	α^{27}	76	01001100	α^{16}	140	10001100	α^{49}	204	11001100	α^{127}
13	00001101	α^{104}	77	01001101	α^{145}	141	10001101	α^{197}	205	11001101	α^{12}
14	00001110	α^{199}	78	01001110	α^{34}	142	10001110	α^{254}	206	11001110	α^{111}
15	00001111	α^{75}	79	01001111	α^{136}	143	10001111	α^{24}	207	11001111	α^{246}
16	00010000	α^{4}	80	01010000	α^{54}	144	10010000	α^{227}	208	11010000	α^{108}
17	00010001	α^{100}	81	01010001	α^{208}	145	10010001	α^{165}	209	11010001	α^{161}
18	00010010	α^{224}	82	01010010	α^{148}	146	10010010	α^{153}	210	11010010	α^{59}
19	00010011	α^{14}	83	01010011	α^{206}	147	10010011	α^{119}	211	11010011	α^{82}
20	00010100	α^{52}	84	01010100	α^{143}	148	10010100	α^{38}	212	11010100	α^{41}
21	00010101	α^{141}	85	01010101	α^{150}	149	10010101	α^{184}	213	11010101	α^{157}
22	00010110	α^{239}	86	01010110	α^{219}	150	10010110	α^{180}	214	11010110	α^{85}
23	00010111	α^{129}	87	01010111	α^{189}	151	10010111	α^{124}	215	11010111	α^{170}
24	00011000	α^{28}	88	01011000	α^{241}	152	10011000	α^{17}	216	11011000	α^{251}
25	00011001	α^{193}	89	01011001	α^{210}	153	10011001	α^{68}	217	11011001	α^{96}
26	00011010	α^{105}	90	01011010	α^{19}	154	10011010	α^{146}	218	11011010	α^{134}
27	00011011	α^{248}	91	01011011	α^{92}	155	10011011	α^{217}	219	11011011	α^{177}
28	00011100	α^{200}	92	01011100	α^{131}	156	10011100	α^{35}	220	11011100	α^{187}
29	00011101	α^{8}	93	01011101	α^{56}	157	10011101	α^{32}	221	11011101	α^{204}
30	00011110	α^{76}	94	01011110	α^{70}	158	10011110	α^{137}	222	11011110	α^{62}
31	00011111	α^{113}	95	01011111	α^{64}	159	10011111	α^{46}	223	11011111	α^{90}
32	00100000	α^{5}	96	01100000	α^{30}	160	10100000	α^{55}	224	11100000	α^{203}
33	00100001	α^{138}	97	01100001	α^{66}	161	10100001	α^{63}	225	11100001	α^{89}
34	00100010	α^{101}	98	01100010	α^{182}	162	10100010	α^{209}	226	11100010	α^{95}
35	00100011	α^{47}	99	01100011	α^{163}	163	10100011	α^{91}	227	11100011	α^{176}
36	00100100	α^{225}	100	01100100	α^{195}	164	10100100	α^{149}	228	11100100	α^{156}
37	00100101	α^{36}	101	01100101	α^{72}	165	10100101	α^{188}	229	11100101	α^{169}
38	00100110	α^{15}	102	01100110	α^{126}	166	10100110	α^{207}	230	11100110	α^{160}
39	00100111	α^{33}	103	01100111	α^{110}	167	10100111	α^{205}	231	11100111	α^{81}
40	00101000	α^{53}	104	01101000	α^{107}	168	10101000	α^{144}	232	11101000	α^{11}
41	00101001	α^{147}	105	01101001	α^{58}	169	10101001	α^{135}	233	11101001	α^{245}
42	00101010	α^{142}	106	01101010	α^{40}	170	10101010	α^{151}	234	11101010	α^{22}
43	00101011	α^{218}	107	01101011	α^{84}	171	10101011	α^{178}	235	11101011	α^{235}
44	00101100	α^{240}	108	01101100	α^{250}	172	10101100	α^{220}	236	11101100	α^{122}
45	00101101	α^{18}	109	01101101	α^{133}	173	10101101	α^{252}	237	11101101	α^{117}
46	00101110	α^{130}	110	01101110	α^{186}	174	10101110	α^{190}	238	11101110	α^{44}
47	00101111	α^{69}	111	01101111	α^{61}	175	10101111	α^{97}	239	11101111	α^{215}
48	00110000	α^{29}	112	01110000	α^{202}	176	10110000	α^{242}	240	11110000	α^{79}
49	00110001	α^{181}	113	01110001	α^{94}	177	10110001	α^{86}	241	11110001	α^{174}
50	00110010	α^{194}	114	01110010	α^{155}	178	10110010	α^{211}	242	11110010	α^{213}
51	00110011	α^{125}	115	01110011	α^{159}	179	10110011	α^{171}	243	11110011	α^{233}
52	00110100	α^{106}	116	01110100	α^{10}	180	10110100	α^{20}	244	11110100	α^{230}
53	00110101	α^{39}	117	01110101	α^{21}	181	10110101	α^{42}	245	11110101	α^{231}
54	00110110	α^{249}	118	01110110	α^{121}	182	10110110	α^{93}	246	11110110	α^{173}
55	00110111	α^{185}	119	01110111	α^{43}	183	10110111	α^{158}	247	11110111	α^{232}
56	00111000	α^{201}	120	01111000	α^{78}	184	10111000	α^{132}	248	11111000	α^{116}
57	00111001	α^{154}	121	01111001	α^{212}	185	10111001	α^{60}	249	11111001	α^{214}
58	00111010	α^{9}	122	01111010	α^{229}	186	10111010	α^{57}	250	11111010	α^{244}
59	00111011	α^{120}	123	01111011	α^{172}	187	10111011	α^{83}	251	11111011	α^{234}
60	00111100	α^{77}	124	01111100	α^{115}	188	10111100	α^{71}	252	11111100	α^{168}
61	00111101	α^{228}	125	01111101	α^{243}	189	10111101	α^{109}	253	11111101	α^{80}
62	00111110	α^{114}	126	01111110	α^{167}	190	10111110	α^{65}	254	11111110	α^{88}
63	00111111	α^{166}	127	01111111	α^{87}	191	10111111	α^{162}	255	11111111	α^{175}