Barcodes – error correction

Identification systems (IDFS)

Department of Control and Telematics
Faculty of Transportation Sciences, CTU in Prague

Contents

- Types of error correction
- Where it is used
- Principles of EC

Types of error correction

Types of error correction and its usage

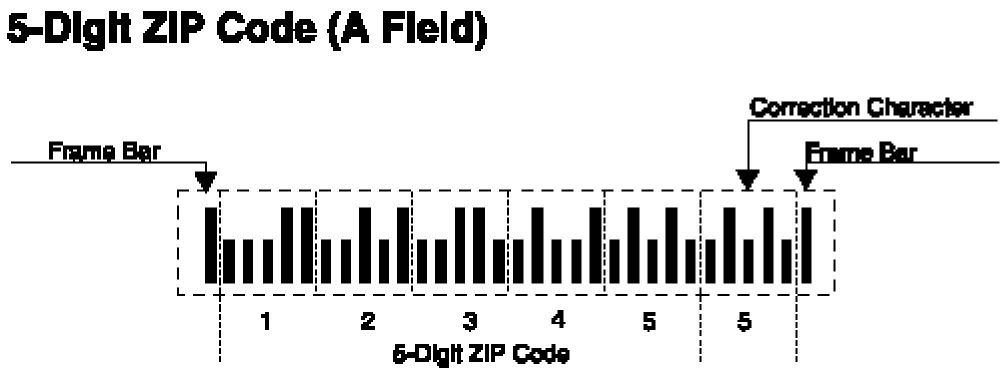
- No correction
  - Some barcodes like 2 of 5
- Simple checksum – no correction just detection
  - 1D barcodes like UPC, EAN etc.
- Self correcting codes (Reed Solomon codes – BCH family)
  - 2D barcodes (PDF417,QR code)

Principles of EC - PostNet

The correction character is always the digit that, when added to the **sum** of the other digits in the barcode, results in a total that is a multiple of 10, so modulo by 10 is zero.

- For example, the sum of the ZIP+4 barcode 12345-6789 is
- 1+2+3+4+5+6+7+8+9 = 45.
- Adding a correction character of 5 results in the
- sum of the 10 digits being a multiple of 10.

- **The check digit is thus 5.**



Principles of EC – UPC-A

The correction character is always the digit calculated in following way:

- For example, in a UPC-A barcode "03600029145x" where x is the unknown check digit, x can be calculated by
- adding the odd-numbered digits (0 + 6 + 0 + 2 + 1 + 5 = 14),
- multiplying by three (14 × 3 = 42),
- adding the even-numbered digits (42 + (3 + 0 + 0 + 9 + 4) = 58),
- calculating modulo ten (58 mod 10 = 8),
- subtracting from ten (10 − 8 = 2). // only in not zero
- **The check digit is thus 2.**

Principles of EC – PDF417

Error correction: 8 levels , Reed Solomon codes

- based on a polynomial equation where **x power is 2^{s+1}** (s = error correction level). Example s=1 leads to polynomial: $a + bx + cx^2 + dx^3 + x^4$ where a, b, c, d are factors of the polynomial equation (pre computed)
- The equation is : $(x - 3)(x - 3^2)(x - 3^3).....(x - 3^k)$ (with k = 2s+1) MOD 929 is applied on each factor

Variables:

- $k = 2^{s+1}$ = # correction CWs
- a = factors array
- m = number of data CWs
- d = data CWs array
- c = correction CWs array

```
For i = 0 To m - 1
  t = (d(i) + c(k - 1)) Mod 929
  For j = k - 1 To 0 Step -1
    If j = 0 Then
      c(j) = (929 - (t * a(j)) Mod 929) Mod 929
    Else
      c(j) = (c(j - 1) + 929 - (t * a(j)) Mod 929) Mod 929
    End If
  Next
Next
For j = 0 To k - 1
  If c(j) <> 0 Then c(j) = 929 - c(j)
Next
```

Principles of EC – QR code – Reed Solomon – 1

QR codes use Reed-Solomon error correction.

- Step 1: Find out how **many error correction code words** you need to generate.
  - *Version 1 with error correction level Q. This combination requires 13 blocks of data, and 13 error correction code words*
- Step 2: Create your message polynomial
  - Our 13 data blocks :00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 11101100 00010001 11101100
  - Convert each 8-bit word from binary to decimal:
    - *32, 91, 11, 120, 209, 114, 220, 77, 67, 64, 236, 17, 236U*
  - Message polynomial:
    - *32x^25 + 91x^24 + 11x^23 + 120x^22 + 209x^21 + 114x^20 + 220x^19 + 77x^18 + 67x^17 + 64x^16 + 236x^15 + 17x^14 + 236x^13*
  - The exponent of the first term is:
    - *(number of data blocks) + (number of error correction code words) - 1*
    - *In our case, this is 13 + 13 - 1 = 25. So, the first term of our polynomial is* $32x^{25}$.

Principles of EC – QR code – Reed Solomon – 2

- Step 3: Create your generator polynomial.
  - QR codes use a Galois field that has 256 elements, the numbers that we will be dealing with will always be at most 255 and at least 0.
  - The generator polynomial is always of the form $(x - \alpha)(x - \alpha^2) \dots (x - \alpha^t)$, where t is equal to the number of required error correction code words minus 1. *We need 13 error correction code words, so t in this case is 12.*
  - More info at: http://www.thonky.com/qr-code-tutorial/part-2-error-correction/
  - Result of these mathematical operation : 168x^12 + 72x^11 + 22x^10 + 82x^9 + 217x^8 + 54x^7 + 156x^6 + 0x^5 + 46x^4 + 15x^3 + 180x^2 + 122x^1 + 16x^0 ... **Next we convert it to alpha notation**
  - Computed correction coefficients code words: 32 91 11 120 209 114 220 77 67 64 236 17 236 168 72 22 82 217 54 156 0 46 15 180 122 16
  - Convert to binary.
  - Nice calculator http://www.pclviewer.com/rs2/calculator.html


00011491980436
00011491980436