

# Code 128

From Wikipedia, the free encyclopedia

**Code 128** is a very high-density barcode symbology. It is used for alphanumeric or numeric-only barcodes. It can encode all 128 characters of ASCII and, by use of an extension symbol (FNC4), the Latin-1 characters defined in ISO/IEC 8859-1.

GS1-128 (formerly known as UCC/EAN-128) is a subset of Code 128 and is used extensively worldwide in shipping and packaging industries as a product identification code for the container and pallet levels in the supply chain. The symbology was formerly defined as ISO/IEC 15417:2007.



A Swiss postal barcode encoding "RI 476 394 652 CH" in Code 128-B

## Contents

- 1 Specification
  - 1.1 Subtypes
  - 1.2 Quiet zone
  - 1.3 Start/stop and encoded data
  - 1.4 Check digit calculation
  - 1.5 Using FNC4 to encode high (128–255) characters
  - 1.6 Bar code widths
  - 1.7 Availability
  - 1.8 Barcode length optimization
- 2 References
- 3 External links
  - 3.1 Sample code

## Specification

A Code 128 barcode has six sections:

1. Quiet zone
2. Start symbol
3. Encoded data
4. Check symbol
5. Stop symbol
6. Final bar (often considered part of the stop symbol)
7. Quiet zone

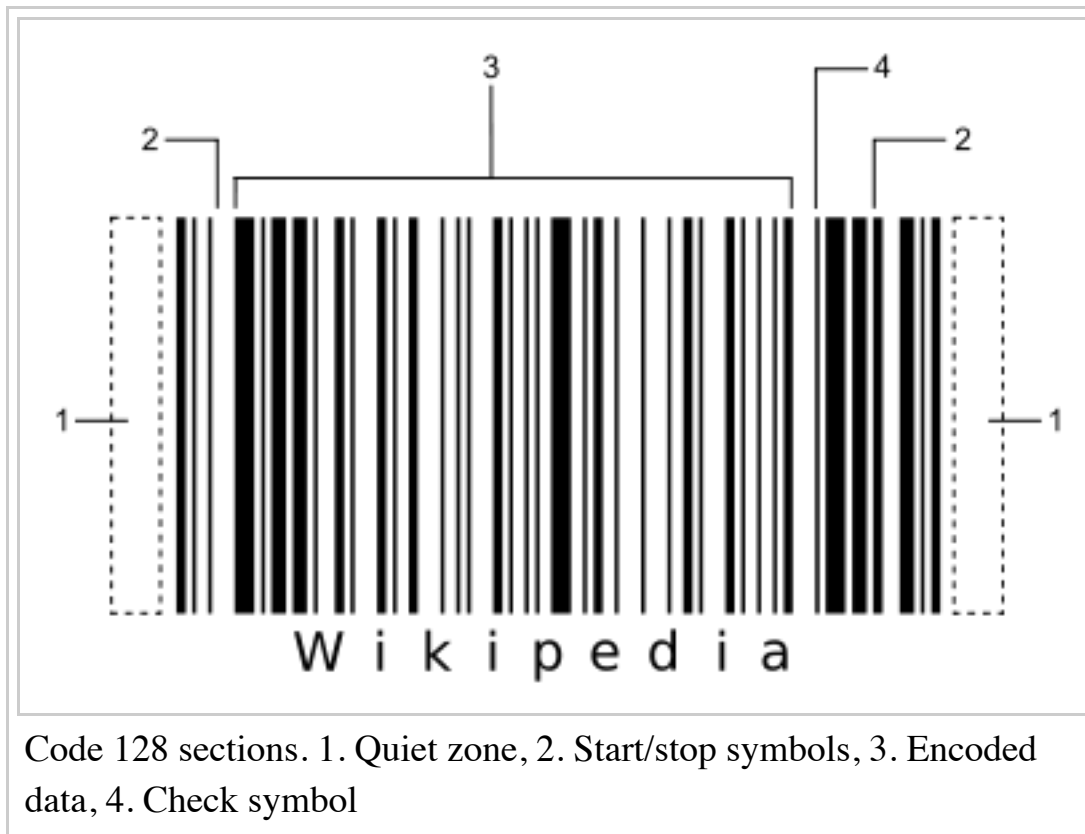
The check symbol is calculated from a weighted sum (modulo 103) of all the symbols.

## Subtypes

Code 128 includes 108 symbols: 103 data symbols, 3 start symbols, and 2 stop symbols. Each symbol consist of three black bars and three white spaces of varying widths. All widths are multiples of a basic "module". Each bar and space is 1 to 4 modules wide, and are fixed width: the sum of the widths of the three

black bars and three white bars is 11 modules.

The stop pattern is composed of two overlapped symbols and has four bars. The stop pattern permits bidirectional scanning. When the stop pattern is read left-to-right (the usual case), the stop symbol (followed by a 2-module bar) is recognized. When the stop pattern is read right-to-left, the reverse stop symbol (followed by a 2-module bar) is recognized. A scanner seeing the reverse stop symbol then knows it must skip the 2-module bar and read the rest of the barcode in reverse.



Despite its name, Code 128 does not have 128 distinct symbols, so it cannot represent 128 code points directly. To represent all 128 ASCII values, it shifts among three code sets (A, B, C). Together, code sets A and B cover all 128 ASCII characters. Code set C is used to efficiently encode digit strings. The initial subset is selected by using the appropriate start symbol. Within each code set, some of the 103 data code points are reserved for shifting to one of the other two code sets. The shifts are done using code points 98 and 99 in code sets A and B, 100 in code sets A and C and 101 in code sets B and C to switch between them):

- 128A (Code Set A) – ASCII characters 00 to 95 (0–9, A–Z and control codes), special characters, and FNC 1–4
- 128B (Code Set B) – ASCII characters 32 to 127 (0–9, A–Z, a–z), special characters, and FNC 1–4
- 128C (Code Set C) – 00–99 (encodes two digits with a single code point) and FNC1

## Quiet zone

The minimum width of the Quiet Zone to the left and right of the 128 Bar Code is  $10x$ , where  $x$  is the minimum width of a module. It is mandatory at the left and right side of the barcode. The quiet zone must not be less than  $4\text{ mm} \times 1.65\text{ mm}$  depending on the type of bar code. Source: Figure 5.2.1.4.4 – 1 of GS1 General Specifications

## Start/stop and encoded data

Each symbol in the barcode is composed of three bars and three spaces. Each bar or space is 1, 2, 3 or 4 units wide, the sum of the widths of bars must be even (4, 6 or 8 units), the sum of the widths of the spaces must be odd (3, 5 or 7 units), and total 11 units per symbol. For instance, encoding the ASCII character "0" can be viewed as 10011101100, where a 1 is a bar and a 0 is a space. A single 1 would be the thinnest line in the bar code. Three 1s in sequence (111) indicates a bar three times as thick as a single 1 bar.

There are 108 possible 11-unit wide symbols, and the code uses all possible symbols. Two of the symbols are used for stop (end-of-barcode) indication, stop and reverse stop. The two stop symbols are merged into a 13-unit long stop pattern. Reading the stop pattern left to right is the stop pattern (followed by a 2-unit bar), and reading the stop pattern right to left is the reverse stop pattern (followed by a 2-unit bar).

## Check digit calculation

The check digit is a weighted modulo-103 checksum. It is calculated by summing the start code 'value' to the products of each symbol's 'value' multiplied by its position in the barcode string. The start symbol *and* first encoded symbol are in position 1. The sum of the products is then reduced modulo 103. The remainder is then converted back to one of the 103 non-delimiter symbols (following the instructions given below) and appended to the barcode, immediately before the stop symbol.

For example, in the following table, the code 128 variant A checksum value is calculated for the alphanumeric string PJJ123C:

| Code              | Value | Position | Value × Position |
|-------------------|-------|----------|------------------|
| Start Code A      | 103   | 1        | 103              |
| P                 | 48    | 1        | 48               |
| J                 | 42    | 2        | 84               |
| J                 | 42    | 3        | 126              |
| 1                 | 17    | 4        | 68               |
| 2                 | 18    | 5        | 90               |
| 3                 | 19    | 6        | 114              |
| C                 | 35    | 7        | 245              |
| Sum               |       |          | 878              |
| Remainder mod 103 |       |          | 54               |

For the purpose of computing the check symbol, the shift and code-switch symbols are treated the same as any other symbol in the bar code. The checksum is computed over the symbol values, without regard for which code set is active at the time. For instance the code set C value "33" and the code set B value "A" are both considered to be a Code 128 value of 33, and the check digit is be computed based on the value of 33 times the symbol's position within the barcode.

## Using FNC4 to encode high (128–255) characters

The special symbol FNC4 ("Function 4"), present only in code sets A and B, can be used to encode all the Latin-1 (ISO-8859-1) characters in a Code 128 barcode.<sup>[1]</sup> The feature is not widely supported and is not used in GS128.<sup>[2][3]</sup> When a single FNC4 is present in a string, the following symbol is converted to ASCII as usual, and then 128 is added to the ASCII value. (If the following symbol is a shift, then a second symbol will be used to obtain the character.) If two FNC4s are used consecutively then all following characters will be treated as such, up to the end of the string or another pair of FNC4s. Between the double FNC4s, a single FNC4 will be used to denote that the following character will be standard ASCII.<sup>[4]</sup>

## Bar code widths

Code128 specifies a combination of 6 bars and spaces for each symbol Thus, each symbol begins with a bar and ends with a space. In barcode fonts, the final bar is generally combined with the stop symbol to make a wider stop pattern. The following table details the widths associated with each bar and space for each symbol. The width of each bar or space may be 1, 2, 3 or 4 units (modules). Using the example above, an 'A' would be depicted with the pattern 10100011000, or as 111323 in the tables below.

|    |       |       |    | Font position<br>(Common/Barcodesoft) |            | Bar/Space   |        |
|----|-------|-------|----|---------------------------------------|------------|-------------|--------|
|    |       |       |    |                                       |            |             |        |
| 0  | space | space | 00 | 32 or 212 / 252                       | ␣ or Ô / ü | 11011001100 | 212222 |
| 1  | !     | !     | 01 | 33                                    | !          | 11001101100 | 222122 |
| 2  | "     | "     | 02 | 34                                    | "          | 11001100110 | 222221 |
| 3  | #     | #     | 03 | 35                                    | #          | 10010011000 | 121223 |
| 4  | \$    | \$    | 04 | 36                                    | \$         | 10010001100 | 121322 |
| 5  | %     | %     | 05 | 37                                    | %          | 10001001100 | 131222 |
| 6  | &     | &     | 06 | 38                                    | &          | 10011001000 | 122213 |
| 7  | '     | '     | 07 | 39                                    | '          | 10011000100 | 122312 |
| 8  | (     | (     | 08 | 40                                    | (          | 10001100100 | 132212 |
| 9  | )     | )     | 09 | 41                                    | )          | 11001001000 | 221213 |
| 10 | *     | *     | 10 | 42                                    | *          | 11001000100 | 221312 |
| 11 | +     | +     | 11 | 43                                    | +          | 11000100100 | 231212 |
| 12 | ,     | ,     | 12 | 44                                    | ,          | 10110011100 | 112232 |
| 13 | -     | -     | 13 | 45                                    | -          | 10011011100 | 122132 |
| 14 | .     | .     | 14 | 46                                    | .          | 10011001110 | 122231 |
| 15 | /     | /     | 15 | 47                                    | /          | 10111001100 | 113222 |
| 16 | 0     | 0     | 16 | 48                                    | 0          | 10011101100 | 123122 |
| 17 | 1     | 1     | 17 | 49                                    | 1          | 10011100110 | 123221 |
| 18 | 2     | 2     | 18 | 50                                    | 2          | 11001110010 | 223211 |
| 19 | 3     | 3     | 19 | 51                                    | 3          | 11001011100 | 221132 |
| 20 | 4     | 4     | 20 | 52                                    | 4          | 11001001110 | 221231 |
| 21 | 5     | 5     | 21 | 53                                    | 5          | 11011100100 | 213212 |
| 22 | 6     | 6     | 22 | 54                                    | 6          | 11001110100 | 223112 |
| 23 | 7     | 7     | 23 | 55                                    | 7          | 11101101110 | 312131 |
| 24 | 8     | 8     | 24 | 56                                    | 8          | 11101001100 | 311222 |
| 25 | 9     | 9     | 25 | 57                                    | 9          | 11100101100 | 321122 |
| 26 | :     | :     | 26 | 58                                    | :          | 11100100110 | 321221 |
| 27 | ;     | ;     | 27 | 59                                    | ;          | 11101100100 | 312212 |
| 28 | <     | <     | 28 | 60                                    | <          | 11100110100 | 322112 |
| 29 | =     | =     | 29 | 61                                    | =          | 11100110010 | 322211 |
| 30 | >     | >     | 30 | 62                                    | >          | 11011011000 | 212123 |
| 31 | ?     | ?     | 31 | 63                                    | ?          | 11011000110 | 212321 |
| 32 | @     | @     | 32 | 64                                    | @          | 11000110110 | 232121 |
| 33 | A     | A     | 33 | 65                                    | A          | 10100011000 | 111323 |
| 34 | B     | B     | 34 | 66                                    | B          | 10001011000 | 131123 |
| 35 | C     | C     | 35 | 67                                    | C          | 10001000110 | 131321 |

|    |     |   |    |     |   |              |        |
|----|-----|---|----|-----|---|--------------|--------|
| 35 | C   | C | 35 | 67  | C | 100010001110 | 131321 |
| 36 | D   | D | 36 | 68  | D | 10110001000  | 112313 |
| 37 | E   | E | 37 | 69  | E | 10001101000  | 132113 |
| 38 | F   | F | 38 | 70  | F | 10001100010  | 132311 |
| 39 | G   | G | 39 | 71  | G | 11010001000  | 211313 |
| 40 | H   | H | 40 | 72  | H | 11000101000  | 231113 |
| 41 | I   | I | 41 | 73  | I | 11000100010  | 231311 |
| 42 | J   | J | 42 | 74  | J | 10110111000  | 112133 |
| 43 | K   | K | 43 | 75  | K | 10110001110  | 112331 |
| 44 | L   | L | 44 | 76  | L | 10001101110  | 132131 |
| 45 | M   | M | 45 | 77  | M | 10111011000  | 113123 |
| 46 | N   | N | 46 | 78  | N | 10111000110  | 113321 |
| 47 | O   | O | 47 | 79  | O | 10001110110  | 133121 |
| 48 | P   | P | 48 | 80  | P | 11101110110  | 313121 |
| 49 | Q   | Q | 49 | 81  | Q | 11010001110  | 211331 |
| 50 | R   | R | 50 | 82  | R | 11000101110  | 231131 |
| 51 | S   | S | 51 | 83  | S | 11011101000  | 213113 |
| 52 | T   | T | 52 | 84  | T | 11011100010  | 213311 |
| 53 | U   | U | 53 | 85  | U | 11011101110  | 213131 |
| 54 | V   | V | 54 | 86  | V | 11101011000  | 311123 |
| 55 | W   | W | 55 | 87  | W | 11101000110  | 311321 |
| 56 | X   | X | 56 | 88  | X | 11100010110  | 331121 |
| 57 | Y   | Y | 57 | 89  | Y | 11101101000  | 312113 |
| 58 | Z   | Z | 58 | 90  | Z | 11101100010  | 312311 |
| 59 | [   | [ | 59 | 91  | [ | 11100011010  | 332111 |
| 60 | \   | \ | 60 | 92  | \ | 11101111010  | 314111 |
| 61 | ]   | ] | 61 | 93  | ] | 11001000010  | 221411 |
| 62 | ^   | ^ | 62 | 94  | ^ | 11110001010  | 431111 |
| 63 | _   | _ | 63 | 95  | _ | 10100110000  | 111224 |
| 64 | NUL | ` | 64 | 96  | ` | 10100001100  | 111422 |
| 65 | SOH | a | 65 | 97  | a | 10010110000  | 121124 |
| 66 | STX | b | 66 | 98  | b | 10010000110  | 121421 |
| 67 | ETX | c | 67 | 99  | c | 10000101100  | 141122 |
| 68 | EOT | d | 68 | 100 | d | 10000100110  | 141221 |
| 69 | ENQ | e | 69 | 101 | e | 10110010000  | 112214 |
| 70 | ACK | f | 70 | 102 | f | 10110000100  | 112412 |
| 71 | BEL | g | 71 | 103 | g | 10011010000  | 122114 |
| 72 | BS  | h | 72 | 104 | h | 10011000010  | 122411 |
| 73 | HT  | i | 73 | 105 | i | 10000110100  | 142112 |

|     |                              |         |        |           |       |               |         |
|-----|------------------------------|---------|--------|-----------|-------|---------------|---------|
| 73  | FF                           | l       | 73     | 105       | l     | 10000110100   | 142112  |
| 74  | LF                           | j       | 74     | 106       | j     | 10000110010   | 142211  |
| 75  | VT                           | k       | 75     | 107       | k     | 11000010010   | 241211  |
| 76  | FF                           | l       | 76     | 108       | l     | 11001010000   | 221114  |
| 77  | CR                           | m       | 77     | 109       | m     | 11110111010   | 413111  |
| 78  | SO                           | n       | 78     | 110       | n     | 11000010100   | 241112  |
| 79  | SI                           | o       | 79     | 111       | o     | 10001111010   | 134111  |
| 80  | DLE                          | p       | 80     | 112       | p     | 10100111100   | 111242  |
| 81  | DC1                          | q       | 81     | 113       | q     | 10010111100   | 121142  |
| 82  | DC2                          | r       | 82     | 114       | r     | 10010011110   | 121241  |
| 83  | DC3                          | s       | 83     | 115       | s     | 10111100100   | 114212  |
| 84  | DC4                          | t       | 84     | 116       | t     | 10011110100   | 124112  |
| 85  | NAK                          | u       | 85     | 117       | u     | 10011110010   | 124211  |
| 86  | SYN                          | v       | 86     | 118       | v     | 11110100100   | 411212  |
| 87  | ETB                          | w       | 87     | 119       | w     | 11110010100   | 421112  |
| 88  | CAN                          | x       | 88     | 120       | x     | 11110010010   | 421211  |
| 89  | EM                           | y       | 89     | 121       | y     | 11011011110   | 212141  |
| 90  | SUB                          | z       | 90     | 122       | z     | 11011110110   | 214121  |
| 91  | ESC                          | {       | 91     | 123       | {     | 11110110110   | 412121  |
| 92  | FS                           |         | 92     | 124       |       | 10101111000   | 111143  |
| 93  | GS                           | }       | 93     | 125       | }     | 10100011110   | 111341  |
| 94  | RS                           | ~       | 94     | 126       | ~     | 10001011110   | 131141  |
| 95  | US                           | DEL     | 95     | 200 / 240 | È / ð | 10111101000   | 114113  |
| 96  | FNC 3                        | FNC 3   | 96     | 201 / 241 | É / ñ | 10111100010   | 114311  |
| 97  | FNC 2                        | FNC 2   | 97     | 202 / 242 | Ê / ò | 11110101000   | 411113  |
| 98  | Shift B                      | Shift A | 98     | 203 / 243 | Ë / ó | 11110100010   | 411311  |
| 99  | Code C                       | Code C  | 99     | 204 / 244 | Ì / ô | 10111011110   | 113141  |
| 100 | Code B                       | FNC 4   | Code B | 205 / 245 | Í / õ | 10111101110   | 114131  |
| 101 | FNC 4                        | Code A  | Code A | 206 / 246 | Î / ö | 11101011110   | 311141  |
| 102 | FNC 1                        | FNC 1   | FNC 1  | 207 / 247 | Ï / ÷ | 11110101110   | 411131  |
| 103 | Start Code A                 |         |        | 208 / 248 | Ð / ø | 11010000100   | 211412  |
| 104 | Start Code B                 |         |        | 209 / 249 | Ñ / ù | 11010010000   | 211214  |
| 105 | Start Code C                 |         |        | 210 / 250 | Ò / ú | 11010011100   | 211232  |
| 106 | Stop                         |         |        | —         | —     | 11000111010   | 233111  |
| —   | Reverse Stop                 |         |        | —         | —     | 11010111000   | 211133  |
| —   | Stop pattern (7 bars/spaces) |         |        | 211 / 251 | Ó / û | 1100011101011 | 2331112 |

The "Code A", "Code B" and "Code C" symbols cause all future symbols to be interpreted according to the corresponding subcode. The "Shift" symbol switches a single following symbol's interpretation between subcodes A and B.

The encoded ASCII char depends on the actual used barcode-font. Especially the ASCII char of value 0 and of value 95 and above may be defined differently in the font that is installed.

The FNC $x$  codes are used for special purposes. FNC1 at the beginning of a bar code indicates a GS1-128 bar code which begins with a 2- 3- or 4-digit *application identifier* assigned by the Uniform Code Council, which explains the following digits. For example, application identifier 421 indicates that an ISO 3166-1 numeric country code and ship-to postal code follows. Thus, the U.S. ZIP code for the White House would generally be printed as "(421) 840 20500", but would actually be coded as "[Start C] [FNC1] 42 18 40 20 50 [Code A] 0 [Check symbol 92] [Stop]"

*Check digit calculation* for the above Zip code example:

| Start C | 105       | 1     | 105  |
|---------|-----------|-------|------|
| FNC1    | 102       | 1     | 102  |
| 42      | 42        | 2     | 84   |
| 18      | 18        | 3     | 54   |
| 40      | 40        | 4     | 160  |
| 20      | 20        | 5     | 100  |
| 50      | 50        | 6     | 300  |
| Code A  | 101       | 7     | 707  |
| 0       | 16        | 8     | 128  |
|         |           | Sum = | 1740 |
| 1740    | Mod 103 = | 92    |      |

## Availability

For the end user, Code 128 barcodes may be generated by either an outside application to create an image of the barcode, or by a font-based barcode solution. Either solution requires the use of an application or an application add in to calculate the check digit and create the barcode.

## Barcode length optimization

Although code set C uses one code symbol to represent two digits, it does not always produce a more compact code than code sets A or B. Using code set C saves one symbol per two digits, but costs a mode-shift symbol to enter and exit the set. Thus, it only worth using if there are enough consecutive digits. For example, encoding the string "X00Y" with code set A or B requires 7 code symbols ([Start B] 56 16 16 57 [checksum] [Stop]), while using code set C for the "00" would result in a code 8 symbols long ([Start B] 56 [Code C] 00 [Code B] 57 [checksum] [Stop]).

Using code set C is only advantageous under the following conditions:

| Location of digits  | Number of consecutive digits |
|---|------------------------------|
| beginning of data   | 4+                           |
| end of data   | 4+                           |
| middle of data (surrounded by symbols from code set A or B) | 6+                           |
| entire data   | either 2 or 4+ (but not 3)   |

At the end of a string, delaying the transition to code set C until there are an even number of digits remaining avoids an extra symbol. Consider the string "...01234": a delayed switch produces ... 0 [Code C] 12 34 [checksum] [Stop] but an early switch produces ... [Code C] 01 23 [Code A] 4 [checksum] [Stop].<sup>[5]</sup>

For example, given the string "098x1234567y23", savings on barcode length using code set C are achieved only if it is applied to middle part of the string. For the beginning and ending part of the string, switching to code set C is not effective. As there are an odd number of digits in the middle of the string, the odd one must be use a different code, set, but it makes no difference whether this is the first or last; 16 symbols are required in either case: [Start B] 0 9 8 x 1 [Code C] 23 45 67 [Code B] y 2 3 [checksum] [Stop], or [Start B] 0 9 8 x [Code C] 12 34 56 [Code B] 7 y 2 3 [checksum] [Stop].

Optimizing the length of the resulting barcode is important when barcode readers are used which must detect the entire barcode image at once in order to read it, such as common laser scanners. The longer the barcode is, the greater distance of laser barcode reader from barcode image is needed, making reading difficult or impossible above some threshold lengths/distances.

The optimal encoding can be found using a dynamic programming algorithm.<sup>[6]</sup>

## References

1. Apparently ISO 15417 Annex F
2. "Code 128 Explained" (<http://www.softmatic.com/barcode-code-128.html>). Softmatic GmbH. Retrieved 2017-01-21. "In principle non-ASCII characters like German umlauts (e.g. ÄÖÜ) can be encoded in a Code 128 symbol by using a special character (FNC4). However, this feature is not widely supported. Using a 2D barcode symbology like Aztec or Datamatrix with dedicated support for non-ASCII data might be a better choice."
3. GS1 General Specifications (January 2006 – Version 7.0), section 5.3.1.1 GS1-128 Symbology Characteristics, stating, "Characters with ASCII values 128 to 255 may also be encoded in Code 128 Symbols. Characters with ASCII values 128 to 255 accessed by Function 4 Character (FNC4) are reserved for future use and are not used in GS1-128 Bar Code Symbols."
4. "TBarcode1D\_Code128" ([http://www.han-soft.com/releases/barcode1d/documents/b\\_code128.html](http://www.han-soft.com/releases/barcode1d/documents/b_code128.html)). Han-soft corporation. Retrieved 2017-01-21. "If a single "FNC 4" character is used, indicates the following data character in the symbol is a extended ASCII character. A 'SHIFT' character may follow the 'FNC 4' character if it is necessary to change character subset for the following data character. Subsequent data characters revert to the standard ASCII character set. If two consecutive 'FNC4' characters are used, all following data characters are extended ASCII characters until two further consecutive 'FNC4' characters are encountered or the end of the symbol is reached. If during this sequence of extended encodation a single "FNC4" character is encountered it is used to revert to standard ASCII encodation for the next data character only. 'SHIFT' and character subset characters shall have their normal effect during such a sequence."
5. GS1 General Specifications, Version 13, Issue 1, Jan-2013, Section 5.4.7.7. Use of Start, Code Set, and Shift symbols to Minimize Symbol Length (Informative), pages 268 to 269. This section gives the compression strategy.
6. Skiena, Steven S. (2010). "8.9 War Story: Text Compression for Bar Codes". *The Algorithm Design Manual* (2nd ed.). ISBN 1-849-96720-2. "dynamic programming led to an 8% tighter encoding on average."

































## External links

- GS1-128 Specification (<http://mdn.morovia.com/kb/GS1-128-Specification-10625.html>) – A detailed list of Application Identifiers.
- Barcodesoft (<http://www.barcodesoft.com/code128mapping.htm>) – Font mapping of Barcodesoft, which differs from the common ascii mapping (see <http://ascii-code.com/>).
- The 128 code (<http://grandzebu.net/informatique/codbar-en/code128.htm>) – Learn the Code 128 encoding algorithm with a font-based barcode solution.
- Online barcode generator (<http://barcode.tec-it.com>) – Free online Barcode generator for the various barcode types.

## Sample code

- ZXing (<https://github.com/zxing/zxing>) – Multiplatform open source barcode scanner / generator with versions available in Java (core project) and ports to ActionScript, C++, C#, ObjectiveC and Ruby.
- Python Bar Code 128 (<http://barcode128.blogspot.com>) – This code appears to draw boxes one pixel wide. It appears it was modified from a short line long line bar code which would have drawn lines. The "Black boxes" should be the same size as the "White Boxes".
- GenCode128 (<http://www.codeproject.com/KB/GDI-plus/GenCode128.aspx>) – Free C# source code implementation of Code128. Almost all features are implemented, but is not 100% complete.
- Barcode1DTools Ruby gem (<https://github.com/mdchaney/Barcode1DTools>) – Ruby source code for many 1D barcode symbologies including Code 128.
- Perl barcode generation code (<https://github.com/mdchaney/barcodes>) – Perl source code for many 1D

barcode symbologies including Code 128.

- Barcode::Code128 (<https://metacpan.org/module/Barcode::Code128/>) – Free Perl barcode generation module.
- GOCR (<http://jocr.sourceforge.net/index.html>) – Free OCR with Code 128 recognition.
- Barcode Code 128 (<http://zanstra.com/my/Barcode.html>) – Free JavaScript source code implementation of Code128.
- Barcode4J (<http://barcode4j.sourceforge.net/index.html>) – Free Java API with implementation of Code128 and other standard barcodes.
- JavaScript Code 128 (<http://www.barcoderesource.com/htmlBarcode.shtml>) – Open source JavaScript implementation of Code128 and other linear barcodes.

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Code\\_128&oldid=781330569](https://en.wikipedia.org/w/index.php?title=Code_128&oldid=781330569)"

Categories: Barcodes

- 
- This page was last edited on 20 May 2017, at 15:32.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.