

ISO/IEC JTC 1/SC 31 N

Date: 2004-04-14

ISO/IEC WD 24719.3

ISO/IEC JTC 1/SC 31/WG 1

Secretariat: ANSI

Information technology – Automatic identification and data capture techniques — Bar code symbology specifications — Micro QR Code

*Technologies de l'information - Techniques d'identification et de saisie de données automatiques —
Spécification de symbologie de code à barre — Micro QR Code*

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International Standard
Document subtype:
Document stage: (20) Preparatory
Document language: E

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Page

Foreword.....	v
Introduction	vi
1 Scope.....	1
2 Normative references	1
3 Terms and definitions	2
4 Symbols (and abbreviations).....	3
5 Conventions.....	4
5.1 Module positions	4
5.2 Byte notation.....	4
6 Symbol description	4
6.1 Basic characteristics.....	4
6.2 Summary of additional features	5
6.3 Symbol Structure.....	5
6.3.1 Symbol Versions and Sizes	6
6.3.2 Function Patterns	7
6.3.3 Encoding Region	8
6.3.4 Quiet Zone.....	8
7 Requirements.....	8
7.1 Encode Procedure Overview	8
7.2 Data Analysis	9
7.3 Modes	10
7.3.1 Numeric Mode.....	10
7.3.2 Alphanumeric Mode	10
7.3.3 8-Bit Byte Mode	10
7.3.4 Kanji Mode	10
7.3.5 Mixing Modes.....	10
7.4 Data Encodation	11
7.4.1 Numeric Mode.....	11
7.4.2 Alphanumeric Mode	13
7.4.3 8-bit Byte Mode.....	14
7.4.4 Kanji Mode	15
7.4.5 Mixing Modes.....	17
7.4.6 Terminator.....	17
7.4.7 Bit Stream to Codeword Conversion	17
7.5 Error Detection and Correction.....	18
7.6 Module placement in matrix	19
7.6.1 Symbol character representation.....	19
7.6.2 Function pattern placement.....	20
7.6.3 Symbol character placement.....	20
7.7 Masking	21
7.7.1 Mask patterns	21
7.7.2 Evaluation of masking results.....	23
7.8 Format Information.....	23
8 Symbol printing and marking	24
8.1 Dimensions	24
8.2 Human-readable interpretation	25
8.3 Marking guidelines	25

9	Symbol quality	25
9.1	Symbol quality parameters	25
9.1.1	Unused Error Correction.....	25
9.1.2	Fixed Pattern Damage	25
9.1.3	Format Information.....	25
9.1.4	Scan grade and overall symbol grade	25
9.2	Process control measurements	25
10	Decoding procedure overview.....	26
11	Reference decode algorithm.....	27
12	Autodiscrimination capability.....	29
13	Transmitted data.....	29
13.1	Encoded data	29
13.2	Symbology Identifier	29
Annex A (normative)	Reed-Solomon Error Correction Generator Polynomials	30
Annex B (normative)	Format Information	31
B.1	Error Correction Bit Calculation.....	31
B.2	Error Correction Decoding Steps.....	31
Annex C (normative)	Reed Solomon Error Correction Decoding Steps	34
Annex D (normative)	Symbology Identifier.....	36
Annex E (normative)	Print quality – Fixed Pattern Damage and Format Information grading.....	37
E.1	Fixed Pattern damage	37
E.1.1	Features to be assessed	37
E.1.2	Fixed Pattern Damage grading procedure	38
E.2	Grading of Format Information.....	39
E.3	Scan grade	40
Annex F (informative)	Symbol encoding example	41
Annex G (informative)	Optimisation of bit stream length	43
Annex H (informative)	User guidelines for printing and scanning of Micro QR Code symbols.....	50
H.1	General	50
H.2	User selection of error correction level	50
Annex I (informative)	Process control techniques	52
I.1	Symbol Contrast.....	52
I.2	Assessing Axial Nonuniformity.....	52
I.3	Visual inspection for symbol distortion and defects.....	52
I.4	Assessing print growth.....	53
Annex J (informative)	Autodiscrimination	54

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24719 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

Introduction

Micro QR Code is a matrix symbology, based on QR Code, which is specified in ISO/IEC 18004. Compared with QR Code, Micro QR Code symbols are designed with a reduced number of overhead modules and are specified in a restricted range of sizes. Micro QR Code enables small to moderate amounts of data to be represented in a small symbol, particularly suited to direct marking on parts and components and to applications where the space available for the symbol is severely restricted. The symbols consist of an array of nominally square modules arranged in an overall square pattern, including a unique finder pattern located at one corner of the symbol and intended to assist in easy location of its position. Four sizes of symbol are provided for together with three levels of error correction. Module dimensions are user-specified to enable symbol production by a wide variety of techniques.

Information technology – Automatic identification and data capture techniques — Bar code symbology specifications — Micro QR Code

1 Scope

This specification defines the requirements for the symbology known as Micro QR Code. It specifies the Micro QR Code symbology characteristics including data character encodation, rules for error control encoding, the graphical symbol structure, symbol dimensions and print quality requirements, a reference decoding algorithm, and user-selectable application parameters.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8859-1:1998, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1*

ISO/IEC 8859-2:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 2: Latin alphabet No. 2*

ISO/IEC 8859-3:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 3: Latin alphabet No. 3*

ISO/IEC 8859-4:1998, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 4: Latin alphabet No. 4*

ISO/IEC 8859-5:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 5: Latin/Cyrillic alphabet*

ISO/IEC 8859-6:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 6: Latin/Arabic alphabet*

ISO/IEC 8859-7:2003, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 7: Latin/Greek alphabet*

ISO/IEC 8859-8:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 8: Latin/Hebrew alphabet*

ISO/IEC 8859-9:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 9: Latin alphabet No. 5*

ISO/IEC 8859-10:1998, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 10: Latin alphabet No. 6*

ISO/IEC 8859-11:2001, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 11: Latin/Thai alphabet*

ISO/IEC 8859-13:1998, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 13: Latin alphabet No. 7*

ISO/IEC 8859-14:1998, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 14: Latin alphabet No. 8 (Celtic)*

ISO/IEC 8859-15:1999, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 15: Latin alphabet No. 9*

ISO/IEC 8859-16:2001, *Information technology -- 8-bit single-byte coded graphic character sets -- Part 16: Latin alphabet No. 10*

ISO/IEC 15415, *Information Technology — Automatic identification and data capture techniques — Bar code print quality test specification — Two-dimensional symbols*

ISO/IEC 15424, *Information Technology — Automatic identification and data capture techniques — Data carrier and symbology identifiers*

ISO/IEC 18004, *Information Technology — Automatic identification and data capture techniques — Bar code symbology specification — QR Code*

ISO/IEC 19762, *Information Technology — Automatic identification and data capture techniques — Harmonised vocabulary*

JIS X 0201, *JIS 8-bit Character Set for Information Interchange*

JIS X 0208-1997, *Japanese Graphic Character Set for Information Interchange*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 and the following apply.

3.1

Character Count Indicator

bit sequence which defines the data string length in a mode.

3.2

encoding region

region of the symbol not occupied by function patterns and available for encodation of data and error correction codewords and for Format Information.

3.3

Format Information

encoded pattern containing information on the symbol number and on the masking pattern used, essential to enable the remainder of the encoding region to be decoded.

3.4

function pattern

overhead component of the symbol required for location of the symbol or identification of its characteristics to assist in decoding.

3.5

Mask Pattern Reference

two-bit identifier of the masking pattern applied to the symbol.

3.6**masking**

process of XORing the bit pattern in the encoding region (or the Format Information) with a masking pattern to provide a symbol with more evenly balanced numbers of dark and light modules and reduced occurrence of patterns which would interfere with fast processing of the image.

3.7**mode**

method of representing a defined character set as a bit string.

3.8**Mode Indicator**

three-bit identifier indicating in which mode the next data sequence is encoded.

3.9**pad codeword**

codeword used to fill data capacity of the symbol unused by data codewords containing the message.

3.10**padding bit**

0 bit, not representing data, used to fill empty positions of the codeword in which the Terminator ends, in a data bit string.

3.11**segment**

sequence of data encoded according to the rules of one ECI or encodation mode.

3.12**separator**

function pattern of all light modules, one module wide, separating the Finder Pattern from the rest of the symbol.

3.13**symbol number**

number indicating the symbol version and error correction level applied, used as part of the Format Information

3.14**Terminator**

bit pattern of between three and nine zero bits (depending on version) used to end the bit string representing data.

3.20**Timing Pattern**

alternating sequence of dark and light modules enabling module coordinates in the symbol to be determined.

3.21**Version**

reference to the size of the symbol represented in terms of its position in the sequence of permissible sizes from 11×11 modules (Version M1) to 17×17 (Version M4) modules.

4 Symbols (and abbreviations)

BCH Bose-Chaudhuri-Hocquenghem

RS Reed-Solomon

5 Conventions

5.1 Module positions

For ease of reference, module positions are defined by their row and column coordinates in the symbol, in the form (i, j) where i designates the row (counting from the top downwards) and j the column (counting from left to right) in which the module is located, with counting commencing at 0. Module (0, 0) is therefore located at the upper left corner of the symbol.

5.2 Byte notation

Byte contents are shown as hexadecimal values.

6 Symbol description

6.1 Basic characteristics

Micro QR Code is a matrix symbology with the following characteristics:

a) Encodable character set:

- numeric data (digits **0 - 9**);
- alphanumeric data (digits **0 - 9**; upper case letters **A - Z**; nine other characters: **space, \$ % * + - . / : ;**);
- 8-bit byte data (JIS 8-bit character set (Latin and Kana) in accordance with JISX0201, or other sets as otherwise defined (see 7.3.3));
- Kanji characters (Shift JIS values: 8140_{HEX} to 9FFC_{HEX} and E040_{HEX} to EAA4_{HEX}. These are values shifted from those of JIS X0208. Refer to JIS X0208 Annex 1 Shift Code Representation for detail.)

b) Representation of data:

A dark module is a binary one and a light module is a binary zero. However, see also 6.2.

c) Symbol size (not including quiet zone):

4 sizes: 11 × 11 modules to 17 × 17 modules, increasing in steps of two modules per side

d) Data characters per symbol (for maximum symbol size, version M4 symbol with error correction level L):

- numeric data: 35 characters
- alphanumeric data: 21 characters
- 8-bit byte data: 15 characters
- Kanji data: 9 characters

e) Selectable error correction

The following Reed-Solomon error correction levels, which can restore the given percentages of the codewords of a symbol, are available depending on to the symbol size.

- Version M1 (11 × 11 modules): error detection only

- Version M2 (13×13 modules): L (7%), M (15%)
- Version M3 (15×15 modules): L (7%), M (15%)
- Version M4 (17×17 modules): L (7%), M (15%), Q (25%)

f) Code type:

Matrix

g) Orientation independence:

Yes

Figure 1 illustrates a Version M2 Micro QR Code symbol in normal colour and with reflectance reversal (see 6.2) (the corner marks indicate the extent of the quiet zone).



Figure 1 – Version M2 Micro QR Code symbol – normal (left) and with reflectance reversal (right)

6.2 Summary of additional features

The following additional features are inherent in Micro QR Code:

- a) *Masking*: This enables the ratio of dark to light modules in the symbol to be approximated to 1:1 whilst minimizing the occurrence of arrangements of adjoining modules which would impede efficient decoding.
- b) *Reflectance reversal*: Symbols are intended to be read when marked so that the image is either dark on light or light on dark (see Figure 1). The specifications in this International Standard are based on dark images on a light background, therefore in the case of symbols produced with reflectance reversal references to dark or light modules should be taken as references to light or dark modules respectively.
- c) *Mirror imaging*: The arrangement of modules defined in this International Standard represents the "normal" orientation of the symbol. It is, however, possible to achieve a valid decode of a symbol in which the arrangement of the modules has been laterally transposed. When viewed with the Finder Pattern at the top left corner of the symbol, the effect of mirror imaging is to interchange the row and column positions of the modules.

6.3 Symbol Structure

A Micro QR Code symbol shall be constructed of nominally square modules set out in a regular square array and shall consist of:

- a encoding region containing symbol characters representing Format Information and data and error correction codewords
- function patterns, namely finder pattern, separator, and timing patterns. Function patterns shall not be used for the encodation of data.

The symbol shall be surrounded on all four sides by a quiet zone. Figure 2 illustrates the structure of a Micro QR Code symbol.

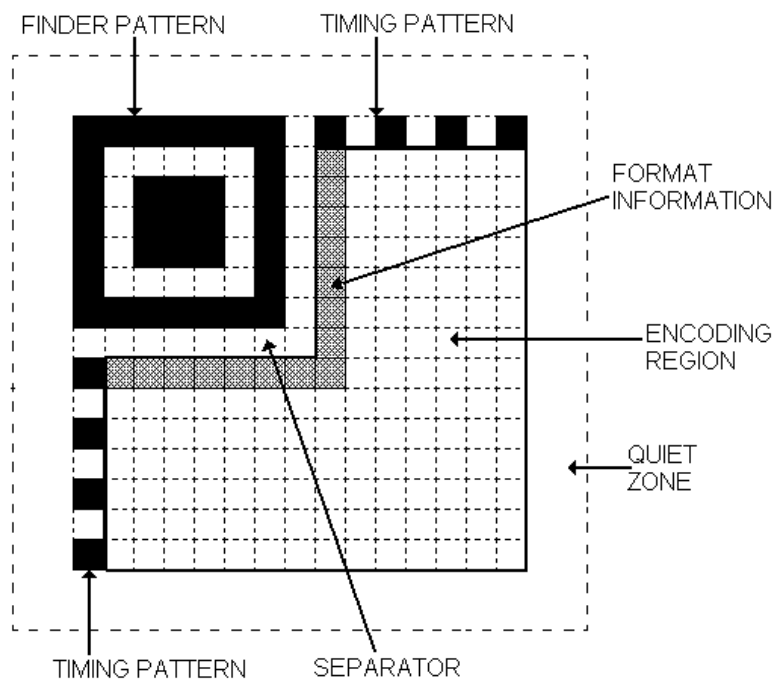


Figure 2 – Structure of MicroQR Code symbol

6.3.1 Symbol Versions and Sizes

There are four sizes of Micro QR Code symbol, referred to as Versions M1 to M4. Version M1 measures 11×11 modules, Version M2 13×13 modules, Version M3 15×15 modules, and Version M4 17×17 modules, i.e. increasing in steps of 2 modules per side. Figure 3 illustrates the structure of Micro QR Code Versions M1 to M4.

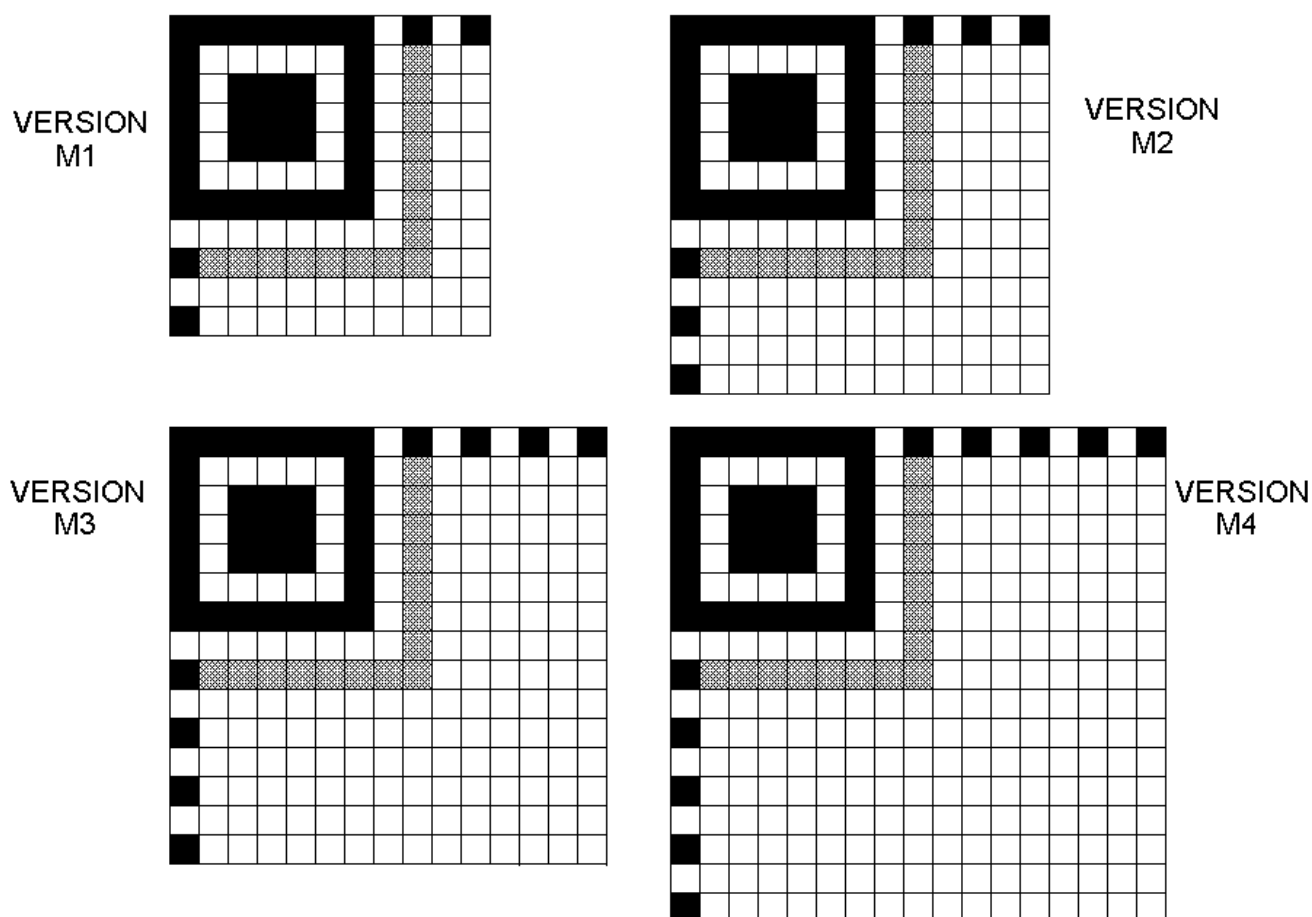


Figure 3 – Versions of MicroQR Code symbol

6.3.2 Function Patterns

6.3.2.1 Finder Pattern

The Finder Pattern may be viewed as three superimposed concentric squares, constructed of dark 7×7 modules, light 5×5 modules and dark 3×3 modules, as illustrated in Figure 4. It is positioned in the upper left corner of the symbol as illustrated in Figure 2. The ratio of module widths in the Finder Pattern is 1:1:3:1:1. The symbol is preferentially encoded so that similar patterns have a low probability of being encountered elsewhere in the symbol, enabling rapid identification of a possible QR Code or Micro QR Code symbol in the field of view. Identification of the Finder Pattern together with the Timing Patterns unambiguously defines the size, location and orientation of the symbol in the field of view.

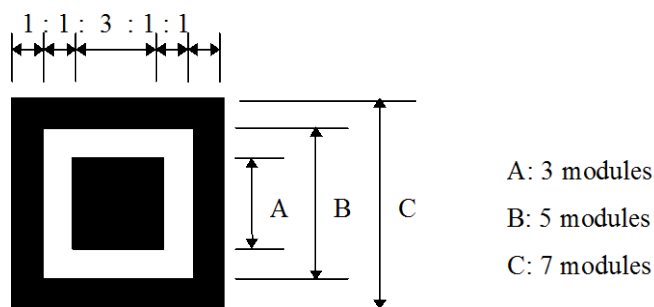


Figure 4 – 1:1:3:1:1 ratio in Finder Pattern

6.3.2.2 Separators

A one-module wide Separator, constructed of all light modules, is placed between the Finder Pattern and the Encoding Region, as illustrated in Figure 2.

6.3.2.3 Timing Pattern

The horizontal and vertical timing patterns respectively consist of a one module wide row or column of alternating dark and light modules, commencing and ending with a dark module. The horizontal timing pattern runs across row 0 of the symbol on the right side of the Separator to the right hand edge of the symbol; the vertical pattern similarly runs down column 0 of the symbol below the Separator to the bottom edge of the symbol. They enable the symbol density and version to be determined and provide datum positions for determining module coordinates.

6.3.3 Encoding Region

This region shall contain the Format Information and the symbol characters representing data and error correction codewords. Refer to 7.6 for details of the symbol characters. Refer to 7.8 for details of the Format Information.

6.3.4 Quiet Zone

This is a region 2 modules wide which shall be free of all other markings, surrounding the symbol on all four sides. Its nominal reflectance value shall be equal to that of the light modules. See Figures 1 and 2 for an indication of its extent.

7 Requirements

7.1 Encode Procedure Overview

This section provides an overview of the steps required to convert input data to a Micro QR Code symbol. An example of the process is shown in Annex E.

Step 1 Data Analysis

Analyze the input data stream to identify the variety of different characters to be encoded. Micro QR Code includes several modes (see 7.3) to allow different sub-sets of characters to be converted into symbol characters in efficient ways. Switch between modes as necessary in order to achieve the most efficient conversion of data into a binary string. Select the required Error Correction Level. If the user has not specified the symbol version to be used, select the smallest version that will accommodate the data, or that will provide the required amount of error correction. A complete list of symbol versions and capacities is shown in Table 7.

Step 2 Data encodation

Convert the data characters into a bit stream in accordance with the rules for the mode in force, as defined in 7.3.1 to 7.3.4, inserting a Mode Indicator as necessary to change mode at the beginning of each new mode segment, and a Terminator at the end of the data sequence. Divide the resulting bit stream into 8 bit codewords and add Pad Characters as necessary to fill the number of data codewords required for the version.

Step 3 Error Correction coding

To enable the error correction algorithms to be processed, generate the error correction codewords, appending them to the end of the data codeword sequence.

Step 4 Module placement

Place the data and error correction codeword modules in the matrix together with the Finder Pattern, Separator, and Timing Pattern.

Step 5 Masking

Apply the masking patterns in turn to the encoding region of the symbol. Evaluate the results and select the pattern which optimizes the dark/light module balance and minimizes the occurrence of undesirable patterns.

Step 6 Format Information

Generate the Format Information and complete the symbol.

Table 1 — Codeword capacity of versions

Version	No. of Modules/side (A)	Function Patterns Modules (B)	Format Information Modules (C)	Encoding Region Modules excl. (C) (D=A ² -B-C)	Encoding Region Capacity [codewords] (E)
M1	11	70	15	36	5
M2	13	74	15	80	10
M3	15	78	15	132	17
M4	17	82	15	192	24

NOTE For Version M1 and M3, the last data codeword is 4 bits in length

7.2 Data Analysis

Analyze the input data string to determine its content and select the appropriate mode to encode each sequence as described in 7.4. Each mode in sequence from Numeric mode to Kanji mode progressively requires more bits per character. It is possible to switch from mode to mode within a symbol in order to minimize the bit stream length for data, some parts of which can more efficiently be encoded in one mode than the other parts, e.g. numeric sequences followed by alphanumeric sequences. It is in theory most efficient to encode data in the mode requiring the fewest bits per data character, but as there is some overhead in the form of Mode Indicator and Character Count Indicator associated with each mode change, it may not always result in the shortest bit stream to change modes for a small number of characters. Also, the capacity of symbols increases in discrete steps from one version to the next, so it may not always be necessary to achieve the maximum conversion efficiency in every case. Annex G gives guidance on efficient encoding of data to maximise the data capacity of each symbol version.

7.3 Modes

7.3.1 Numeric Mode

Numeric mode encodes data from the decimal digit set (0 - 9) (ASCII values 30_{HEX} to 39_{HEX}) at a normal density of 10 bits per 3 data characters. Numeric mode is the only mode available in version M1 symbols.

7.3.2 Alphanumeric Mode

Alphanumeric mode encodes data from a set of 45 characters, i.e.

- 10 numeric digits (0 - 9) (ASCII values 30_{HEX} to 39_{HEX})
- 26 alphanumeric characters (A - Z) (ASCII values 41_{HEX} to 5A_{HEX})
- 9 symbols (Space, \$, %, *, +, -, ., /, :) (ASCII values 20_{HEX}, 24_{HEX}, 25_{HEX}, 2A_{HEX}, 2B_{HEX}, 2D_{HEX} to 2F_{HEX}, 3A_{HEX}).

Normally, two input characters are represented by 11 bits.

Alphanumeric mode is not available in version M1 symbols.

7.3.3 8-Bit Byte Mode

The default interpretation of the 8-bit Byte mode represents data from the 8-bit Latin/Kana character set in accordance with JIS X 0201 (character values 00_{HEX} to FF_{HEX}). In this mode, data is encoded at a density of 8 bits per character.

In closed-system national or application-specific implementations of Micro QR Code, an alternative 8-bit character set, as defined in an appropriate part of ISO 8859, may be specified for 8-bit Byte mode; when an alternative character set is specified, however, the parties intending to read the Micro QR Code symbols require to be notified of the applicable character set in the application specification or by bilateral agreement. When an alternative character set is specified, Kanji mode shall not be used.

<<DRAFTING NOTE: Above paragraph added. Reference character sets to be defined. However if the chosen character set includes values in the range 80 to 9F and E0 to FF (ISO 8859-1, for example, fills the E0 to FF range) it will not be possible to discriminate automatically between 8-bit byte and Kanji data in the output byte stream.>>

8-bit Byte mode is not available in version M1 or M2 symbols.

7.3.4 Kanji Mode

The Kanji mode encodes Kanji double-byte characters in accordance with the Shift system based on JIS X 0208. The Shift JIS values are shifted from the JIS X 0208 values. Refer to JIS X 0208 Annex 1 Shift Coded Representation for detail. Each two-byte character value is compacted to a 13-bit binary codeword.

Kanji mode is not available in version M1 or M2 symbols, nor in symbols for which an alternative 8-bit byte mode character set has been specified (see 7.3.3).

7.3.5 Mixing Modes

Micro QR Code symbols may contain sequences of data in a combination of any of the modes described in 7.3.1 to 7.3.4, although version M1 symbols may only contain data in Numeric mode and version M2 symbols may only contain data in Numeric or Alphanumeric modes. Annex G gives guidance on symbol capacity when data may be encoded in more than one mode.

7.4 Data Encodation

Input data is converted into a bit stream consisting of one or more segments each in one mode. A complete bit stream is made up of segments each containing the following.

- Mode Indicator, which defines the applicable mode for the segment
- Character Count Indicator, which is the binary representation of the number of input data characters in the segment
- Data bit stream, encoded according to the rules for the applicable mode.

Each segment shall begin with the first (most significant) bit of the Mode Indicator and end with the final (least significant) bit of the data bit stream. The length of each segment is defined unambiguously by the rules for the mode in force and the number of input data characters, therefore no explicit separator between segments is required.

To encode a sequence of input data in a given mode, the steps defined in 7.4.1 to 7.4.4 shall be followed. Table 2 defines the length (number of bits) and the bit sequences of the Mode Indicators for each version. Table 3 defines the length (number of bits) of the Character Count Indicators.

Table 2 — Mode Indicators

Version	Mode Indicator Length (bits)	Numeric Mode	Alphanumeric Mode	8-bit Byte Mode	Kanji Mode
M1	0	-	-	-	-
M2	1	0	1	-	-
M3	2	00	01	10	11
M4	3	000	001	010	011

Table 3 — Length of Character Count Indicators

Version	Character Count Indicator length (bits)			
	Numeric Mode	Alphanumeric Mode	8-bit Byte Mode	Kanji Mode
M1	3	-	-	-
M2	4	3	-	-
M3	5	4	4	3
M4	6	5	5	4

7.4.1 Numeric Mode

The input data string is divided into groups of three digits, and each group is converted to its 10-bit binary equivalent. If the number of input digits is not an exact multiple of three, the final one or two digits are

converted to 4 or 7 bits respectively. The binary data is then concatenated and prefixed by the Mode Indicator and the Character Count Indicator. In the Numeric Mode, the Mode Indicator has 0, 1, 2, or 3 bits and the Character Count Indicator has 3, 4, 5, or 6 bits, as defined in Table 3. The Mode Indicator and the number of input data characters are converted to binary numbers and added in the sequence of Mode Indicator, Character Count Indicator followed by the binary data sequence.

EXAMPLE 1 (for version M4-M symbol)

Input data: 01234567

1. Divide into groups of three digits: 012 345 67

2. Convert each group to its binary equivalent: 012 = 0000001100
345 = 0101011001
67 = 1000011

3. Connect the binary data in sequence: 0000001100 0101011001 1000011

4. Convert Character Count Indicator to binary (6 bits for version M4-M):
No. of input data characters: 8 = 001000

5. Add Mode Indicator (000 for version M4-M) and Character Count Indicator to binary data:
000 001000 0000001100 0101011001 1000011

EXAMPLE 2 (for version M4-M symbol)

Input data: 0123456789012345

1. Divide into groups of three digits: 012 345 678 901 234 5

2. Convert each group to its binary equivalent: 012 = 0000001100
345 = 0101011001
678 = 1010100110
901 = 1110000101
234 = 0011101010
5 = 0101

3. Connect the binary data in sequence:
0000001100 0101011001 1010100110 1110000101 0011101010 0101

4. Convert Character Count Indicator to binary (6 bits for version M4-M):
No. of input data characters: 16 = 010000

5. Add Mode Indicator (000 for version M4-M) and Character Count Indicator to binary data:
000 010000 0000001100 0101011001 1010100110 1110000101 0011101010 0101

In Numeric Mode, the length of the bit stream for any number of data characters is given by the following formula:

$$B = M + C + 10(D \text{ DIV } 3) + R$$

where:

B = number of bits in bit stream

M = number of bits in Mode Indicator (from Table 2)

C = number of bits in Character Count Indicator (from Table 3)

D = number of input data characters

R = 0 if (D MOD 3) = 0

R = 4 if (D MOD 3) = 1

R = 7 if (D MOD 3) = 2

7.4.2 Alphanumeric Mode

Each input data character is assigned a character value V from 0 to 44 according to Table 4.

Table 4 — Character values for Alphanumeric mode

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

The input data characters are divided into groups of two characters which are encoded to 11-bit binary codes. The character value of the first character is multiplied by 45 and the character value of the second is added to the product. The sum is then converted to an 11 bit binary number. If the number of the input data characters is not a multiple of two, the character value of the final character is encoded as a 6-bit binary number. In the Alphanumeric Mode, the Mode Indicator has 1, 2 or 3 bits, as defined in Table 2, and the Character Count Indicator has 3, 4 or 5 bits, as defined in Table 3. The Mode Indicator and the number of input data characters are converted to binary numbers and added in the sequence of Mode Indicator, Character Count Indicator followed by the binary data sequence.

EXAMPLE (for version M4-M symbol)

Input data: AC-42

1. Determine character values according to Table 5: AC-42 (10,12,41,4,2)
2. Divide the result into groups of two decimal values: (10,12) (41,4) (2)

3. Convert each group to its 11-bit binary equivalent: (10,12) $10 * 45 + 12 = 462$ 00111001110

(41,4) $41 * 45 + 4 = 1849$ 11100111001

Convert single final digit to its 6-bit binary equivalent (2) 2 000010

4. Connect the binary data in sequence: 00111001110 11100111001 000010

5. Convert Character Count Indicator to binary (5-bit for version M4-M):

No. of input data characters: 5 = 00101

6. Add Mode Indicator (001 for version M4-M) and Character Count Indicator to binary data:

001 00101 00111001110 11100111001 000010

In Alphanumeric Mode, the length of the bit stream for any number of data characters is given by the following formula:

$$B = M + C + 11(D \text{ DIV } 2) + 6(D \text{ MOD } 2)$$

where:

B = number of bits in bit stream

M = number of bits in Mode Indicator (from Table 3)

C = number of bits in Character Count Indicator (from Table 3)

D = number of input data characters

7.4.3 8-bit Byte Mode

In this mode, one 8-bit codeword directly represents the JIS8 character value of the input data character as shown in Table 5, i.e. a density of 8 bits per character.

Table 5 — Character values for 8-bit Byte mode

Char. HEX	Char. HEX	Char. HEX	Char. HEX	Char. HEX	Char. HEX	Char. HEX	Char. HEX
NUL 00	SP 20	@ 40	` 60	80	A0	タ C0	E0
SOH 01	! 21	A 41	a 61	81	。 A1	チ C1	E1
STX 02	" 22	B 42	b 62	82	「 A2	ツ C2	E2
ETX 03	# 23	C 43	c 63	83	」 A3	テ C3	E3
EOT 04	\$ 24	D 44	d 64	84	、 A4	ト C4	E4
ENQ 05	% 25	E 45	e 65	85	・ A5	ナ C5	E5
ACK 06	& 26	F 46	f 66	86	ヲ A6	ニ C6	E6
BEL 07	' 27	G 47	g 67	87	ヅ A7	ヌ C7	E7
BS 08	(28	H 48	h 68	88	イ A8	ネ C8	E8
HT 09) 29	I 49	i 69	89	ウ A9	ノ C9	E9
LF 0A	* 2A	J 4A	j 6A	8A	エ AA	ハ CA	EA
VT 0B	+ 2B	K 4B	k 6B	8B	オ AB	ヒ CB	EB
FF 0C	, 2C	L 4C	l 6C	8C	ヤ AC	フ CC	EC
CR 0D	- 2D	M 4D	m 6D	8D	ユ AD	ヘ CD	ED

Char.	HEX	Char.	HEX	Char.	HEX	Char.	HEX	Char.	HEX	Char.	HEX	Char.	HEX
SO	0E	.	2E	N	4E	n	6E		8E	ヨ	AE	ホ	CE
SI	0F	/	2F	O	4F	o	6F		8F	ッ	AF	マ	CF
DLE	10	0	30	P	50	p	70		90	ー	B0	ミ	D0
DC1	11	1	31	Q	51	q	71		91	ア	B1	ム	D1
DC2	12	2	32	R	52	r	72		92	イ	B2	メ	D2
DC3	13	3	33	S	53	s	73		93	ウ	B3	モ	D3
DC4	14	4	34	T	54	t	74		94	エ	B4	ヤ	D4
NAK	15	5	35	U	55	u	75		95	オ	B5	ユ	D5
SYN	16	6	36	V	56	v	76		96	カ	B6	ヨ	D6
ETB	17	7	37	W	57	w	77		97	キ	B7	ラ	D7
CAN	18	8	38	X	58	x	78		98	ク	B8	リ	D8
EM	19	9	39	Y	59	y	79		99	ケ	B9	ル	D9
SUB	1A	:	3A	Z	5A	z	7A		9A	コ	BA	レ	DA
ESC	1B	;	3B	[5B	{	7B		9B	サ	BB	ロ	DB
FS	1C	<	3C	¥	5C		7C		9C	シ	BC	ワ	DC
GS	1D	=	3D]	5D	}	7D		9D	ス	BD	ン	DD
RS	1E	>	3E	^	5E	-	7E		9E	セ	BE	・	DE
US	1F	?	3F	_	5F	DEL	7F		9F	ソ	BF	・	DF
													FF

NOTE In the JIS8 character set, byte values 80 to 9F and E0 to FF are not assigned but are reserved. Some of these values are used as the first byte in the Shift JIS character set and can therefore be used to distinguish between the JIS8 and Shift JIS character sets. Refer to JIS X 0208 Annex 1 Shift Coded Representation for details.

The binary data is then concatenated and prefixed by the Mode Indicator and the Character Count Indicator. In the 8-bit Byte mode, the Mode Indicator has 2 or 3 bits, as defined in Table 2, and the Character Count Indicator has 4 or 5 bits, as defined in Table 3. The Mode Indicator and the number of input data characters are converted to binary numbers and added in the sequence of Mode Indicator, Character Count Indicator followed by the binary data sequence.

In the 8-bit Byte Mode, the length of the bit stream for any number of data characters is given by the following formula:

$$B = M + C + 8D$$

where:

B = number of bits in bit stream

M = number of bits in Mode Indicator (from Table 2)

C = number of bits in Character Count Indicator (from Table 3)

D = number of input data characters

7.4.4 Kanji Mode

In the Shift JIS system, Kanji characters are represented by a two-byte combination. These byte values are shifted from the JIS X 0280 values. Refer to JIS X 0208 Annex 1 Shift Coded Representation for details. The input data characters in Kanji Mode are compacted to 13-bit binary codes as described below. The binary data is then concatenated and prefixed by the Mode Indicator and the Character Count Indicator. In the Alphanumeric Mode, the Mode Indicator has 2 or 3 bits and the Character Count Indicator has 4 or 5 bits, as defined in Table 3. The Mode Indicator and the number of input data characters are converted to the binary numbers and added in the sequence of Mode Indicator, Character Count Indicator followed by the binary data

sequence. To distinguish Kanji data unambiguously from data in 8-bit Byte mode the most significant byte of each pair is from a group of byte values that are not used in 8-bit Byte mode.

a) For characters with Shift JIS values from 8140_{HEX} to 9FFC_{HEX}:

- 1) Subtract 8140_{HEX} from Shift JIS value;
- 2) Multiply most significant byte of result by C0_{HEX};
- 3) Add least significant byte to product from 2);
- 4) Convert result to a 13-bit binary string.

b) For characters with Shift JIS values from E040_{HEX} to EAA4_{HEX}:

- 1) Subtract C140_{HEX} from Shift JIS value;
- 2) Multiply most significant byte of result by C0_{HEX};
- 3) Add least significant byte to product from 2);
- 4) Convert result to a 13-bit binary string.

EXAMPLES

Input character	“点”	“茗”
(Shift JIS value):	935F	E4AA
1. Subtract 8140 or C140	935F - 8140 = 121F	E4AA - C140 = 236A
2. Multiply m.s.b. by C0	12 × C0 = D80	23 × C0 = 1A40
3. Add l.s.b.	D80 + 1F = D9F	1A40 + 6A = 1AAA
4. Convert to 13-bit binary	0D9F = 0 1101 1001 1111	1AAA = 1 1010 1010 1010

c) For all characters:

Prefix Mode Indicator (2 or 3 bits) and Character Count Indicator binary equivalent (3 or 4 bits) to the binary sequence representing input data characters;

In the Kanji Mode, the length of the bit stream for any number of data characters is given by the following formula:

$$B = M + C + 13D$$

where:

B = number of bits in bit stream

M = number of bits in Mode Indicator (from Table 2)

C = number of bits in Character Count Indicator (from Table 3)

D = number of input data characters

7.4.5 Mixing Modes

There is the option for a symbol containing a sequence of data in one mode to change modes if the data content requires it, or in order to increase the density of encodation. Each segment of data is encoded in the appropriate mode as indicated in 7.4.1 to 7.4.4, with the basic structure of each segment consisting of Mode Indicator/Character Count Indicator/Data, and the next segment commencing with the subsequent Mode Indicator. See Annex G for guidance on efficient use of the capacity of the different symbol versions when modes may be combined.

Mode Indicator 1	Character Count Indicator	Data	Mode Indicator 2	Character Count Indicator	Data	Mode Indicator 3	Character Count Indicator	Data	...
---------------------	---------------------------	------	---------------------	---------------------------	------	---------------------	---------------------------	------	-----

7.4.6 Terminator

The Terminator defined in Table 6 is added to the end of data in the symbol for each version. The Terminator may be omitted if the data bit stream completely fills the capacity of the symbol or contracted if sufficient data bits are not left.

Table 6 — Terminator length and bit sequences

Version	No. of Terminator Bits	Bit Sequence of Terminator
M1	3	000
M2	5	00000
M3	7	0000000
M4	9	000000000

7.4.7 Bit Stream to Codeword Conversion

The bit streams corresponding to each mode segment shall be connected in order. The Terminator shall be appended to complete the bit stream, unless the data bit stream completely fills the capacity of the symbol. The resulting message bit stream shall then be divided into codewords. All codewords are 8 bits in length, except for the final data codeword in versions M1 and M3, which is 4 bits in length. If the bit stream length is such that it does not end at a codeword boundary, padding bits with binary value 0 shall be added after the final bit (least significant bit) of the data stream to extend it to the codeword boundary. The message bit stream shall then be extended to fill the data capacity of the symbol corresponding to the Version and Error Correction Level, as defined in Table 7, by adding the Pad Codewords 11101100 and 00010001 alternately. For versions M1 and M3, the final data codeword is 4 bits long. The Pad Codeword used in this position shall be represented as 0000.

Table 7 — Data capacity

Version	Error Correction Level	Number of Data Codewords*	Number of Data Bits**	Data Capacity			
				Numeric	Alphanumeric	8-bit Byte	Kanji
M1	Error Detection	3	20	5	-	-	-
M2	L M	5	40	10	6	-	-
		4	32	8	5	-	-
M3	L M	11	84	23	14	9	6
		9	68	18	11	7	4
M4	L	16	128	35	21	15	9
	M	14	112	30	18	13	8
	Q	10	80	21	13	9	5

NOTE 1 All codewords are 8 bits in length, except that the final data codeword for Versions M1 and M3 is 4 bits long.

NOTE 2 The number of data bits includes bits for Mode Indicator and Character Count Indicator.

NOTE 3 Annex G indicates data capacity for symbols containing data in a combination of modes.

7.5 Error Detection and Correction

Micro QR Code employs Reed-Solomon error control coding to detect and correct errors. RS codewords for Reed-Solomon coding are appended to the end of the data codeword sequence. Annex A defines the generator polynomials to be used for the generation of the RS codewords. The final data codeword in version M1 and M3 symbols shall have its length extended from 4 to 8 bits by suffixing 0000 to it for the purpose of calculating the RS codewords.

The polynomial arithmetic for Micro QR Code shall be calculated using bit-wise modulo 2 arithmetic and byte-wise modulo 100011101 arithmetic. This is a Galois field of 2^8 with 100011101 representing the field's prime modulus polynomial $x^8 + x^4 + x^3 + x^2 + 1$.

The data codewords are the coefficients of the terms of a polynomial with the coefficient of the highest term being the first data codeword and that of the lowest power term being the last data codeword before the first error correction codeword.

The error correction codewords are the remainder after dividing the data codewords by a polynomial $g(x)$ used for error correction codes (see Annex A). The highest order coefficient of the remainder is the first error correction codeword and the zero power coefficient is the last error correction codeword and the last codeword in the symbol.

The total number of codewords and the number of RS codewords for each version and Error Correction Level are defined in Table 8.

Table 8 — Reed-Solomon characteristics

Version	Total No. of Codewords	Error Correction Level	No. of RS Codewords	RS Code Blocks (c, k, r) ^a
M1	5	Error detection only	2	(5,3,0) ^b
M2	10	L	5	(10,5,1) ^b
		M	6	(10,4,2) ^b
M3	17	L	6	(17,11,2) ^b
		M	8	(17,9,4)
M4	24	L	8	(24,16,3) ^b
		M	10	(24,14,5)
		Q	14	(24,10,7)

^a (c, k, r) : c = total number of codewords (See Table 1 (E)), k = number of data codewords, r = number of error correction capacity

^b Error correction capacity is less than half the number of error correction codewords, to reduce the probability of misdecodes.

7.6 Module placement in matrix

7.6.1 Symbol character representation

Most codewords shall be represented as a regular 2 x 4 module block in the symbol. There are two ways of positioning these blocks, in a vertical arrangement (2 modules wide and 4 modules high) and, if necessary when placement changes direction, in a horizontal arrangement (4 modules wide and 2 modules high). In Version M1 and M3 symbols, the final symbol character for data shall be a 2 x 2 module block. Figure 5 illustrates both the position and the sequence of placement of the symbol characters in all four versions. In each symbol character, the position of the most significant bit is marked.

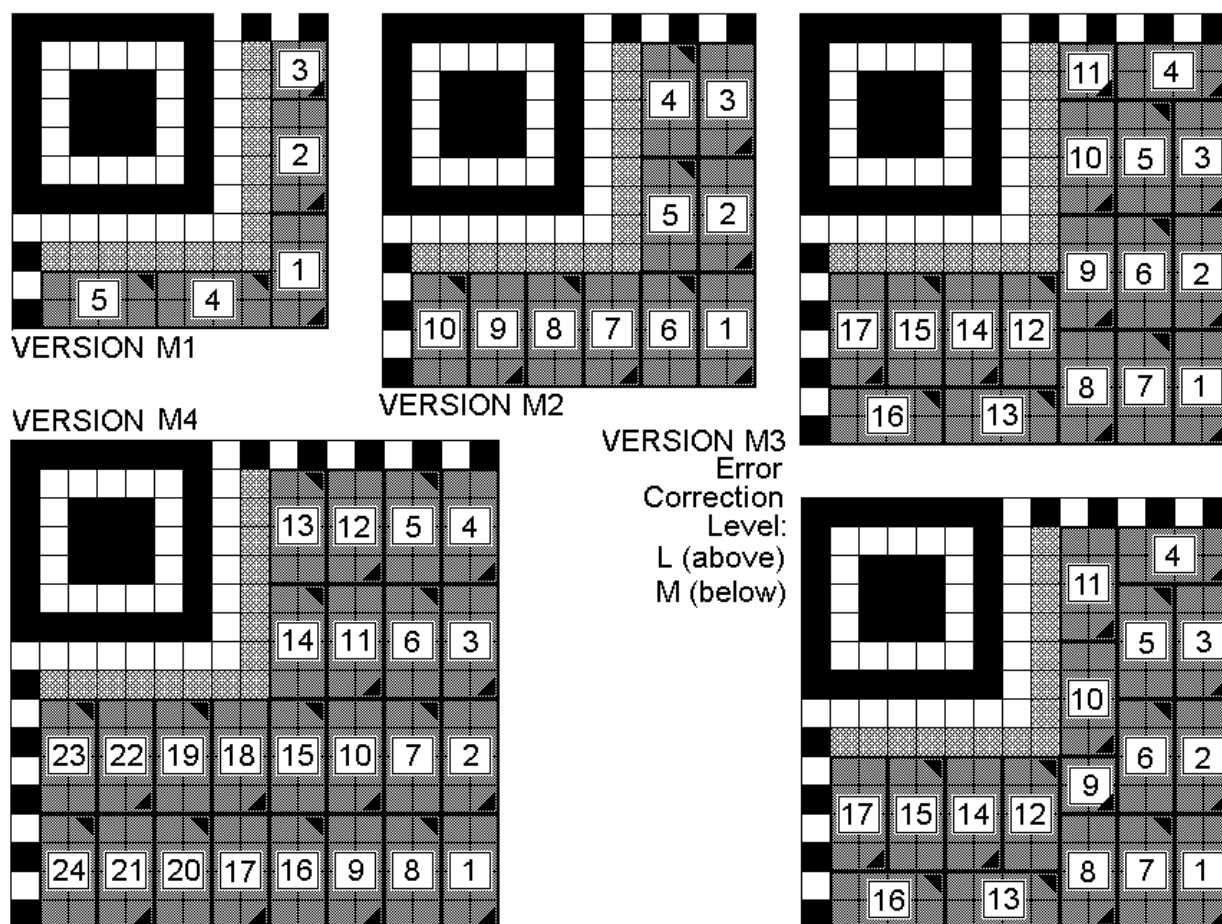


Figure 5 — Position and sequence of placement of symbol characters

7.6.2 Function pattern placement

A square blank matrix shall be constructed with the number of modules horizontally and vertically corresponding to the Version in use. Positions corresponding to the Finder Pattern, Separator, and Timing Pattern shall be filled with either dark modules or light modules as appropriate. These positions are shown in Figure 2 and are common to all Versions.

7.6.3 Symbol character placement

In the encoding region of the Micro QR Code symbol, module positions for the Format Information (see Figure 2) shall be left temporarily blank. The symbol characters are positioned in two-module wide columns commencing at the lower right corner of the symbol and running alternately upwards and downwards from the right to the left, as shown in Figure 5. The principles governing the placement of characters and of bits within the characters are given below. Figures 6 and 7 illustrate the placement of bits in the symbol characters.

- Symbol characters representing data shall be placed first in the symbol, with the first codeword occupying position 1, and those representing Reed-Solomon error correction codewords shall follow the last character representing data.
- The sequence of bit placement in the column shall be from right to left and either upwards or downwards in accordance with the direction of symbol character placement.
- The most significant bit (shown as bit 7) of each codeword shall be placed in the first available module position. Subsequent bits shall be placed in the next module positions. The most significant bit therefore occupies the lower right module of a regular symbol character when the direction of placement is upwards,

and the upper right module when the direction of placement is downwards. Figure 6 shows the position of the most significant bit in each symbol character.



Figure 6 — Bit placement in symbol character in upwards and downwards directions

d) When the boundary of the area available for symbol characters is reached (i.e. the lower edge of the Timing Pattern or Format Information, or the lower edge of the symbol) any remaining bits in the codeword shall be placed in the next column to the left. The direction of placement reverses.

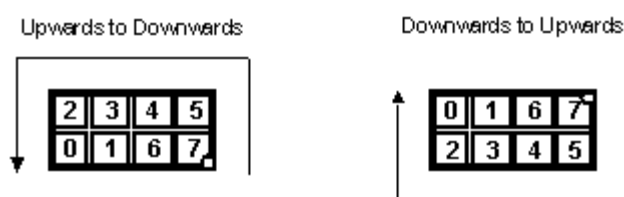


Figure 7 — Bit placement in symbol characters when direction of placement changes

7.7 Masking

For reliable Micro QR Code reading, it is preferable for dark and light modules to be arranged in a well-balanced manner in the symbol. The characteristic bit pattern 1011101 found in the Finder Pattern should be avoided in other areas of the symbol as much as possible. To meet the above conditions, masking should be applied following the steps described below:

1. Masking is not applied to function patterns, or to the Format Information, which has its own masking provision.
2. Convert the given module pattern in the encoding region (excluding the Format Information) with multiple matrix patterns successively through the XOR operation. For the XOR operation, lay the module pattern over each of the masking matrix patterns in turn and reverse the modules (from light to dark or vice versa) which correspond to dark modules of the masking pattern.
3. Then evaluate all the resulting converted patterns by charging penalties for undesirable features on each conversion result.
4. Select the pattern with the lowest penalty points score.

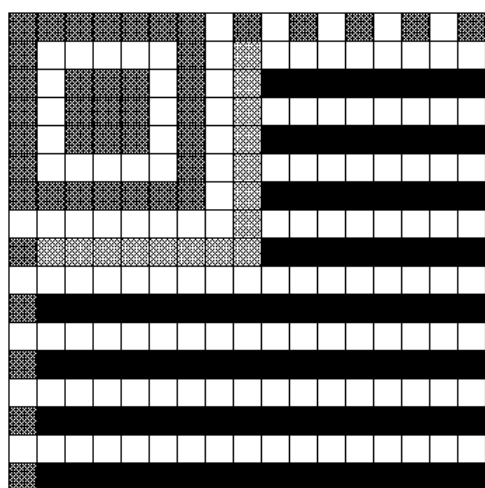
7.7.1 Mask patterns

Table 9 shows the Mask Pattern References (binary reference used in the Format Information) and the mask pattern generation conditions. The mask pattern is generated by defining as dark any module in the encoding region (excluding the area reserved for Format Information), for which the condition is true. In this condition, i refers to the row position of the module and j to its column position, with $(i, j) = (0, 0)$ for the top left module in the symbol.

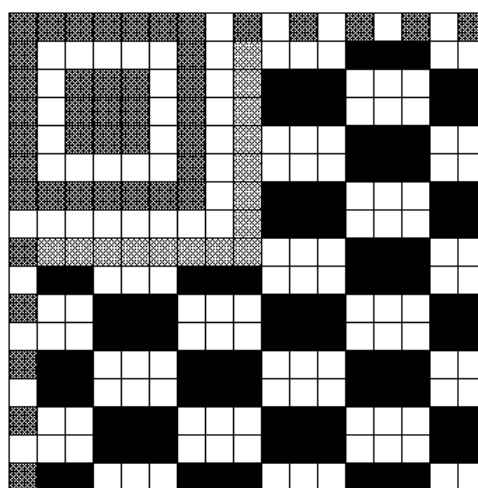
Table 9 — Mask pattern references and conditions

Mask Pattern Reference	Condition
00	$i \bmod 2 = 0$
01	$((i \text{ div } 2) + (j \text{ div } 3)) \bmod 2 = 0$
10	$((ij) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$
11	$((i+j) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$

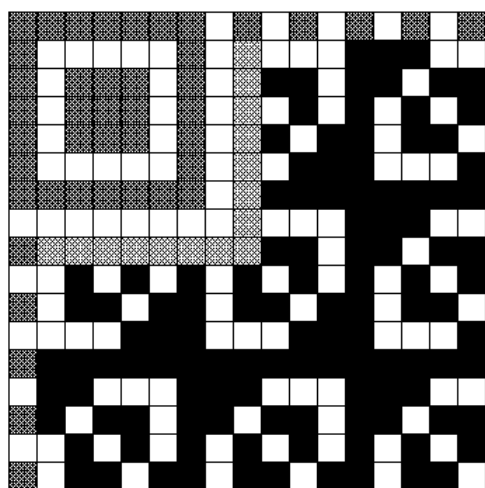
Figure 8 shows the mask patterns.



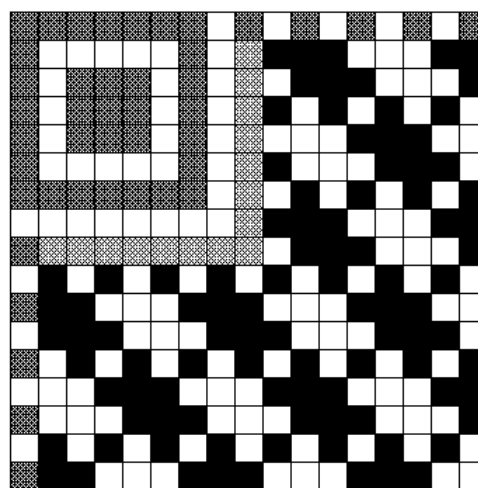
Mask Pattern 00



Mask Pattern 01



Mask Pattern 10



Mask Pattern 11

Figure 8 — Mask patterns

7.7.2 Evaluation of masking results

After performing the masking operation with each Mask Pattern in turn, the results shall be evaluated by scoring penalty points for each occurrence of the following features. The higher the number of points, the less acceptable the result. In Table 10 below, the variables N_1 to N_4 represent weighted penalty scores for the undesirable features ($N_1=3$, $N_2=3$, $N_3=40$, $N_4=10$), i is the amount by which the number of adjacent modules of the same color exceeds 5 and k is the rating of the deviation of the proportion of dark modules in the symbol from 50% in steps of 5%. Although the masking operation is only performed on the encoding region of the symbol excluding the Format Information, the area to be evaluated is the complete symbol.

Table 10 — Scoring of masking results

Feature	Evaluation condition	Points
Adjacent modules in row/column in same color	No. of modules = $(5 + i)$	$N_1 + i$
Block of modules in same color	Block size = $m \times n$	$N_2 \times (m - 1) \times (n - 1)$
1:1:3:1:1 ratio (dark:light:dark:light:dark)	Pattern present in row/column	N_3
Proportion of dark modules in entire symbol	$50 \pm (5 \times k)\%$ to $50 \pm (5 \times (k + 1))\%$	$N_4 \times k$

7.8 Format Information

The Format Information is a 15 bit sequence containing 5 data bits, with 10 error correction bits calculated using the (15, 5) BCH code. For details of the error correction calculation for the Format Information, refer to Annex B. The first three data bits contain the Symbol Number (in binary), which identifies the version and error correction level, as shown in Table 11:

Table 11 — Symbol numbers

Symbol Number	Version	Error Correction Level	Binary Indicator
0	M1	Error detection only	000
1	M2	L	001
2	M2	M	010
3	M3	L	011
4	M3	M	100
5	M4	L	101
6	M4	M	110
7	M4	Q	111

The fourth and fifth data bits of the Format Information contain the Mask Pattern Reference shown in Table 9 for the pattern selected according to 7.7.

The 10 error correction bits shall be calculated as described in Annex B and appended to the 5 data bits.

The 15-bit error corrected Format Information shall then be XORed with the bit pattern 100010001000101, in order to ensure that no combination of Symbol Number and Mask Pattern will result in an all-zero data string.

The resulting masked Format Information shall be mapped into the areas reserved for it in the symbol as shown in Figure 9. The least significant bit of the Format Information is located in the module numbered 0, and the most significant bit in the module numbered 14 in Figure 9.

EXAMPLE

Symbol Number 0:	000
Mask Pattern Reference:	11
Data bits(Symbol Number, Mask Pattern Reference):	00011
BCH bits:	1101011001
Unmasked bit sequence:	000111101011001
Mask pattern for XOR operation:	100010001000101
Format Information module pattern:	100101100011100

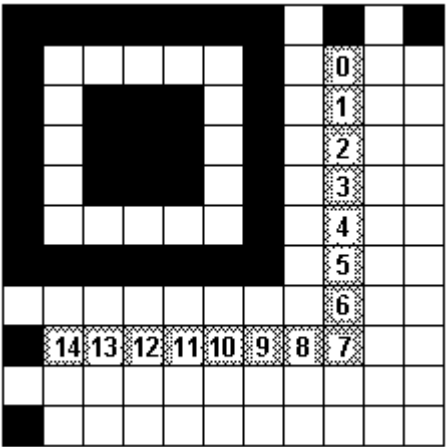


Figure 9 – Format Information bit positions

8 Symbol printing and marking

8.1 Dimensions

Micro QR Code symbols shall conform to the following dimensions:

X dimension: the width of a module shall be specified by the application, taking into account the scanning technology to be used, and the technology to produce the symbol;

Y dimension: the height of a module shall be equal to the X dimension;

minimum quiet zone: equal to 2X on all four sides.

8.2 Human-readable interpretation

Because Micro QR Code symbols are capable of encoding a relatively large number of characters, or may be marked directly on a surface with little additional free space, a human readable interpretation of the data characters may not be practical. As an alternative, descriptive text rather than literal text may accompany the symbol.

The character size and font are not specified, and the message may be printed anywhere in the area surrounding the symbol. The human readable text should not interfere with the symbol itself nor the quiet zones.

8.3 Marking guidelines

Micro QR Code symbols can be printed or marked using a number of different techniques. Annex I provides user guidelines.

9 Symbol quality

Micro QR Code symbols shall be assessed for quality using the 2D matrix bar code symbol print quality guidelines defined in ISO/IEC 15415, as augmented and modified below.

9.1 Symbol quality parameters

9.1.1 Unused Error Correction

The Format Information shall be excluded from the calculation and grading of UEC.

9.1.2 Fixed Pattern Damage

Annex E.1 defines the measurement and grading basis for Fixed Pattern Damage.

9.1.3 Format Information

Annex E.2 defines the measurement and grading basis for the Format Information.

9.1.4 Scan grade and overall symbol grade

The scan grade shall be the lowest of the grades for symbol contrast, modulation, fixed pattern damage, format information, decode, axial non-uniformity, grid non-uniformity and unused error correction. The overall symbol grade is the arithmetic mean of the individual scan grades for a number of tested images of the symbol.

9.2 Process control measurements

A variety of tools and methods can be used to perform useful measurements for monitoring and controlling the process of creating Micro QR Code symbols. These include:

- Symbol contrast readings using a linear bar code verifier.
- Horizontal (and vertical) print growth by measurement of the Finder Pattern in both axes using a linear bar code verifier.
- Determination of axial nonuniformity by physical measurement.
- Visual inspection of the Finder and Timing Patterns for grid nonuniformity and defects.

Each of these tools and methods is described in Annex J.

10 Decoding procedure overview

The decoding steps from reading a Micro QR Code symbol to outputting data characters are the reverse of the encoding procedure. Figure 10 shows an outline of the process flow.

1. Locate and obtain an image of the symbol. Recognize dark and light modules as an array of “0” and “1” bits.
2. Read the Format Information. (Release the masking pattern and perform error correction on the Format Information modules as necessary; identify Symbol Number and Mask Pattern Reference). Recognise symbol orientation and correct for mirror imaging if necessary.
3. Release the Masking by XORing the encoding region bit pattern with the Mask Pattern the reference of which has been extracted from the Format Information.
4. Read the symbol characters according to the placement rules for the version and restore the data and error correction codewords of the message.
5. Detect errors using the error correction codewords corresponding to the error correction level defined by the symbol number. If any error is detected, correct it.
6. Divide the data codewords into bit stream segments according to the Mode Indicators and Character Count Indicators.
7. Finally, decode the Data Characters in accordance with the Mode(s) in use and output the result.

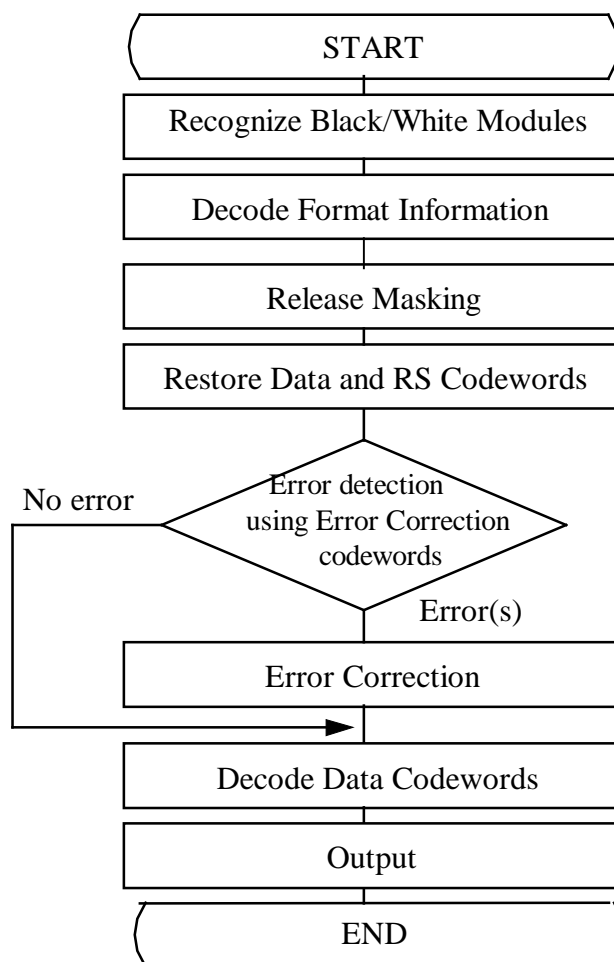


Figure 10 – Decoding flowchart for Micro QR Code

11 Reference decode algorithm

This reference decode algorithm finds the symbol in an image and decodes it. The decode algorithm refers to dark and light states in the image.

1. Determine a Global Threshold by taking a reflectance value midway between the maximum reflectance and minimum reflectance in the image. Convert the image to a set of dark and light pixels using the Global Threshold.
2. Locate the Finder Pattern. As described in 6.3.2.1, module widths in the Finder Pattern are arranged to form a dark-light-dark-light-dark sequence, the relative widths of each element of which are in the ratios 1:1:3:1:1. For the purposes of this algorithm the tolerance for each of these widths is 0,5 (i.e. a range of 0,5 to 1,5 for the single module box and 2,5 to 3,5 for the three module square box).
 - a) When a candidate area is detected note the position of the first and last points A and B respectively at which a line of pixels in the image encounters the outer edges of the Finder Pattern (see Figure 29). Repeat this for adjacent pixel lines in the image until all lines crossing the central box of the Finder Pattern in the x axis of the image have been identified.

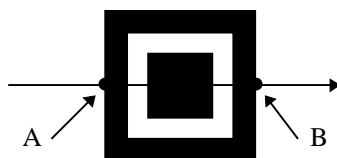


Figure 11 — Scan line in Position Detection Pattern

- b) Repeat step a) for pixel columns crossing the central box of the Finder Pattern in the y axis of the image.
 - c) Locate the outer edges of the pattern. Construct least squares lines through all the points A and all the points B on the pixel lines crossing the central box of the position detection pattern in the x axis. Construct similar lines through all the points A and B on the outermost pixel columns crossing the central box in the y axis. These lines determine the outer edge of the Finder Pattern.
 - d) If no valid Finder Pattern is detected, reverse the colouring of the dark and light pixels and repeat from step 2.
3. Determine the possible angles of rotation of the symbol by analysing the angles of the lines from step 2 c). relative to the imaging sensor axes, as ϑ , $\vartheta + 90^\circ$, $\vartheta + 180^\circ$ and $\vartheta + 270^\circ$.
4. Plot three lines parallel to each axis of the Finder Pattern and equally spaced across the pattern and measure the distances from point A to point B on each line
5. Calculate the provisional module dimension X of the symbol in each axis as one seventh of the mean of the three distances A to B from step 4.
6. Taking each side of the outer box of the Finder Pattern in turn, extend a line outward from the Finder Pattern in both directions, parallel to the edge and 0.5 X in from the edge.
7.
 - a) Identify two edges of the Finder Pattern nominally perpendicular to each other, each of which has both (a) a clear area of at least 1.5X in one direction and (b) alternating light and dark areas evenly spaced at 1X centres from the edge of the Finder Pattern in the opposite direction (a candidate Timing Pattern);
 - b) check that there is the same number of dark modules in each candidate Timing Pattern and that this number is between two and five.
8. Determine the provisional version of the symbol from the number of dark elements in the Timing Pattern:
 - If there are two dark elements, the symbol version is M1;
 - If there are three dark elements, the symbol version is M2;
 - If there are four dark elements, the symbol version is M3
 - If there are five dark elements, the symbol version is M4.
9. From the centre of the first dark module in each side of the Timing Patterns extend a line parallel with the adjacent side of the Finder Pattern to intersect with the corresponding line from the other side and sample an area of 3 x 3 image pixels at 1X intervals along the line to determine the light or dark status of each module of the Format Information. Determine the Format Information bit string by taking the dark pixels as binary 1 and light pixels as binary 0.
10. Release masking of the Format Information by XORing the bit string with the pattern given in 7.8 and decode the Format Information (applying the error correction procedure given in Annex B if necessary) to yield the Symbol Number (and hence the version and error correction level of the symbol) and the Mask Pattern applied to the symbol.

11. If the Format Information bit string is not a valid sequence, determine whether it is a valid sequence if read in the reverse direction and if so continue decoding with the image laterally transposed.
12. Confirm the module pitch X in each axis by dividing the overall width from the outer edge of the Finder Pattern adjacent to the quiet zone to the outer edge of the last dark module in the Timing Pattern by the number of modules corresponding to the symbol version
13. Establish a sampling grid, corresponding to the version of the symbol, of lines spaced $1X$ apart in each axis, parallel to each other and to the side of the Finder Pattern, and running from the centres of the Timing Pattern modules and from similar positions in the Finder Pattern.
14. Sample an area of 3×3 image pixels, centred on each intersection of the grid lines, and determine whether it is dark or light based on the Global Threshold. Construct a bit matrix mapping the dark modules as binary 1 and light modules as binary 0.
15. XOR the Mask Pattern determined in step 10 with the encoding region of the symbol (excluding the Format Information) to release the masking and restore the symbol characters representing data and error correction codewords. This reverses the effect of the masking process applied during the encoding procedure.
16. Determine the symbol codewords in accordance with the placement rules in 7.6.3.
17. Follow the error detection and correction decoding procedure in Annex C to correct errors and erasures up to the maximum correction capacity for the symbol version and Error Correction Level.
18. Subdivide the message data bit stream into segments each commencing with a Mode Indicator and the length of which is determined by the Character Count Indicator following the Mode Indicator.
19. Decode each segment according to the rules for the Mode in force.

12 Autodiscrimination capability

Micro QR Code can be used in an autodiscrimination environment with a number of other symbologies. (See Annex J). In addition, it can be autodiscriminated from QR Code symbols.

13 Transmitted data

13.1 Encoded data

All encoded data characters shall be included in the data transmission. The function patterns, format information, error correction characters, and Pad characters shall not be transmitted. The default transmission mode for all data shall be as their 8-bit JIS8 values or 16-bit Shift JIS values. Because of the character value assignments this gives unambiguous transmission of any sequence of numeric, Latin, Kana and Kanji data.

13.2 Symbology Identifier

ISO/IEC 15424 provides a standard procedure for reporting the symbology which has been read, together with options set in the decoder and any special features encountered in the symbol.

Once the structure of the data has been identified, the appropriate Symbology Identifier should be added by the decoder as a preamble to the transmitted data. See Annex D for the Symbology Identifier and option values which apply to QR Code and to Micro QR Code

Annex A (normative)

Reed-Solomon Error Correction Generator Polynomials

The check character generation polynomial is used to divide the data codeword polynomial, where each codeword is the coefficient of the divided polynomial in descending power order. The coefficients of the remainder of this division are the error correction values.

The polynomial arithmetic for Micro QR Code shall be calculated using bit-wise modulo 2 arithmetic and byte-wise modulo 100011101 arithmetic. This is a Galois field of 2^8 with 100011101 representing the field's prime modulus polynomial $x^8 + x^4 + x^3 + x^2 + 1$. Table A.1 shows the generator polynomials for the RS codes which are used for each Version and Level. In the table, α is the primitive element 2 under $GF(2^8)$. Each generator polynomial is the product of the first degree polynomials: $x - 2^0, x - 2^1, \dots, x - 2^{n-1}$; where n is the degree of the generator polynomial.

Table A.1 – Reed-Solomon generator polynomials

Number of RS Codewords	Generator Polynomials
2	$x^2 + \alpha^{25}x + \alpha$
5	$x^5 + \alpha^{113}x^4 + \alpha^{164}x^3 + \alpha^{166}x^2 + \alpha^{119}x + \alpha^{10}$
6	$x^6 + \alpha^{116}x^5 + \alpha^{134}x^4 + \alpha^{134}x^3 + \alpha^5x^2 + \alpha^{176}x + \alpha^{15}$
8	$x^8 + \alpha^{175}x^7 + \alpha^{238}x^6 + \alpha^{208}x^5 + \alpha^{249}x^4 + \alpha^{215}x^3 + \alpha^{252}x^2 + \alpha^{196}x + \alpha^{28}$
10	$x^{10} + \alpha^{251}x^9 + \alpha^{67}x^8 + \alpha^{46}x^7 + \alpha^{61}x^6 + \alpha^{118}x^5 + \alpha^{70}x^4 + \alpha^{64}x^3 + \alpha^{94}x^2 + \alpha^{32}x + \alpha^{45}$
14	$x^{14} + \alpha^{199}x^{13} + \alpha^{249}x^{12} + \alpha^{155}x^{11} + \alpha^{48}x^{10} + \alpha^{190}x^9 + \alpha^{124}x^8 + \alpha^{218}x^7 + \alpha^{137}x^6 + \alpha^{216}x^5 + \alpha^{87}x^4 + \alpha^{207}x^3 + \alpha^{59}x^2 + \alpha^{22}x + \alpha^{91}$

Annex B (normative)

Format Information

The Format Information consists of a 15-bit sequence comprising 5 data bits and 10 BCH error correction bits. This Annex describes the calculation of the error correction bits and the error correction decoding process.

B.1 Error Correction Bit Calculation

The Bose-Chaudhuri-Hocquenghem $\square 15,5$ code shall be used for error correction. The polynomial whose coefficient is the data bit string shall be divided by the generator polynomial $G(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$. The coefficient string of the remainder polynomial shall be appended to the data bit string to form the $\square 15,5$ BCH code string. Finally, masking is applied as described in 7.8 to ensure that the format information bit pattern is not all zeroes or all ones for any combination of Mask Pattern and Error Correction Level.

Example:

Symbol Number **0**; Mask Pattern **11**

Binary string: **00011**

Polynomial: **$x + 1$**

Raise power to the (15 - 5)th: **$x^{11} + x^{10}$**

Divided by $G(x)$: **$(x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1)(x + 1) + (x^9 + x^8 + x^6 + x^4 + x^3 + 1)$**

Add coefficient string of above remainder polynomial to Format Information data string:

00011 + 1101011001 \rightarrow 000111101011001

XOR with mask **100010001000101**

Result: **100101100011100**

Place these bits in the Format Information areas as described in 7.7.2.

B.2 Error Correction Decoding Steps

Remove the masking of the Format Information modules by XORing the bit sequence with the mask pattern **100010001000101**.

This will yield the following code:

$$R = (r_0, r_1, r_2, \dots, r_{14})$$

That is,

$$R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{14}x^{14}$$

where r_i ($i=0\sim 14$) is 0 or 1.

Calculate the syndrome.

Find the syndrome S_i ($i = 1, 3, 5$).

$$S_1 = R(\alpha) = r_0 + r_1\alpha + r_2\alpha^2 + \dots + r_{14}\alpha^{14}$$

$$S_3 = R(\alpha^3) = r_0 + r_1\alpha^3 + r_2\alpha^6 + \dots + r_{14}\alpha^{42}$$

$$S_5 = R(\alpha^5) = r_0 + r_1\alpha^5 + r_2\alpha^{10} + \dots + r_{14}\alpha^{70}$$

where α is a primitive element of $GF(2^4)$

Find the error position.

$$S_1 + \sigma_1 = 0$$

$$S_3 + S_2\sigma_1 + S_1\sigma_2 + S_3 = 0$$

$$S_5 + S_4\sigma_1 + S_3\sigma_2 + S_2\sigma_3 = 0$$

where: $S_2 = (S_1)^2$, $S_4 = (S_2)^2$

Find the variable σ_i ($i = 1 \sim 3$) for each error position using the above formulae. Then substitute the variable for the following polynomial and substitute elements of $GF(2^4)$ one by one.

$$\sigma(x) = x^3 + \sigma_1x^2 + \sigma_2x + \sigma_3$$

Now, it is found that an error is on the j th digit (counting from the 0-th digit) for the element α_j , which makes $\sigma(\alpha_j) = 0$

Correct the error by reversing the bit value of each error position.

B.3 Alternative method

Since the number of valid combinations of Format Information data and error correction bits is limited, it is possible to use a lookup table both to create the fifteen-bit sequence and to compare the bit sequence that has been read with each tabulated sequence until a match, or the nearest sequence to a match, is obtained; if no more than two bits differ the sequence in table B.1 shall be substituted for the erroneous Format Information sequence.

Table B.1 — Valid Format Information bit sequences

Sequence before masking		Sequence after masking	
Data bits	Error correction bits	binary	hex
00000	0000000000	100010001000101	0x4445
00001	0100110111	100000101110010	0x4172
00010	1001101110	100111000101011	0x4e2b
00011	1101011001	100101100011100	0x4b1c
00100	0111101011	101010110101110	0x55ae
00101	0011011100	101000010011001	0x5099
00110	1110000101	101111111000000	0x5fc0
00111	1010110010	101101011110111	0x5af7
01000	1111010110	110011110010011	0x6793
01001	1011100001	110001010100100	0x62a4
01010	0110111000	110110111111101	0x6dfd
01011	0010001111	110100011001010	0x68ca
01100	1000111101	111011001111000	0x7678
01101	1100001010	111001101001111	0x734f
01110	0001010011	111110000010110	0x7c16
01111	0101100100	111100100100001	0x7921
10000	1010011011	000011011011110	0x06de
10001	1110101100	000001111101001	0x03e9
10010	0011110101	000110010110000	0x0cb0
10011	0111000010	000100110000111	0x0987
10100	1101110000	001011100110101	0x1735
10101	1001000111	001001000000010	0x1202
10110	0100011110	001110101011011	0x1d5b,
10111	0000101001	001100001101100	0x186c
11000	0101001101	010010100001000	0x2508
11001	0001111010	010000000111111	0x203f
11010	1100100011	010111101100110	0x2f66
11011	1000010100	010101001010001	0x2a51
11100	0010100110	011010011100011	0x34e3
11101	0110010001	011000111010100	0x31d4
11110	1011001000	011111010001101	0x3e8d
11111	1111111111	011101110111010	0x3bba

Annex C (normative)

Reed Solomon Error Correction Decoding Steps

Take the Version M3 symbol with error correction level M as an example. For the symbol, the (17, 9, 4) Reed-Solomon code under $GF(2^8)$ is used for error correction. Provided that the code after releasing Masking from the symbol is:

$$R = (r_0, r_1, r_2, \dots, r_{16})$$

That is,

$$R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{16}x^{16}$$

$r_i (i = 0 - 16)$ is an element of $GF(2^8)$

(i) Calculate the syndrome.

Find the syndrome $S_i (i = 0 - 7)$.

$$S_0 = R(1) = r_0 + r_1 + r_2 + \dots + r_{16}$$

$$S_1 = R(\alpha) = r_0 + r_1\alpha + r_2\alpha^2 + \dots + r_{16}\alpha^{16}$$

...

...

$$S_7 = R(\alpha^7) = r_0 + r_1\alpha^7 + r_2\alpha^{14} + \dots + r_{16}\alpha^{112}$$

where α is a primitive element of $GF(2^8)$

(ii) Find the error position.

$$S_0\sigma_4 - S_1\sigma_3 + S_2\sigma_2 - S_3\sigma_1 + S_4 = 0$$

$$S_1\sigma_4 - S_2\sigma_3 + S_3\sigma_2 - S_4\sigma_1 + S_5 = 0$$

$$S_2\sigma_4 - S_3\sigma_3 + S_4\sigma_2 - S_5\sigma_1 + S_6 = 0$$

$$S_3\sigma_4 - S_4\sigma_3 + S_5\sigma_2 - S_6\sigma_1 + S_7 = 0$$

Find the variable $\sigma_i (i = 1 - 4)$ for each error position using the above formulae.

Then, substitute the variable for the following polynomial and substitute elements of $GF(2^8)$ one by one.

$$\sigma(x) = \sigma_4 + \sigma_3x + \sigma_2x^2 + \sigma_1x^3 + x^4$$

Now, it is found that an error is on the j th digit (counting from the 0-th digit) for the element α_j which makes $\sigma(\alpha) = 0$.

(iii) Find the error size.

Supposing that an error is on the j_1, j_2, j_4 digits in (ii) above, then find the size of the error.

$$Y_1\alpha j^1 + Y_2\alpha j^2 + Y_3\alpha j^3 + Y_4\alpha j^4 = S_0$$

$$Y_1\alpha^2 j^1 + Y_2\alpha^2 j^2 + Y_3\alpha^2 j^3 + Y_4\alpha^2 j^4 = S_1$$

$$Y_1\alpha^3 j^1 + Y_2\alpha^3 j^2 + Y_3\alpha^3 j^3 + Y_4\alpha^3 j^4 = S_2$$

$$Y_1\alpha^4 j^1 + Y_2\alpha^4 j^2 + Y_3\alpha^4 j^3 + Y_4\alpha^4 j^4 = S_3$$

Solve the above equations to find the size of each error $Y_i (i = 1 - 4)$.

(iv) Correct the error.

Correct the error by adding the complement of the error size value to each error position.

Annex D (normative)

Symbology Identifier

Micro QR Code and QR Code share the Symbology Identifier assigned to QR Code in ISO/IEC 15424, which should be added as a preamble to the decoded data by a suitably programmed decoder and is in the form:

]Qm

where:] is the Symbology Identifier flag (ASCII value 93)

Q is the code character for the QR Code (and Micro QR Code) symbology

m is the modifier character with one of the values defined in Table D.1. In the case of Micro QR Code, m shall always have the value 1.

The symbology identifier for Micro QR Code is therefore]Q1.

Table D.1 shows, for information, all the valid values of m for QR Code and their significance.

Table D.1 — QR Code Symbology Identifier options and modifier values

Modifier value	Option
0	Model 1 symbol
1	Model 2 symbol, ECI protocol not implemented, OR Micro QR Code symbol
2	Model 2 symbol, ECI protocol implemented
3	Model 2 symbol, ECI protocol not implemented, FNC1 implied in first position
4	Model 2 symbol, ECI protocol implemented, FNC1 implied in first position
5	Model 2 symbol, ECI protocol not implemented, FNC1 implied in second position
6	Model 2 symbol, ECI protocol implemented, FNC1 implied in second position

Annex E (normative)

Print quality – Fixed Pattern Damage and Format Information grading

Because of differences in symbology structures and reference decode algorithms, the effect of certain parameters on a symbol's reading performance may vary. ISO/IEC 15415 provides for symbology specifications to define the grading of certain symbology-specific attributes. This annex therefore defines the method of grading Fixed Pattern Damage and the Format Information for Micro QR Code to be used in the application of ISO/IEC 15415 to it.

E.1 Fixed Pattern damage

E.1.1 Features to be assessed

The features to be assessed are:

- The corner segment, including:
 - the Finder Pattern
 - the 1X wide separators adjoining the two inner sides of the Finder Pattern
 - part of the Quiet Zone of a minimum of two modules width (or more if specified by the application) extending for a length of 11 modules along the two outer sides of the Finder Pattern.
- The two Timing Patterns of alternating dark and light modules running along the top and left side of the symbol from the Finder Pattern

The features listed above shall be assessed as three segments, viz.:

- the corner segment (Finder Pattern with its associated separators and part of the quiet zone) (Segment A), which occupies 104 modules;
- the two Timing Patterns (Segments B1 and B2 respectively),

In a version M4 symbol (17 x 17 modules) for example, each Segment B is 9 modules long.

These segments, in the case of a Version M4 symbol, are illustrated in Figure A.5 below. A indicates the corner segment; and B1 and B2 indicate the two timing pattern segments.

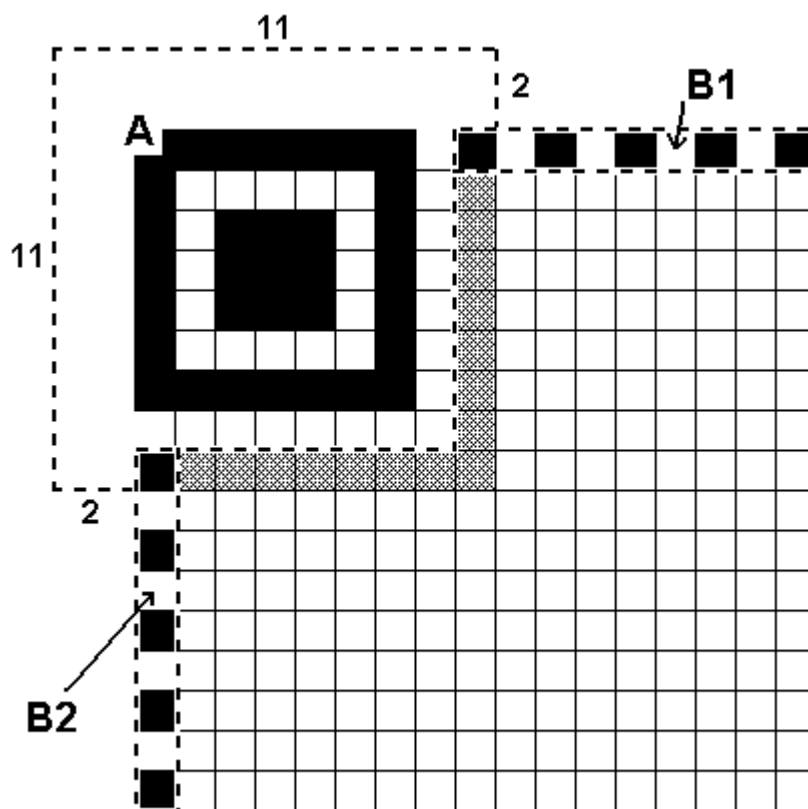


Figure E.1 — Micro QR Code fixed pattern segments

E.1.2 Fixed Pattern Damage grading procedure

Damage to each segment shall be graded based on the modulation of the individual modules that compose it.

The procedure described below shall be applied to each segment in turn

- 1) From the reference grey-scale image of the symbol, find the modulation grade for each module based on the values in Table 6 of ISO/IEC 15415. Since the intended light or dark nature of the module is known, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall be given modulation grade 0.
- 2) For each modulation grade level, assume that all modules not achieving that grade or a higher grade are module errors, and derive a notional damage grade based on the grade thresholds shown in Table A.6. Take the lower of the modulation grade level and the notional damage grade. The notional damage grade is determined as follows:
 - a) For Segment A, count the number of module errors.
 - b) For segments B1 and B2, count the number of module errors. Express this number as a percentage of the total number of modules in the segment.
 - c) For segments B1 and B2, taking groups of five adjacent modules and progressing along the segment in steps of one module, verify that in any group of five adjacent modules no more than two are damaged; if this test fails, the grade for the segment shall be 0.

- d) Assign a notional damage grade to each segment based on the grade thresholds shown in Table A.6.
- 3) The Fixed Pattern Damage grade for the segment shall be the highest resulting grade for all modulation grade levels

The Fixed Pattern Damage grade for the symbol shall be the lowest of the six segment grades.

Table E.1 — Grade thresholds for Micro QR Code Fixed Pattern Damage

Segment A	Segments B1 and B2	
Number of module errors	Percentage of total modules with module errors	Grade
0	0%	4
1	$\leq 7\%$	3
2	$\leq 11\%$	2
3	$\leq 14\%$	1
≥ 4	$> 14\%$	0

<<COMMENT REQUESTED FROM AIM TSC: Since Timing Patterns have 3, 5, 7 or 9 modules only these percentages (similar to those specified in 15415 for QR Code) are not helpful. In a M1 symbol even 1 module error (33%) will destroy the TP, in a M2 symbol 1 module error is 20%, in a M3 it is 14.3% and in a M4 it is 11.1%. Suggest we either (a) consider any module error in the TP as giving a F or (b) set a 1 error A/F threshold for M1 M2 and M3 and a 2 error threshold for M4; in either case the second test (no more than two errors in five consecutive modules) becomes irrelevant.>>

E.2 Grading of Format Information

Micro QR Code symbols contain a set of modules representing information that defines the format of the symbol. This data requires to be reliably detected at an early stage of the decoding procedure, and if it cannot be decoded, the remainder of the symbol cannot be decoded. For this reason the Format Information module block is graded separately (in a similar way to Fixed Pattern Damage), and its grade is included in the overall symbol grade determination.

E.2.1.1 Grading of Format Information

Determine a grade for the Format Information block according to the following method.

- 1) From the reference grey-scale image of the symbol, find the modulation grade for each module based on the values in Table 6 of ISO/IEC 15415. Since the intended light or dark nature of the module is known after decode, any module intended to be dark but the reflectance of which is above the global threshold, and any module intended to be light but the reflectance of which is below the global threshold shall be given modulation grade 0. If the Format Information cannot be decoded, the grade for the Format Information shall be 0.
- 2) For each modulation grade level:
 - a) Assume that all modules not achieving that modulation grade or a higher grade are module errors, and derive a notional grade based on the following:

Table E.2 — Format Information notional grading

Number of module errors	Grade
0	4
1	3
2	2
3	1
≥4	0

- b) Select the lower of the MOD grade and the notional grade at each level as the grade for that level, as illustrated in Table A.8.
- c) The grade for the Format Information shall be the highest resulting grade, as illustrated in Table A.8

Table E.3 —Example of Grading of Format Information block

Modulation grade	Notional grade	Lower of grades
4	2	2
3	2	2
2	3	2
1	3	1
0	4	0
	Selected (highest) Grade->	2

E.3 Scan grade

The scan grade shall be the lowest of the grades for the standard parameters evaluated according to ISO/IEC 15415 together with the grades for Fixed Pattern Damage and Format Information evaluated in accordance with this Annex.

Annex F (informative)

Symbol encoding example

This Annex describes the encoding of the data string **01234567** into a Version M2 symbol with EC level L, using the Numeric Mode in accordance with 7.4.1.

Step 1: Data Encodation

- Divide into groups of three digits and convert each group to its 10 or 7 bit binary equivalent:

012 → 0000001100

345 → 0101011001

67 → 1000011

- Mode Indicator for Numeric Mode in Version M2 is **0**
- Character count is 8; convert to binary (4 bits for Version M2-L):

Character Count Indicator (8) = **1000**

- Terminator is 5 zero bits for Version M2, **00000**
- Connect Mode Indicator for Numeric Mode (**0**), Character Count Indicator (**1000**), binary data, and Terminator (**00000**)

0 1000 0000001100 0101011001 1000011 00000

- Divide into 8-bit codewords, adding 3 padding bits (shown underlined for illustration) since final codeword contained only 5 bits

01000000 00011000 10101100 11000011 00000000

- No Pad codewords are required to fill data codeword capacity of symbol (for version M2-L, 5 data codewords)

Step 2: Error Correction Codeword generation

Using the Reed-Solomon algorithm to generate the required number of error correction codewords (for a Version M2-L symbol, 5 are needed), these (shown underlined for illustration) should be added to the bit stream, resulting in:

01000000 00011000 10101100 11000011 00000000 10000110 00001101 00100010 10101110 00110000

Step 3: Module placement in matrix.

The Finder Pattern and Timing Patterns are placed in a blank 13×13 matrix and the module positions for the Format Information are left temporarily blank. The codewords from Step 2 are placed in the matrix in accordance with 7.6.3.



Figure F.1 — Data modules placed in symbol prior to masking

Step 4: Masking Pattern selection

Apply the masking patterns defined in 7.7.1 in turn and evaluate the results in accordance with 7.7.2. The Masking Pattern selected is referenced **01**. Apply the selected masking pattern to the encoding region of the matrix, as described in 7.7.

Step 5: Format Information

The Symbol Number for an M2-L symbol is 1, which is represented in binary form as **001**, and the masking pattern is **01**. Therefore, the data bits of the Format Information are **001 01**.

The BCH error correction calculation gives **0011011100** as the bit sequence to be added to the data, giving:

001010011011100 as the unmasked Format Information.

XOR this bit stream with the mask **100010001000101**:

001010011011100 (raw bit stream)

100010001000101 (mask)

101000010011001 (Format Information to be placed in symbol)

Step 6: Final symbol construction

Add Format Information modules in positions reserved in step 3.



Figure F.2 — Final version M2-L symbol encoding 01234567

Annex G (informative)

Optimisation of bit stream length

As described in this standard, Micro QR Code offers various modes of encodation each of which differs in the number of bits it requires to represent a given data string. Since there is an overlap between the character sets of each mode - for example, numeric data may be encoded in Numeric, Alphanumeric, 8-bit Byte and Kanji modes, and Latin alphanumeric data may be encoded in Alphanumeric, 8-bit Byte and Kanji modes - the symbol generation software needs to choose the most appropriate mode in which to encode data characters which appear in more than one mode.

This choice must be made initially and may also be possible part way through a data stream.

A number of alternative approaches may be adopted to minimize the bit stream length. The algorithm will need not only to consider the immediate sequence of characters but also look ahead to the next sequence of data in view of the overhead required for switching modes.

The following terminology is appropriate:

Modes are referred to in an ascending order, from Numeric mode at the lowest level, through Alphanumeric, 8-bit byte and Kanji modes. "Exclusive subset" refers to the set of character values in a higher mode that are not also found in a lower mode.

Table G.1 lists the byte values forming the exclusive subset in each mode.

Table G.1 — Exclusive subset byte values for Micro QR Code modes

Mode	Byte values (hexadecimal)
Numeric	30 to 39
Alphanumeric	20, 24, 25, 2A, 2B, 2D - 3A, and 41 - 5A
8-bit byte	00 to 1F, 21 to 23, 26 to 29, 2C, 3B to 40, 5B to FF (excluding reserved values 80 to 9F and E0 to FF)
Kanji	All double bytes in ranges of 8140 to 9FFC and E040 to EAA4

The compaction efficiencies given in 7.4.1 to 7.4.4 need to be interpreted carefully. The best scheme for a given set of data may not be the one with the fewest bits per data character. If the highest degree of compaction is required, account has to be taken of the additional bits required to change modes (additional Mode Indicator and Character Count Indicator). It should also be noted that even if the number of codewords is minimized, the codeword stream may need to be expanded to fill a symbol. This fill process is done using pad characters.

In the case of Micro QR Code the restricted number of symbol versions necessitates a different approach from QR Code.

Assuming that the data to be encoded is in the exclusive subsets of not more than two modes, and that all the data in each subset is grouped together (e.g. "123abcdef"), an algorithm to determine the shortest bit stream for Micro QR Code data can be derived from Table G.2. These principles can be extended to cater for more than two modes, although care must be taken that the resulting bit stream will fit one of the available symbols.

Because the lower modes use fewer bits per character than the higher modes, there is a point at which the extra overhead of the additional Mode Indicator and Character Count Indicator for a change of mode is offset by the greater encoding density of the lower mode. Table G.2 shows the minimum number of consecutive characters in a lower mode for which a shorter total bit stream is achieved by changing modes. For fewer characters, encoding all the data in the higher mode will give a shorter bit stream.

Table G.2 — Minimum characters in lower mode for minimising bit stream length by changing modes

Mode combination	M2 symbols	M3 symbols	M4 symbols
Numeric + Alphanumeric	3 numeric	4 numeric	5 numeric
Numeric + 8 bit byte	n/a	2 numeric	3 numeric
Alphanumeric + 8-bit byte	n/a	3 alphanumeric	4 alphanumeric
Numeric + Kanji	n/a	1 numeric	2 numeric
Alphanumeric + Kanji	n/a	1 alphanumeric	2 alphanumeric
8-bit byte + Kanji	n/a	2 8-bit	2 8-bit

Based on the principles of the above table, and the capacities of the various symbol versions, Figures G.1 to G.6 below show, for each combination of modes, the options available for encoding given amounts of data in combinations of modes.

The column and row headings identify the number of characters in each mode. The figures show the symbol versions and error correction levels, omitting the initial M; thus, for example, 4Q refers to a version M4 symbol with error correction level Q. For any given combination of characters and modes, the available symbol versions are those at the appropriate row and column intersection and those shown to the right of or below that intersection.

For example, if the data string was "123456abcdefgh", consisting of six numeric characters and eight from the alphanumeric character set, Figure G.1 shows that the data would fit into a version M3L symbol (total of 77 bits including Mode Indicators and Character Count Indicators), or a version M4M symbol or a version M4L symbol (81 bits for either). The options may be narrowed down either by the space available or the required level of error correction.

Num.	Alphanumeric																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1				2M	2L					3M		4Q	3L				4M			4L
2			2M	2L					3M		4Q	3L				4M			4L	
3		2M	2L					3M		4Q	3L				4M				4L	
4	2M		2L				3M		4Q	3L				4M				4L		
5	2M	2L				3M		4Q	3L					4M				4L		
6		2L				3M	4Q		3L				4M				4L			
7	2L				3M		4Q	3L				4M			4L					
8				3M	4Q		3L				4M			4L						
9				3M		4Q	3L				4M			4L						
10			3M		4Q	3L					4M			4L						
11			3M	4Q	3L					4M			4L							
12		3M		4Q	3L					4M		4L								
13		3M	4Q	3L					4M			4L								
14	3M	4Q		3L				4M			4L									
15		4Q	3L				4M			4L										
16	4Q		3L			4M			4L											
17	4Q	3L			4M			4L												
18		3L			4M			4L												
19	3L			4M			4L													
20				4M			4L													
21				4M			4L													
22			4M			4L														
23			4M			4L														
24		4M			4L															
25		4M		4L																
26	4M			4L																
27			4L																	
28			4L																	
29		4L																		
30		4L																		
31	4L																			

Figure G.1 — Symbol capacities - numeric and alphanumeric data

Num.	8-bit byte													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						3M		3L/4Q				4M		4L
2						3M	4Q	3L			4M		4L	
3					3M	4Q	3L			4M		4L		
4					3M	4Q	3L			4M		4L		
5				3M	4Q	3L			4M		4L			
6				3M	4Q	3L			4M		4L			
7			3M	4Q	3L			4M		4L				
8			3M	4Q	3L			4M		4L				
9			3M	4Q	3L			4M		4L				
10		3M	4Q	3L			4M		4L					
11		3M	4Q	3L			4M		4L					
12	3M	4Q	3L			4M		4L						
13	3M	4Q	3L			4M		4L						
14	3M	4Q	3L			4M		4L						
15	4Q	3L			4M		4L							
16	4Q	3L			4M		4L							
17	3L			4M		4L								
18	3L			4M		4L								
19			4M		4L									
20			4M		4L									
21			4M		4L									
22		4M		4L										
23		4M		4L										
24	4M		4L											
25	4M		4L											
26	4M		4L											
27		4L												
28		4L												
29	4L													
30	4L													

Figure G.2 — Symbol capacities - numeric and 8-bit byte data

	8-bit													
A/num.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						3M		3L/4Q				4M		4L
2					3M		3L/4Q				4M		4L	
3				3M		3L/4Q				4M		4L		
4				3M	4Q	3L			4M		4L			
5			3M	4Q	3L			4M		4L				
6		3M	4Q	3L			4M		4L					
7		3M	4Q	3L			4M		4L					
8	3M	4Q	3L			4M		4L						
9	4Q	3L			4M		4L							
10	4Q	3L			4M		4L							
11	3L			4M		4L								
12			4M		4L									
13			4M		4L									
14		4M		4L										
15	4M		4L											
16	4M		4L											
17		4L												
18	4L													

Figure G.3 — Symbol capacities - alphanumeric and 8-bit byte data

	Kanji							
Num.	1	2	3	4	5	6	7	8
1				3M/4Q	3L		4M	4L
2			3M	4Q	3L	4M		4L
3			3M	3L/4Q		4M	4L	
4			3M/4Q	3L		4M	4L	
5			3M/4Q	3L		4M	4L	
6		3M	4Q	3L	4M		4L	
7		3M	3L/4Q		4M	4L		
8		3M/4Q	3L		4M	4L		
9		3M/4Q	3L		4M	4L		
10	3M	3L/4Q		4M		4L		
11	3M	3L/4Q		4M	4L			
12	3M/4Q	3L		4M	4L			
13	4Q	3L		4M	4L			
14	3L/4Q		4M		4L			
15	3L/4Q		4M	4L				
16	3L		4M	4L				
17	3L		4M	4L				
18		4M		4L				
19		4M	4L					
20		4M	4L					
21		4M	4L					
22	4M	4L						
23	4M	4L						
24	4M	4L						
25		4L						
26	4L							
27	4L							
28	4L							
29	4L							

Figure G.4 — Symbol capacities - numeric and Kanji data

	Kanji							
A/num.	1	2	3	4	5	6	7	8
1			3M	4Q	3L		4M	4L
2			3M	3L/4Q		4M	4L	
3			3M/4Q	3L		4M	4L	
4		3M	3L/4Q		4M		4L	
5		3M/4Q	3L		4M	4L		
6	3M	4Q	3L	4M		4L		
7	3M	3L/4Q		4M	4L			
8	3M/4Q	3L		4M	4L			
9	3L/4Q		4M	4L				
10	3L		4M	4L				
11		4M		4L				
12		4M	4L					
13	4M		4L					
14	4M	4L						
15	4M	4L						
16	4L							
17	4L							

Figure G.5 — Symbol capacities - alphanumeric and Kanji data

	Kanji							
8-bit	1	2	3	4	5	6	7	8
1			3M	4Q	3L		4M	4L
2			3M/4Q	3L		4M	4L	
3		3M	3L/4Q		4M	4L		
4	3M	4Q	3L		4M	4L		
5	3M/4Q	3L		4M	4L			
6	3L/4Q		4M		4L			
7	3L		4M	4L				
8		4M	4L					
9	4M		4L					
10	4M	4L						
11	4L							
12	4L							

Figure G.6 — Symbol capacities - 8-bit byte and Kanji data

Annex H (informative)

User guidelines for printing and scanning of Micro QR Code symbols

H.1 General

Any Micro QR Code application must be viewed as a total system solution. All the symbology encoding/decoding components (surface markers or printers, labels, readers) making up an installation need to operate together as a system. A failure in any link of the chain, or a mismatch between them, could compromise the performance of the overall system.

While compliance with the specifications is one key to assuring overall system success, other considerations come into play which may influence performance as well. The following guidelines suggest some factors to keep in mind when specifying or implementing bar or matrix code systems:

1. Select a print density which will yield tolerance values that can be achieved by the marking or printing technology being used. Ensure that the module dimension is an integer multiple of the print head pixel dimension (both parallel to and perpendicular to the print direction). Ensure also that any adjustment for print gain (or loss) is performed by changing an equal integer number of pixels from dark to light (or light to dark) on all dark-to-light boundaries of individual or groups of adjoining dark modules in order to ensure that the module center spacing remains constant, although the apparent bit-map representation of the individual dark (or light) modules is adjusted in size to suit the direction of compensation.
2. Choose a reader with a resolution suitable for the symbol density and quality produced by the marking or printing technology.
3. Ensure that the optical properties of the printed symbol are compatible with the wavelength of the scanner light source or sensor.
4. Verify symbol compliance in the final label or package configuration. Overlays, show-through and curved or irregular surfaces can all affect symbol readability.

The effects of specular reflection from glossy symbol surfaces must be considered. Scanning systems must take into account the variations in diffuse reflection between dark and light features. At some scanning angles, the specular component of the reflected light can greatly exceed the desired diffuse component, changing the scanning performance. In cases where the surface of the material or part can be altered, matt, non-glossy surfaces may help minimize specular effects. Where this option is not available, particular must be taken to ensure the illumination of the symbol to be read optimizes the desired contrast components.

H.2 User selection of error correction level

The users should define the appropriate level of error correction to suit the application requirements. As shown in Table 12, the three levels from L to Q offer increasing capabilities of detecting and correcting errors, at the cost of some increase in symbol size for a given message length. For example, a Version M4-Q symbol can contain a total of 9 characters of 8-bit byte data, but if a lower level of error correction was acceptable, the same data could also be represented in a Version M3-L symbol.

The error correction level should be determined in relation to:

- the expected level of symbol quality: the lower the expected quality grade, the higher the level to be applied;
- the importance of a high first read rate;

- the opportunity for re-scanning in the event of a read failure;
- the space constraints which might reduce the opportunity to use a higher error correction level.

Error correction level L is appropriate for high symbol quality and/or the need for the smallest possible symbol for given data. Level M is described as “Standard” level and offers a good compromise between small size and increased reliability. Level Q is a “High reliability” level and suitable for more critical or poor print quality applications.

Annex I (informative)

Process control techniques

This Annex describes tools and procedures useful for monitoring and controlling the process of creating scannable Micro QR Code symbols. These techniques do not constitute a print quality check of the produced symbols - the method defined in 9 is the required method for assessing symbol quality - but they individually and collectively yield good indications of whether the symbol production process is creating workable symbols.

I.1 Symbol Contrast

Most verifiers for linear bar code symbols have either a reflectometer mode or a mode for plotting scan reflectance profiles and/or reporting Symbol Contrast, as defined in ISO/IEC 15416, from undecodable scans. Except with symbols requiring special illumination configurations, the symbol contrast readings that can be obtained using a 0,150 mm or 0,250 mm aperture at 660 nm wavelength - either the reported symbol contrast value, the maximum to minimum scan reflectance profile excursions, or the difference between maximum and minimum reflectometer readings - are found to correlate well with an image-derived symbol contrast value. In particular these reading can be used to check that symbol contrast stays well above the minimum allowed for the intended symbol quality grade.

I.2 Assessing Axial Nonuniformity

Measure the distance D from the left edge of the Finder Pattern to the right edge of the last module in the upper Timing Pattern, and the distance D' from the top edge of the Finder Pattern to the bottom edge of the last module in the left-side Timing Pattern. Substitute the results in the formula below and grade the result for an assessment of Axial Nonuniformity.

$$AN = \left| \frac{(D - D')}{(D + D')/2} \right|$$

where $\left| \right|$ yields the absolute value. Axial Nonuniformity is then graded as:

4,0 (A) if $AN \leq 0,06$

3,0 (B) if $AN \leq 0,08$

2,0 (C) if $AN \leq 0,10$

1,0 (D) if $AN \leq 0,12$

0,0 (F) if $AN > 0,12$

I.3 Visual inspection for symbol distortion and defects

Ongoing visual inspection of the Finder and Timing Patterns in sample symbols can monitor an important aspect of the production process.

Matrix code symbols are susceptible to errors caused by local distortions of the matrix grid. Any such distortions may show up visually as either crooked edges on the Finder Patterns or uneven spacings within the alternating Timing Patterns.

The Finder Patterns and the adjacent quiet zone areas should always be solidly dark and light. Failures in the print mechanism which may produce defects in the form of light or dark streaks through the symbol should be visibly evident where they traverse the finder pattern or the quiet zone. Such systematic failures in the print process should be corrected.

I.4 Assessing print growth

A linear bar code verifier capable of outputting direct measurements of bar and space patterns may be used for the assessment of print gain or loss in both horizontal and vertical axes, by measuring along two scan paths at right angles, one passing through the Finder Pattern and crossing the centre 3 x 3 block of modules horizontally, and the other similarly passing through it vertically. Analysis of the output should reveal an apparent bar/space/bar/space/bar pattern; the print gain (or loss) can be assessed by comparing the five measured element widths with the ideal 1:1:3:1:1 ratio of the widths.

Print growth in each axis can be calculated as follows:

Measure five element widths a , b , c , d and e

Calculate X :

$$X = \frac{a + 2b + 2c + 2d + e}{12}$$

Print growth is expressed as:

$$PG_{abs} = \frac{[(a + c + e) - (b + d)] - 3X}{5}.$$

or, as a percentage of X :

$$PG_{percent} = \frac{20[(a + c + e) - (b + d)]}{X} - 60$$

Annex J (informative)

Autodiscrimination

Micro QR Code may be read by suitably programmed decoders which have been designed to autodiscriminate it from other symbologies. A properly programmed Micro QR Code reader will not decode a symbol in another symbology as a valid Micro QR Code symbol; however, representations of short linear symbols may be found in any matrix symbol including Micro QR Code.

The decoder's valid set of symbologies should be limited to those needed by a given application in order to maximize reading security.