



**International
Standard**

ISO/IEC 18004

**Information technology —
Automatic identification and data
capture techniques — QR code bar
code symbology specification**

*Technologies de l'information — Technologie d'identification
automatique et de capture des données — Spécification de la
symbologie de code à barres code QR*

**Fourth edition
2024-08**

Reference number
ISO/IEC 18004:2024(en)

Full document details: ISO IEC 18004-2024

 **Documents**
© ISO/IEC 2024 Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Mathematical and logical symbols, abbreviated terms and conventions	3
4.1 Mathematical and logical symbols	3
4.2 Abbreviated terms	3
4.3 Conventions	4
4.3.1 Module positions	4
4.3.2 Byte notation	4
4.3.3 Version references	4
5 Symbol description	4
5.1 Basic characteristics	4
5.2 Summary of additional features	5
5.3 Symbol structure	6
5.3.1 General	6
5.3.2 Symbol versions and sizes	8
5.3.3 Finder pattern	14
5.3.4 Separator	15
5.3.5 Timing pattern	15
5.3.6 Alignment patterns	15
5.3.7 Encoding region	15
5.3.8 Quiet zone	15
6 Conformance	16
7 Requirements	16
7.1 Encode procedure overview	16
7.1.1 General	16
7.1.2 Step 1: Data analysis	16
7.1.3 Step 2: Data encoding	16
7.1.4 Step 3: Error correction coding	16
7.1.5 Step 4: Structure final message	16
7.1.6 Step 5: Module placement in matrix	17
7.1.7 Step 6: Data masking	17
7.1.8 Step 7: Format and version information	17
7.2 Data analysis	18
7.3 Modes	18
7.3.1 General	18
7.3.2 Extended channel interpretation mode	18
7.3.3 Numeric mode	19
7.3.4 Alphanumeric mode	19
7.3.5 Byte mode	19
7.3.6 Kanji mode	19
7.3.7 Mixing modes	19
7.3.8 Structured append mode	20
7.4 Data encoding	20
7.4.1 FNC1 mode	20
7.4.2 Sequence of data	20
7.4.3 Extended channel interpretation mode	21
7.4.4 Numeric mode	23
7.4.5 Alphanumeric mode	24
7.4.6 Byte mode	25

7.4.7	Kanji mode	26
7.4.8	Mixing modes	27
7.4.9	FNC1 modes	27
7.4.10	Terminator	29
7.4.11	Bit stream to codeword conversion	29
7.5	Error correction	33
7.5.1	Error correction capacity	33
7.5.2	Generating the error correction codewords	40
7.6	Constructing the final message codeword sequence	41
7.7	Codeword placement in matrix	42
7.7.1	Symbol character representation	42
7.7.2	Function pattern placement	42
7.7.3	Symbol character placement	43
7.8	Data masking	46
7.8.1	General	46
7.8.2	Data mask patterns	46
7.8.3	Evaluation of data masking results	49
7.9	Format information	51
7.9.1	QR code symbols	51
7.9.2	Micro QR code symbols	52
7.10	Version information	54
8	Structured append	55
8.1	Basic principles	55
8.2	Symbol sequence indicator	56
8.3	Parity data	56
9	Symbol printing and marking	57
9.1	Dimensions	57
9.2	Human-readable interpretation	57
9.3	Marking guidelines	57
10	Symbol quality	57
10.1	Methodology	57
10.2	Symbol quality parameters	58
10.2.1	Fixed pattern damage	58
10.2.2	Scan grade and overall symbol grade	58
10.2.3	Grid non-uniformity	58
10.2.4	Print growth	58
10.3	Process control measurements	58
11	Decoding procedure overview	58
12	Reference decode algorithm for QR code	59
13	Autodiscrimination capability	67
14	Transmitted data	67
14.1	General principles	67
14.2	Symbology identifier	67
14.3	Extended channel interpretations	67
14.4	FNC1 mode	68
Annex A (normative)	Error detection and correction generator polynomials	69
Annex B (normative)	Error correction decoding steps	72
Annex C (normative)	Format information	74
Annex D (normative)	Version information	76
Annex E (normative)	Position of alignment patterns	78
Annex F (normative)	Symbology identifier	80
Annex G (normative)	QR code print quality — Symbology-specific aspects	81

Annex H (informative) JIS8 and shift JIS character sets	87
Annex I (informative) Symbol encoding examples	89
Annex J (informative) Optimisation of bit stream length	94
Annex K (informative) User guidelines for printing and scanning of QR code symbols	103
Annex L (informative) Autodiscrimination	105
Annex M (informative) Process control techniques	106
Annex N (informative) Characteristics of model 1 symbols	107
Bibliography	110

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

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

This fourth edition cancels and replaces the third edition (ISO/IEC 18004:2015), which has been technically revised.

The main changes are as follows:

- continuous grading according to ISO/IEC 15415 has been adopted for grade fixed pattern damage;
- the reference decoding algorithm has been clarified.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

There are four technically different, but closely related members of the QR code family, which represent an evolutionary sequence.

- QR code model 1 is the original specification for QR code and is described in AIM ITS 97-001^[21].
- QR code model 2 is an enhanced form of the symbology with additional features (primarily, the addition of alignment patterns to assist navigation in larger symbols) and is the basis of the first edition of this document (i.e. ISO/IEC 18004:2000).
- QR code [the basis of the second edition of this document (i.e. ISO/IEC 18004:2006)] is very similar to QR code model 2; its QR code format differs only in the addition of the facility for symbols to appear in a mirror image orientation for reflectance reversal (light symbols on dark backgrounds) and the option for specifying alternative character is set to the default.
- The micro QR code format [also specified in the second edition of this document (i.e. ISO/IEC 18004:2006)], is a variant of QR code with a reduced number of overhead modules and a restricted range of sizes, which enables small to moderate amount 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.

QR code is a matrix symbology. The symbols consist of an array of nominally square modules arranged in an overall square pattern, including a unique finder pattern located at three corners of the symbol (in micro QR code symbols, at a single corner) and intended to assist in easy location of its position, size and inclination. A wide range of sizes of symbol is provided for, together with four levels of error correction. Module dimensions are user-specific to enable symbol production by a wide variety of techniques.

QR code model 2 symbols are fully compatible with QR code reading systems.

QR code model 1 symbols are recommended only to be used in closed system applications. Equipment complying with this document are not required to support QR code model 1 symbols. Since QR code is the recommended model for new, open system application of QR code, this document describes QR code fully. This document also lists the features in which QR code model 1 differs from QR code in [Annex N](#).

Information technology — Automatic identification and data capture techniques — QR code bar code symbology specification

1 Scope

This document specifies requirements for the symbology known as “QR code”. It specifies the QR code symbology characteristics, data character encoding methods, symbol formats, dimensional characteristics, error correction rules, reference decoding algorithm, production quality requirements and user-selectable application parameters.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 15415, *Information technology — Automatic identification and data capture techniques — Bar code symbol print quality test specification — Two-dimensional symbols*

ISO/IEC 15424:—¹⁾, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

ISO/IEC 19762, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

3 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 character count indicator

bit sequence which defines the data string length in a *mode* ([3.10](#))

3.2 data masking

process of XORing the bit pattern in the *encoding region* ([3.4](#)) with a data mask 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

3.3 data mask pattern reference

3-bit identifier of the *data masking* ([3.2](#)) patterns applied to the symbol

1) Under preparation. Stage at the time of publication: ISO/IEC DIS 15424:2024.

3.4

encoding region

region of the symbol not occupied by *function patterns* (3.8) and available for encoding data and error correction codewords, and for *version* (3.20) and *format information* (3.7)

3.5

exclusive subset

subset of characters within the character set of a *mode* (3.10) which are not shared with the more restricted character set of another mode

3.6

extension pattern

function pattern (3.8) in model 1 symbols, which does not encode data

3.7

format information

encoded pattern containing information on symbol characteristics essential to enable the remainder of the *encoding region* (3.4) to be decoded

3.8

function pattern

overhead component of the symbol (*finder*, *separator* (3.16), *timing patterns* (3.19) and alignment patterns) required for location of the symbol or identification of its characteristics to assist in decoding

3.9

masking

process of XORing the bit pattern in an area of the symbol with a mask pattern to reduce the occurrence of patterns which would interfere with fast processing of the image

3.10

mode

method of representing a defined character set as a bit string

3.11

mode indicator

1-bit to 4-bit identifier (depending on symbol size) indicating in which *mode* (3.10) the following data sequence is encoded

3.12

padding bit

zero bit, not representing data, used to fill empty positions of the final codeword after the *terminator* (3.18) in a data bit string

3.13

remainder bit

zero bit, not representing data, used to fill empty positions of the symbol *encoding region* (3.4) after the final symbol character, where the area of the encoding region available for symbol characters does not divide exactly into 8-bit symbol characters

3.14

remainder codeword

pad codeword, placed after the error correction codewords, used to fill empty codeword positions to complete the symbol if the total number of data and error correction codewords does not exactly fill its nominal capacity

3.15

segment

sequence of data encoded according to the rules of one extended channel interpretation or encoding mode

3.16

separator

function pattern (3.8) of all light modules, one module wide, separating the finder patterns from the rest of the symbol

3.17

symbol number

3-bit field indicating the symbol *version* (3.20) and error correction level applied, used as part of the *format information* (3.7) in micro QR code symbols

3.18

terminator

bit pattern of defined number (depending on the symbol) of all zero bits used to end the bit string representing data

3.19

timing pattern

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

3.20

version

size of the symbol represented in terms of its position in the sequence of permissible sizes for micro QR code symbols from 11×11 modules (version M1) to 17×17 modules (version M4) or, for QR code symbols, from 21×21 modules (version 1) to 177×177 (version 40) modules

Note 1 to entry: The error correction level applied to the symbol can be suffixed to the version designation, e.g. version 4-L or version M3-Q.

3.21

version information

encoded pattern in certain QR code symbols containing information on symbol *version* (3.20) together with error correction bits for this data

4 Mathematical and logical symbols, abbreviated terms and conventions

4.1 Mathematical and logical symbols

div integer division operator

mod integer remainder after division

XOR exclusive-or logic function whose output is one only when its two inputs are not equivalent
NOTE XOR is represented by the symbol $\oplus$.

4.2 Abbreviated terms

BCH Bose-Chaudhuri-Hocquenghem

DPM direct part mark

ECI extended channel interpretation

FNC1 function 1 symbol character

RS Reed-Solomon

4.3 Conventions

4.3.1 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. The module $(0, 0)$ is therefore located at the upper left corner of the symbol.

4.3.2 Byte notation

Byte content is shown as a hex value.

4.3.3 Version references

For QR code symbols, symbol versions are referred to in the form version V-E where V identifies the version number (1 to 40) and E indicates the error correction level (L, M, Q, H).

For micro QR code symbols, symbol versions are referred to in the form version MV-E where the letter M indicates the micro QR code format, V identifies the version number (with a range of 1 to 4) and E indicates the error correction level (with values L, M and Q).

5 Symbol description

5.1 Basic characteristics

QR code is a matrix symbology with the following characteristics:

- a) Formats:
 - 1) QR code, with full range of capabilities and maximum data capacity;
 - 2) micro QR code, with reduced overhead, some restrictions on capabilities and reduced data capacity (compared with QR code symbols).
- b) Encodable character set:
 - 1) numeric data (digits 0 - 9);
 - 2) alphanumeric data (digits 0 - 9; upper case letters A - Z; nine other characters: space, \$ % * + - . /:);
 - 3) byte data [default: ISO/IEC 8859-1; or other sets as otherwise defined (see [7.3.5](#))];
 - 4) Kanji characters (Kanji characters in QR code can be compacted into 13 bits).
- c) Representation of data: a dark module is nominally a binary one and a light module is nominally a binary zero. However, see [5.2](#) for details of reflectance reversal.
- d) Symbol size (not including quiet zone):
 - 1) for micro QR code symbols: 11×11 modules to 17×17 modules (versions M1 to M4, increasing in steps of two modules per side);
 - 2) for QR code symbols: 21×21 modules to 177×177 modules (versions 1 to 40, increasing in steps of four modules per side).
- e) Data characters per symbol:
 - 1) maximum micro QR code symbol size, version M4-L:
 - numeric data: 35 characters;

- alphanumeric data: 21 characters;
 - byte data: 15 characters;
 - Kanji data: 9 characters;
- 2) maximum QR code symbol size, version 40-L:
- numeric data: 7 089 characters;
 - alphanumeric data: 4 296 characters;
 - byte data: 2 953 characters;
 - Kanji data: 1 817 characters.
- f) Selectable error correction: four levels of Reed-Solomon error correction (referred to as L, M, Q and H in increasing order of capacity) allowing the symbol codeword recovery of
- L: 7 %;
 - M: 15 %;
 - Q: 25 %;
 - H: 30 %.

For micro QR code symbols, error correction level H is not available. For version M1 micro QR code symbols, the RS capacity is limited to error detection only.

- g) Code type: matrix.
- h) Orientation independence: yes (both rotation and reflection).

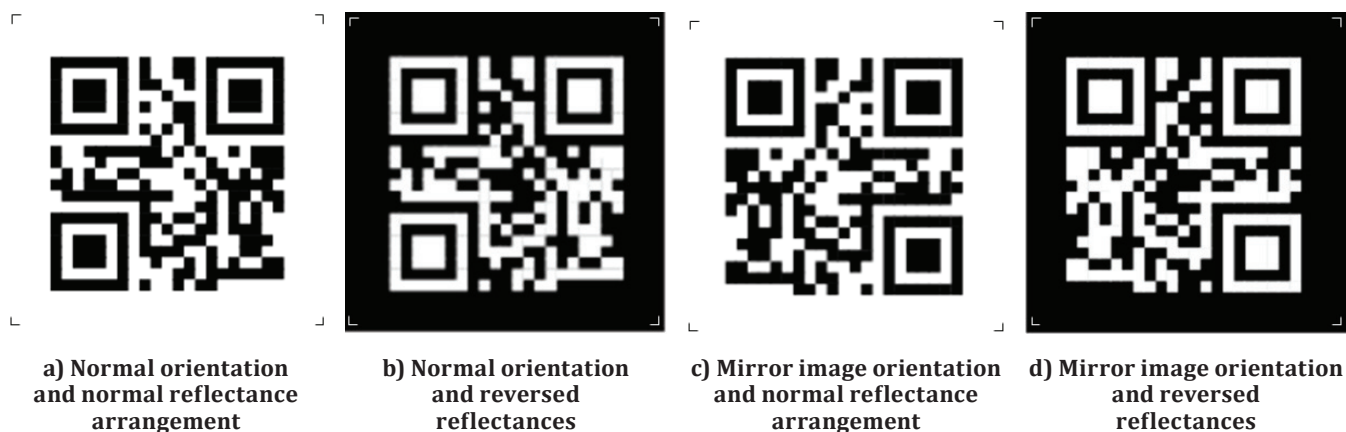
[Figure 1](#) illustrates a version 1 QR code symbol in normal colour and with reflectance reversal (see [5.2](#)), in both normal and mirror image orientations.

[Figure 2](#) illustrates a version M2 micro QR code symbol in normal colour and with reflectance reversal (see [5.2](#)), in both normal and mirror image orientations.

5.2 Summary of additional features

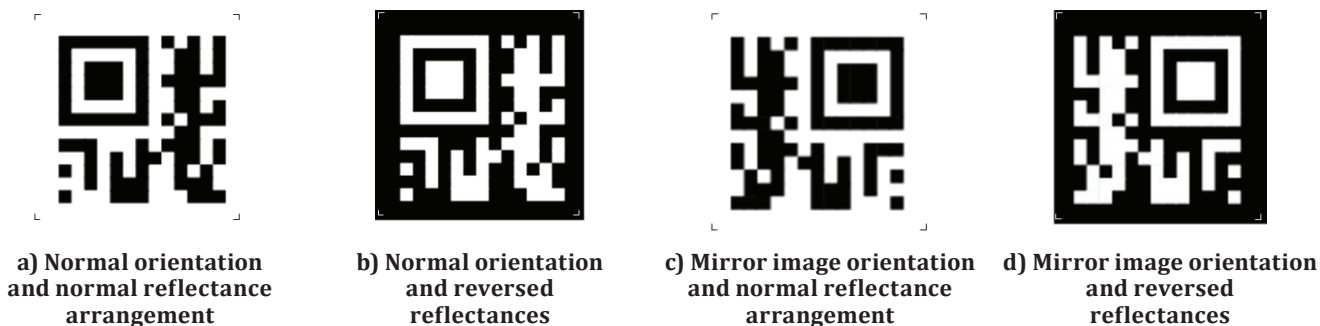
The use of the following additional features is optional in QR code.

- Structured append: This allows files of data to be represented logically and continuously in up to 16 QR code symbols. These can be scanned in any sequence to enable the original data to be correctly reconstructed. Structured append is not available with micro QR code symbols.
- Extended channel interpretations: This mechanism enables data using character sets other than the default encodable set (e.g. Arabic, Cyrillic, Greek) and other data interpretations (e.g. compacted data using defined compression schemes) or other industry-specific requirements to be encoded. ECIs other than the default interpretation are not available in micro QR code symbols.
- Reflectance reversal: Symbols are intended to be read when marked so that the image is either dark on light or light on dark (see [Figures 1](#) and [2](#)). The specifications in this document 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.
- Mirror imaging: The arrangement of modules defined in this document 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 patterns in the top left, top right and bottom left corners of the symbol, mirror imaging interchanges the row and column positions of the modules.



NOTE The corner marks indicate the extent of the quiet zone.

Figure 1 — Examples of a QR code symbol encoding the text "QR code Symbol"



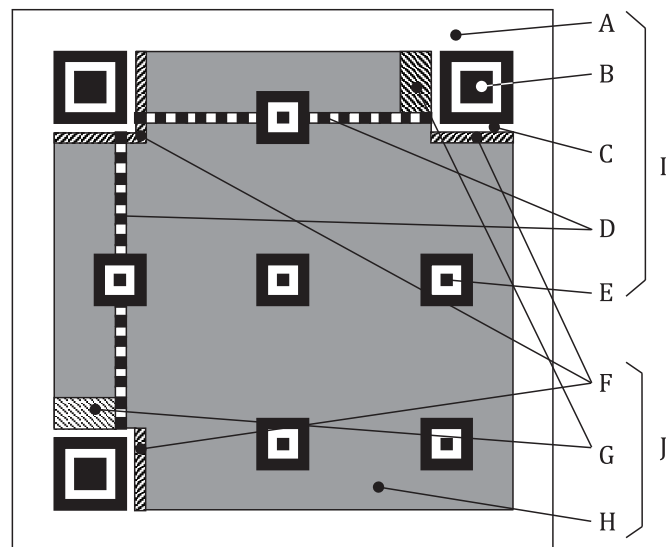
NOTE The corner marks indicate the extent of the quiet zone.

Figure 2 — Examples of a version M2 micro QR code symbol encoding the text "01234567"

5.3 Symbol structure

5.3.1 General

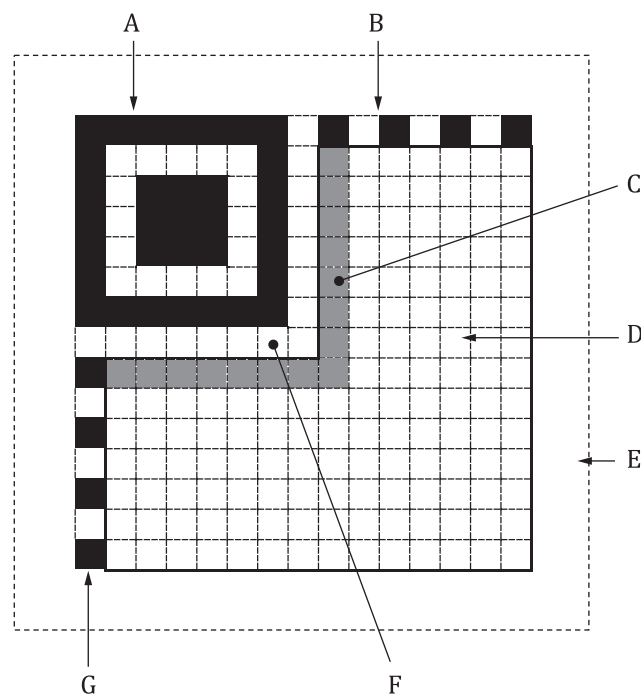
Each QR code symbol shall be constructed of nominally square modules set out in a regular square array and shall consist of an encoding region and function patterns, namely finder, separator, timing patterns and alignment patterns. Function patterns do not encode data. The symbol shall be surrounded on all four sides by a quiet zone border. [Figure 3](#) illustrates the structure of a version 7 symbol. [Figure 4](#) illustrates the structure of a version M3 symbol.



Key

- | | | | |
|---|--------------------|---|-------------------------------------|
| A | quiet zone | F | format information |
| B | finder pattern | G | version information |
| C | separator | H | data and error correction codewords |
| D | timing patterns | I | function patterns |
| E | alignment patterns | J | encoding region |

Figure 3 — Structure of a QR code symbol



Key

- | | | | |
|---|---------------------------|---|----------------|
| A | finder pattern quiet zone | E | quiet zone |
| B | timing patterns | F | separator |
| C | format information | G | timing pattern |
| D | encoding region | | |

Figure 4 — Structure of a version M3 micro QR code symbol

5.3.2 Symbol versions and sizes

5.3.2.1 QR code symbols

There are 40 sizes of QR code symbol referred to as version 1, version 2 ... version 40. Version 1 measures 21×21 modules, version 2 measures 25×25 modules and so on, increasing in steps of 4 modules per side up to version 40 which measures 177×177 modules. [Figures 5](#) to [10](#) illustrate the structure of versions 1, 2, 6, 7, 14, 21 and 40.

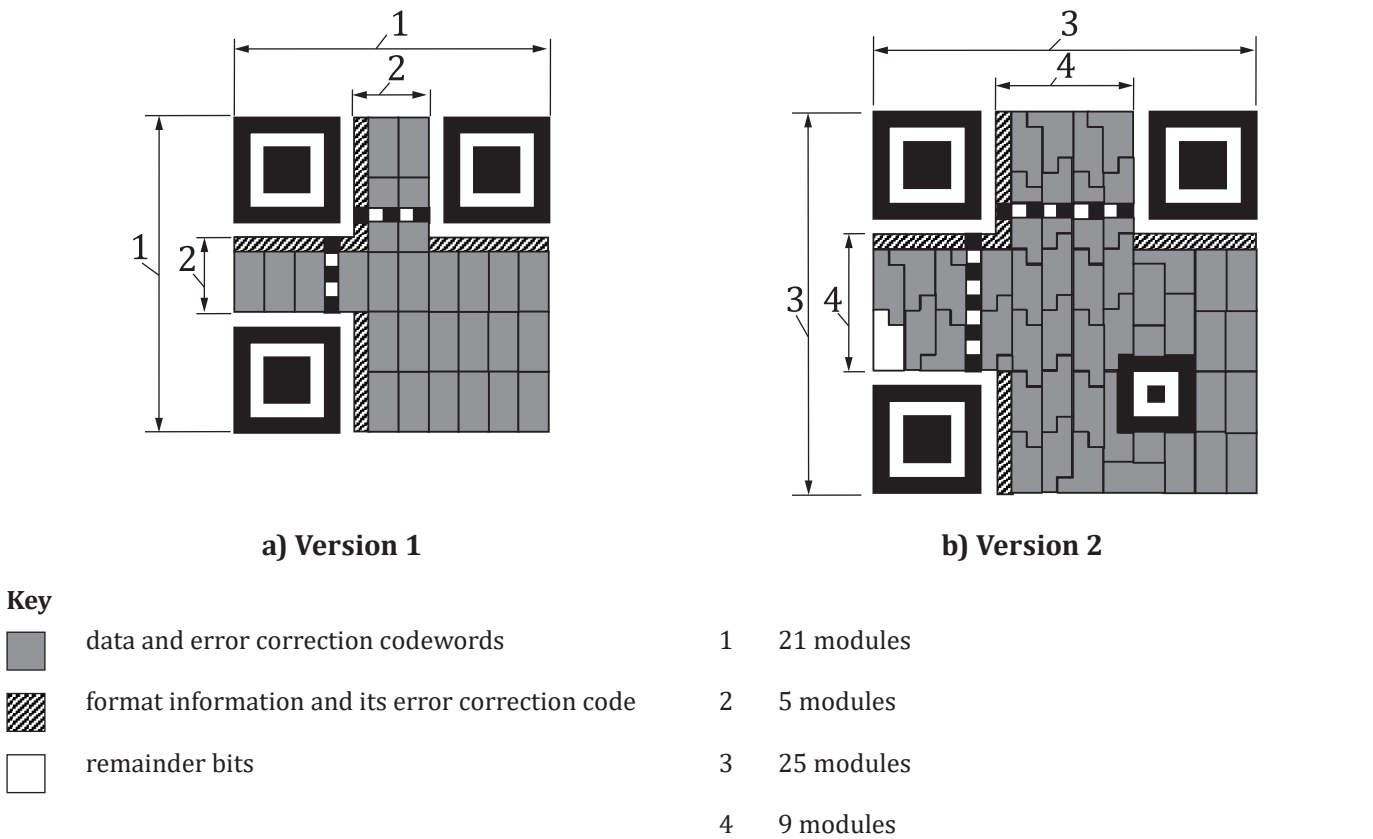
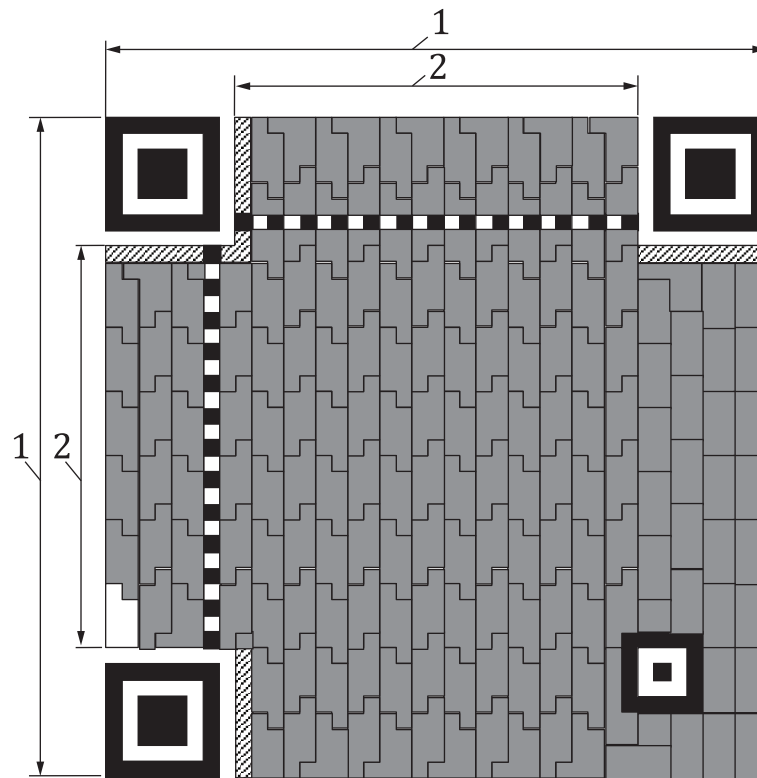


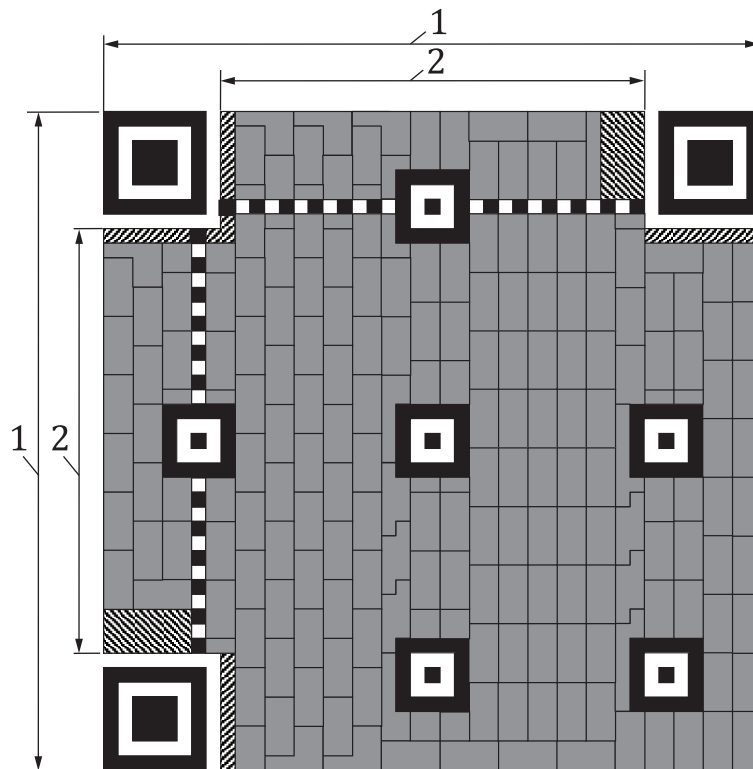
Figure 5 — Versions 1 and 2 symbols



Key

	data and error correction codewords	1	41 modules
	format information and its error correction code	2	25 modules
	remainder bits		

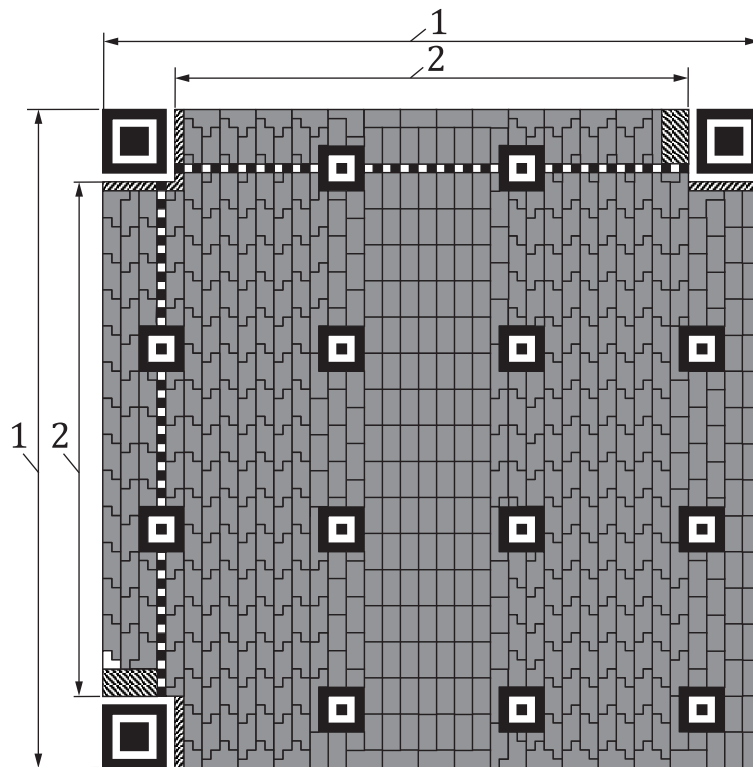
Figure 6 — Version 6 symbol



Key

	Data and error correction codewords	1	45 modules
	format information and its error correction code	2	29 modules
	version information and its error correction code		

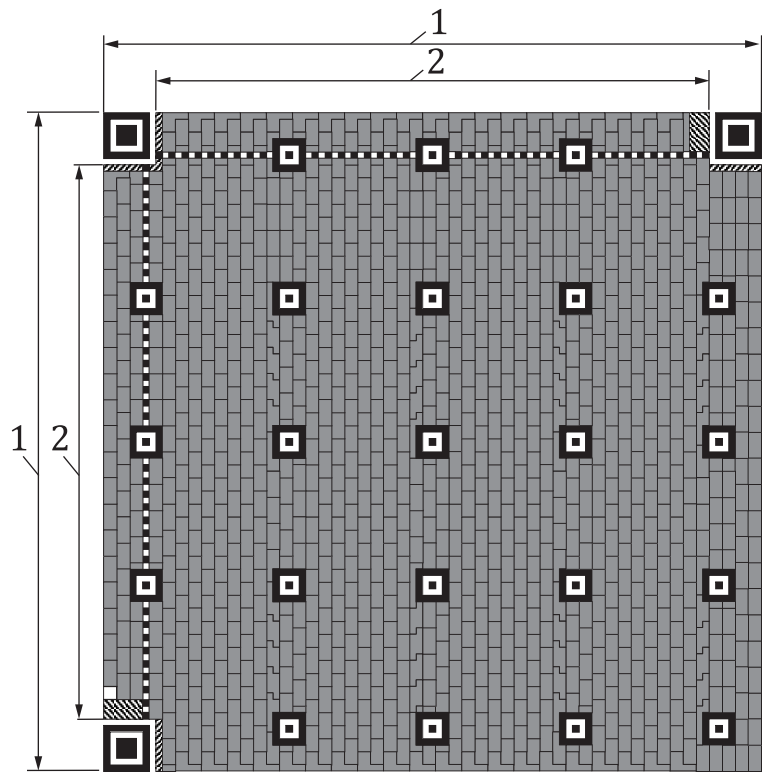
Figure 7 — Version 7 symbol



Key

	data and error correction codewords	1	73 modules
	format information and its error correction code	2	57 modules
	version information and its error correction code		
	remainder bits		

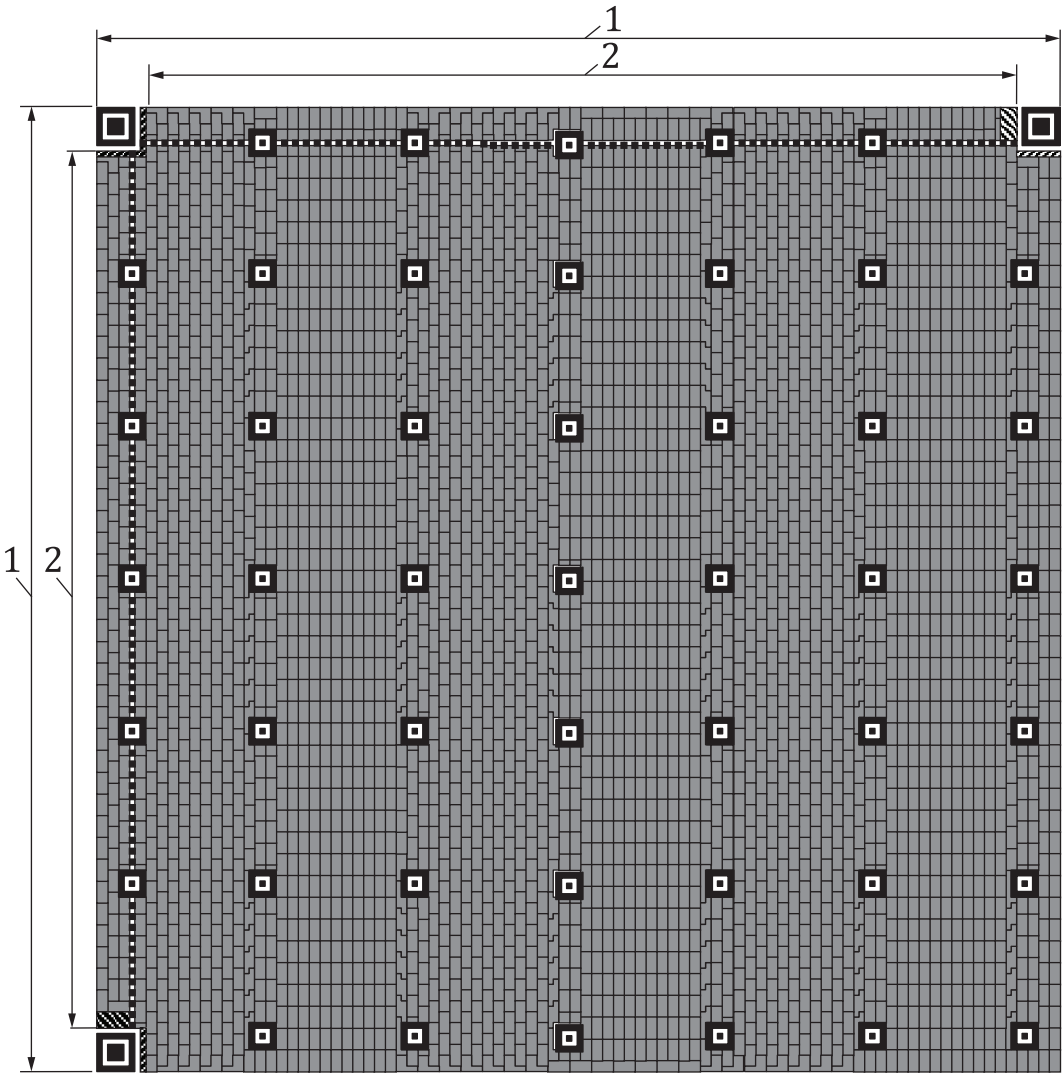
Figure 8 — Version 14 symbol



Key

	data and error correction codewords	1	101 modules
	format information and its error correction code	2	85 modules
	version information and its error correction code		
	remainder bits		

Figure 9 — Version 21 symbol



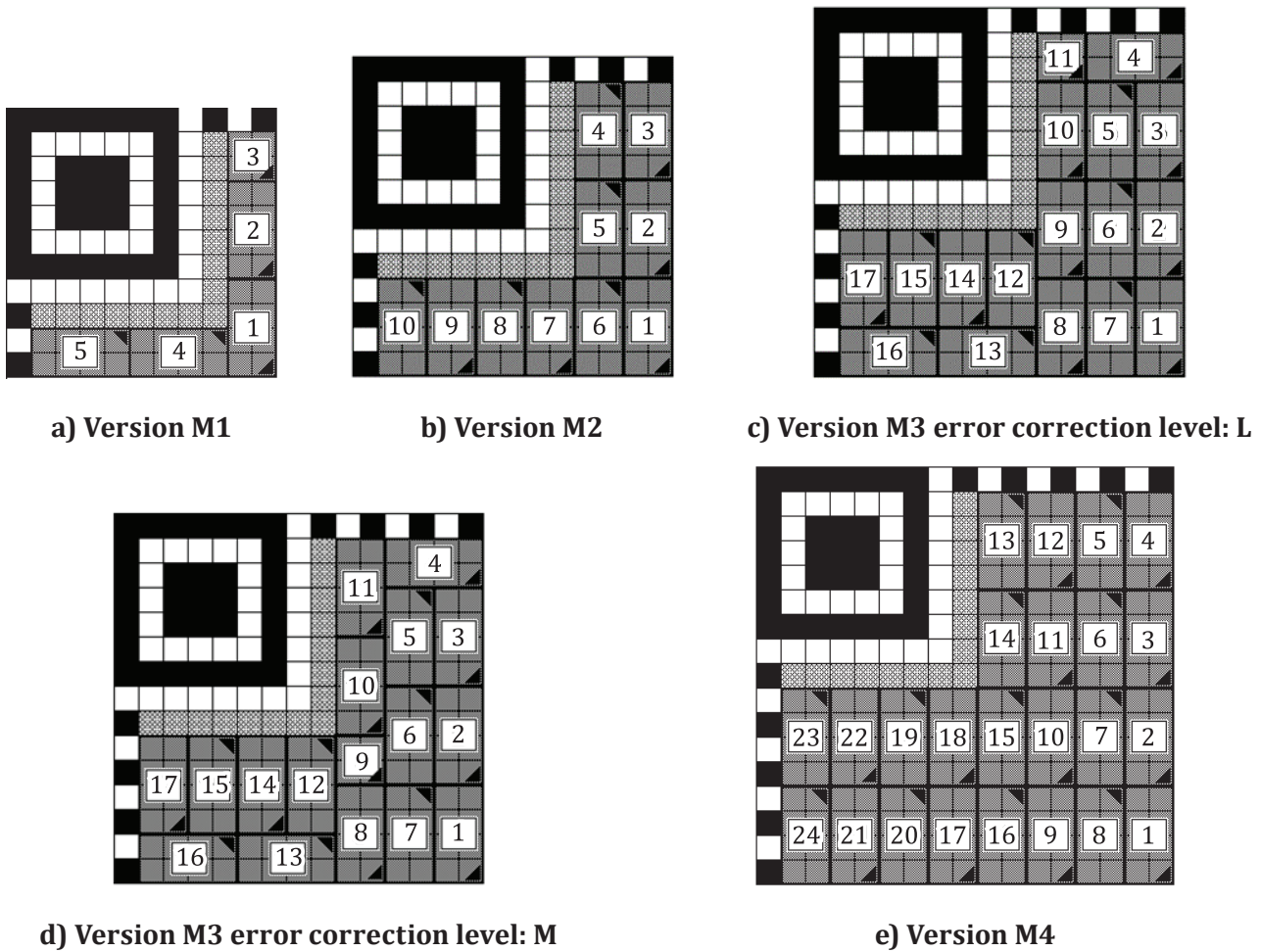
Key

	data and error correction codewords	1	177 modules
	format information and its error correction code	2	161 modules
	version information and its error correction code		

Figure 10 — Version 40 symbol

5.3.2.2 Micro QR code symbols

There are four sizes of micro QR code symbol, referred to as versions M1 to M4. Version M1 measures 11 × 11 modules, version M2 measures 13 × 13 modules, version M3 measures 15 × 15 modules and version M4 measures 17 × 17 modules, i.e. increasing in steps of 2 modules per side. [Figure 11](#) illustrates the structure of micro QR code versions M1 to M4.



NOTE Two formats of M3 symbol are shown, which differ only in the codeword placement according to the error correction level.

Figure 11 — Versions of a micro QR code symbol

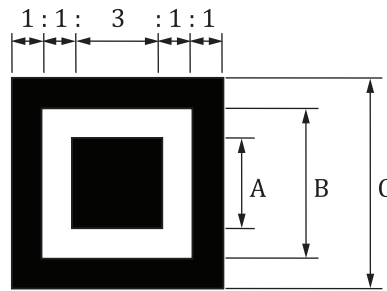
5.3.3 Finder pattern

5.3.3.1 QR code symbols

There are three identical finder patterns located at the upper left, upper right and lower left corners of the symbol respectively as illustrated in [Figure 3](#). Each finder pattern may be viewed as three superimposed concentric squares and is constructed of 7×7 dark modules, 5×5 light modules and 3×3 dark modules. The ratio of module widths in each finder pattern is 1:1:3:1:1 as illustrated in [Figure 12](#). 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 symbol in the field of view. The identification of the three finder patterns unambiguously defines the location and rotational orientation of the symbol in the field of view.

5.3.3.2 Micro QR code symbols

A single finder pattern, as defined in [5.3.3.1](#), is located at the upper left corner of the symbol as illustrated in [Figure 4](#). Identification of the finder pattern together with the timing patterns unambiguously defines the size, location and rotational orientation of the symbol in the field of view.

**Key**

A 3 modules

C 7 modules

B 5 modules

Figure 12 — Structure of a finder pattern**5.3.4 Separator**

A one-module wide separator, constructed of all light modules, is placed between each finder pattern and the encoding region, as illustrated in [Figures 3](#) and [4](#).

5.3.5 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. They enable the symbol density and version to be determined and provide datum positions for determining module coordinates.

In QR code symbols, the horizontal timing pattern runs across row 6 of the symbol between the separators for the upper finder patterns; the vertical timing pattern similarly runs down column 6 of the symbol between the separators for the left-hand finder patterns. See [Figure 3](#).

In micro QR code symbols, 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 timing pattern similarly runs down column 0 of the symbol below the separator to the bottom edge of the symbol. See [Figure 4](#).

5.3.6 Alignment patterns

Alignment patterns are present only in QR code symbols of version 2 or larger. Each alignment pattern may be viewed as three superimposed concentric squares and is constructed of 5×5 dark modules, 3×3 light modules and a single central dark module. The number of alignment patterns depends on the symbol version and they shall be placed in all symbols of version 2 or larger in positions defined in [Annex E](#).

5.3.7 Encoding region

The encoding region shall contain the symbol characters representing data, those representing error correction codewords, the format information and, where appropriate, the version information. Refer to [7.7.1](#) for details on symbol characters. Refer to [7.9](#) for details on format information. Refer to [7.10](#) for details on version information.

5.3.8 Quiet zone

The quiet region 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.

For QR code symbols, its width shall be $4X$.

For micro QR code symbols, its width shall be $2X$.

6 Conformance

QR code symbols (and equipment designed to produce or read QR code symbols) shall be considered as conforming with this document if they provide or support the features defined in this document.

Symbols complying with the requirements for QR code model 1, as described in the first edition of this document (i.e. ISO/IEC 18004:2000), can be unreadable with equipment complying with this document.

Symbols complying with the requirements for QR code model 2, as defined in the first edition of this document (i.e. ISO/IEC 18004:2000), are readable with equipment complying with this document.

Reading equipment complying with the first edition of this document (i.e. ISO/IEC 18004:2000) are not able to read all symbols complying with this document. Symbols that make use of the additional features of QR code are not readable by such equipment.

Printing equipment complying with the first edition of this document (i.e. ISO/IEC 18004:2000) are not able to print all symbols defined in this document. Symbols that make use of the additional features of QR code are not printable by such equipment.

It should be noted, however, that QR code model 2 and micro QR code are the form of the symbology recommended for new and open systems applications.

7 Requirements

7.1 Encode procedure overview

7.1.1 General

[Subclause 7.1](#) provides an overview of the steps required to convert input data to a QR code symbol.

7.1.2 Step 1: Data analysis

Analyse the input data stream to identify the variety of different characters to be encoded. The QR code format (but not the micro QR code format) supports the ECI feature, enabling data differing from the default character set to be encoded. 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 detection and correction level. If the user has not specified the symbol version to be used, select the smallest version that will accommodate the data. A complete list of symbol versions and capacities is shown in [Table 1](#).

7.1.3 Step 2: Data encoding

Convert the data characters into a bit stream in accordance with the rules for the mode in force, as defined in [7.4.3](#) to [7.4.7](#), inserting mode indicators as necessary to change modes at the beginning of each new mode segment, and a terminator at the end of the data sequence. Split the resulting bit stream into 8-bit codewords. Add pad characters as necessary to fill the number of data codewords required for the version.

7.1.4 Step 3: Error correction coding

Divide the codeword sequence into the required number of blocks (as defined in [Table 9](#)) to enable the error correction algorithms to be processed. Generate the error correction codewords for each block, appending the error correction codewords to the end of the data codeword sequence.

7.1.5 Step 4: Structure final message

Interleave the data and error correction codewords from each block as described in [7.6 c\)](#) and add remainder bits as necessary.

7.1.6 Step 5: Module placement in matrix

Place the codeword modules in the matrix together with the finder pattern, separators, timing pattern and, if required, alignment patterns.

7.1.7 Step 6: Data masking

Apply the data 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.

7.1.8 Step 7: Format and version information

Generate the format information and, where applicable, the version information. Complete the symbol.

Table 1 — Codeword capacity of all versions of QR code

Version	Number of modules per side <i>A</i>	Function pattern modules <i>B</i>	Format and version information modules <i>C</i>	Data modules except <i>D</i> ^b	Data capacity [codewords] ^a <i>E</i>	Remainder bits
M1	11	70	15	36	5	0
M2	13	74	15	80	10	0
M3	15	78	15	132	17	0
M4	17	82	15	192	24	0
1	21	202	31	208	26	0
2	25	235	31	359	44	7
3	29	243	31	567	70	7
4	33	251	31	807	100	7
5	37	259	31	1 079	134	7
6	41	267	31	1 383	172	7
7	45	390	67	1 568	196	0
8	49	398	67	1 936	242	0
9	53	406	67	2 336	292	0
10	57	414	67	2 768	346	0
11	61	422	67	3 232	404	0
12	65	430	67	3 728	466	0
13	69	438	67	4 256	532	0
14	73	611	67	4 651	581	3
15	77	619	67	5 243	655	3
16	81	627	67	5 867	733	3
17	85	635	67	6 523	815	3
18	89	643	67	7 211	901	3
19	93	651	67	7 931	991	3
20	97	659	67	8 683	1 085	3
21	101	882	67	9 252	1 156	4
22	105	890	67	10 068	1 258	4
23	109	898	67	10 916	1 364	4
24	113	906	67	11 796	1 474	4
25	117	914	67	12 708	1 588	4
26	121	922	67	13 652	1 706	4
27	125	930	67	14 628	1 828	4
28	129	1 203	67	15 371	1 921	3

^a All codewords are 8 bits in length, except in versions M1 and M3 where the final data codeword is 4 bits in length.

^b $D = A^2 - B - C$

Table 1 (continued)

Version	Number of modules per side <i>A</i>	Function pattern modules <i>B</i>	Format and version information modules <i>C</i>	Data modules except <i>D^b</i>	Data capacity [codewords] ^a <i>E</i>	Remainder bits
29	133	1 211	67	16 411	2 051	3
30	137	1 219	67	17 483	2 185	3
31	141	1 227	67	18 587	2 323	3
32	145	1 235	67	19 723	2 465	3
33	149	1 243	67	20 891	2 611	3
34	153	1 251	67	22 091	2 761	3
35	157	1 574	67	23 008	2 876	0
36	161	1 582	67	24 272	3 034	0
37	165	1 590	67	25 568	3 196	0
38	169	1 598	67	26 896	3 362	0
39	173	1 606	67	28 256	3 532	0
40	177	1 614	67	29 648	3 706	0

^a All codewords are 8 bits in length, except in versions M1 and M3 where the final data codeword is 4 bits in length.

^b $D = A^2 - B - C$

7.2 Data analysis

Analyse the input data string to determine its content and select the default or other appropriate ECI and 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, parts of which can more efficiently be encoded in one mode than other parts, for example, numeric sequences followed by alphanumeric sequences. In theory, it is 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 does not always result in the shortest overall bit stream to change modes for a small number of characters. Also, as the capacity of symbols increases in discrete steps from one version to the next, it is not always necessary to achieve the maximum conversion efficiency in every case. Guidance on minimising the bit stream length is given in Annex J. In micro QR code symbols, there are restrictions on the modes available for smaller versions. Clause J.2 shows appropriate micro QR code symbol versions for various combinations of two modes.

7.3 Modes

7.3.1 General

The modes described in 7.3 are based on the character values and assignments associated with the default ECI. When any other ECI is in force (in QR code symbols only), the byte values rather than the specific character assignments shall be used to select the optimum data compaction mode. For example, the numeric mode would be appropriate if there is a sequence of data byte values within the range 30_{HEX} to 39_{HEX} inclusive. In this case, the compaction is carried out using the default numeric or alphabetic equivalents of the byte values.

7.3.2 Extended channel interpretation mode

The ECI protocol defined in Reference [21] allows the output data stream to have interpretations different from that of the default character set. The ECI protocol is defined consistently across many symbologies. The ECI protocol provides a consistent method to specify particular interpretations of byte values before printing and after decoding.

The ECI protocol is not supported in micro QR code symbols.

The default interpretation for QR code is ECI 000003 representing the ISO/IEC 8859-1 character set. International applications using other character sets should use the ECI protocol. For instance, the interpretation corresponding to the JIS8 (JIS X 0201, 7-bit and 8-bit coded character sets for information interchange) and shift JIS (JIS X 0208:1997, 7-bit and 8-bit double byte coded Kanji sets for information interchange) character sets is ECI 000020. The ECI mode causes the insertion of an ECI escape sequence at that point in the data. It is immediately followed by another mode indicator (e.g. for efficient data encoding) and remains in force until the end of the message or a subsequent ECI mode indicator.

7.3.3 Numeric mode

The numeric mode encodes data from the decimal digit set (0 - 9) (byte values 30_{HEX} to 39_{HEX}). Normally, three data characters are represented by 10 bits.

7.3.4 Alphanumeric mode

The alphanumeric mode encodes data from a set of 45 characters, i.e. 10 numeric digits (0 - 9) (byte values 30_{HEX} to 39_{HEX}), 26 alphabetic characters (A - Z) (byte values 41_{HEX} to 5A_{HEX}), and 9 symbols (SP, \$, %, *, +, -, ., /, :) (byte values 20_{HEX}, 24_{HEX}, 25_{HEX}, 2A_{HEX}, 2B_{HEX}, 2D to 2F_{HEX}, 3A_{HEX}, respectively). Normally, two input characters are represented by 11 bits.

The alphanumeric mode is not available in version M1 micro QR code symbols.

7.3.5 Byte mode

In byte mode, data is encoded at 8 bits per character.

In closed-system national or application-specific implementations of QR code, an alternative 8-bit character set, for example as defined in an appropriate part of the ISO/IEC 8859 series, may be specified for byte mode. When an alternative character set is specified, however, the parties intending to read the QR code symbols require to be notified of the applicable character set in the application specification or by bilateral agreement.

Byte mode is not available in version M1 or M2 micro QR code symbols.

7.3.6 Kanji mode

The Kanji mode efficiently encodes Kanji characters according to the shift JIS system based on JIS X 0208. The shift JIS values are shifted from the JIS X 0208 values. JIS X 0208 gives details of the shift coded representation. Each two-byte character value is compacted to a 13-bit binary codeword.

When the character set specified for 8-bit byte mode makes use of byte values in the ranges of either 81_{HEX} to 9F_{HEX} or E0_{HEX} to EB_{HEX}, or both, it is not always possible to use Kanji mode unambiguously, as reading systems will be unable to determine from the transmitted data whether such byte values are the lead byte of a double byte character. It can be possible to achieve a shorter bit stream by using the Kanji mode compaction rules when an appropriate sequence of byte values occurs in the data (i.e. lead bytes in the ranges of either 81_{HEX} to 9F_{HEX} or E0_{HEX} to EB_{HEX}, or both, followed by trailer bytes in the range 40_{HEX} to FC_{HEX}, except 7F_{HEX}, or EB_{HEX} followed by 40_{HEX} to BF_{HEX}). [Figure H.1](#) shows the byte combinations graphically.

The Kanji mode is not available in version M1 or M2 micro QR code symbols.

7.3.7 Mixing modes

The QR code symbol can contain sequences of data in a combination of any of the modes described in [7.3.2](#) to [7.3.8](#). Micro QR code symbols may contain sequences of data in a combination of any of the modes available for the version of the symbol and described in [7.3.3](#) to [7.3.7](#).

Refer to [Annex J](#) for guidance to select the most efficient way of representing a given input data string in multiple modes in QR code symbols and [Clause J.3](#) for the available versions of micro QR code symbols for given combinations of data in two modes.

7.3.8 Structured append mode

The structured append mode is used to split the encoding of the data from a message over a number of QR code symbols. All of the symbols require to be read and the data message can be reconstructed in the correct sequence. The structured append header is encoded in each symbol to identify the length of the sequence and the symbol's position in it, and verify that all the symbols read belong to the same message. Refer to [Clause 8](#) for details of encoding in structured append mode.

The structured append mode is not available for micro QR code symbols.

7.4 Data encoding

7.4.1 FNC1 mode

FNC1 mode is used for messages containing specific data formats. In the "first position" it designates data formatted according to GS1 General Specifications. In the "second position", data shall be formatted in accordance with a specific industry application listed in ISO/IEC 15424:—, Annex E. FNC1 mode applies to the entire symbol and is not affected by subsequent mode indicators.

NOTE "First position" and "second position" do not refer to actual locations but are based on the positions of the character in Code 128 symbols, when used in an equivalent manner.

The FNC1 mode is not available for micro QR code symbols.

7.4.2 Sequence of data

Input data is converted into a bit stream consisting of one or more segments each in a separate mode. In the default ECI, the bit stream commences with the first mode indicator. If the initial ECI is other than the default ECI, the bit stream commences with an ECI header, followed by the first segment.

The ECI header (if present) shall comprise:

- an ECI mode indicator (4 bits);
- an ECI designator (8 bits, 16 bits or 24 bits).

The ECI header shall begin with the first (most significant) bit of the ECI mode indicator and end with the final (least significant) bit of the ECI designator.

The remainder of the bit stream is then made up of segments each comprising:

- a mode indicator;
- a character count indicator;
- a data bit stream.

Each mode 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. There shall be no explicit separator between segments as their length is defined unambiguously by the rules for the mode in force and the number of input data characters.

To encode a sequence of input data in a given mode, the steps described in [7.4.3](#) to [7.4.8](#) shall be followed. [Table 2](#) defines the mode indicators for each mode. [Table 3](#) defines the length of the character count indicator, which varies according to the mode and the symbol version in use.

Table 2 — Mode indicators for QR code

Mode	QR code symbols for all versions	Micro QR code symbols			
		Version M1	Version M2	Version M3	Version M4
Mode indicator length (bits)	4	0	1	2	3
ECI	0111	NA	NA	NA	NA
Numeric	0001	NA	0	00	000
Alphanumeric	0010	NA	1	01	001
Byte	0100	NA	NA	10	010
Kanji	1000	NA	NA	11	011
Structured append	0011	NA	NA	NA	NA
FNC1 ^a	0101 (1 st position) 1001 (2 nd position)	NA	NA	NA	NA
Terminator (end of message) ^b	0000	000	00000	0000000	000000000
Key NA not applicable ^a See 7.4.9.2 and 7.4.9.3. ^b The terminator is not a mode indicator as such.					

Table 3 — Number of bits in the character count indicator for QR code

Version	Numeric mode	Alphanumeric mode	Byte mode	Kanji mode
M1	3	NA	NA	NA
M2	4	3	NA	NA
M3	5	4	4	3
M4	6	5	5	4
1 to 9	10	9	8	8
10 to 26	12	11	16	10
27 to 40	14	13	16	12
Key NA not applicable				

The end of the data in the complete symbol is indicated by a terminator consisting of between 3 and 9 zero bits (see Table 2), which is omitted or abbreviated if the remaining symbol capacity after the data bit stream is less than the required bit length of terminator. The terminator is not a mode indicator as such.

7.4.3 Extended channel interpretation mode

7.4.3.1 General

The ECI mode, used for encoding data subject to alternative interpretations of byte values (e.g. alternative character sets) according to the ECI protocol defined in Reference [21] which defines the pre-processing of this type of data, is invoked by the use of mode indicator 0111.

The ECI can only be used with readers enabled to transmit the symbology identifier. Readers that cannot transmit the symbology identifier cannot transmit the data from any symbol containing an ECI.

Input ECI data shall be handled by the encoding system as a series of byte values.

Data in an ECI sequence may be encoded in whatever mode or modes permit the most efficient encoding of the byte values of the data, irrespective of their significance. For example, a sequence of bytes in the range 30_{HEX} to 39_{HEX} can be encoded in numeric mode (see 7.4.4) as though it were a sequence of digits from 0 to 9,

even though it does not necessarily represent numeric data. In order to determine the value of the character count indicator, the number of bytes (or, in Kanji mode, of byte pairs) shall be used.

7.4.3.2 ECI designator

Each extended channel interpretation is designated by a 6-digit assignment number which is encoded in the QR code symbol as the first one, two or three codewords following the ECI mode indicator. The encoding rules are defined in [Table 4](#). The ECI designator appears in the data to be encoded as character 5C_{HEX} [\ or backslash (reverse solidus) in ISO/IEC 8859-1, ¥ or yen sign in JIS8] followed by the 6-digit assignment number. Where 5C_{HEX} appears as true data, it shall be doubled in the data string before encoding in symbols to which the ECI protocol applies.

When a single occurrence of 5C_{HEX} is encountered in the input to the decoder, an ECI mode indicator is inserted followed by the ECI designator. When a doubled 5C_{HEX} is encountered, it is encoded as two 5C_{HEX} bytes.

On decoding, the binary pattern of the first ECI Designator codeword (i.e. the codeword following the mode indicator in ECI mode), determines the length of the ECI Designator sequence. The number of 1 bits before the first 0 bit defines the number of additional codewords after the first used to represent the ECI assignment number. The bit sequence after the first 0 bit is the binary representation of the ECI assignment number. The lower numbered ECI assignments may be encoded in multiple ways, but the shortest way is preferred.

Table 4 — Encoding the ECI assignment number

ECI assignment value	Number of codewords	Codeword values ^a
000000 to 000127	1	0bbbbbbb
000000 to 016383	2	10bbbbbb bbbbbbbb
000000 to 999999	3	110bbbbb bbbbbbbb bbbbbbbb

^a where b ... b is the binary value of the ECI assignment number.

EXAMPLE When encoding data in Greek using a character set ISO/IEC 8859-7 (ECI 000009) a in version 1-H symbol:

Data to be encoded: \000009ABΓΔΕ (character values A1_{HEX}, A2_{HEX}, A3_{HEX}, A4_{HEX}, A5_{HEX})

Bit sequence in symbol:

ECI mode indicator: 0111

ECI assignment number (000009): 0 0001001

Mode indicator (byte): 0100

Character count indicator (5): 00000101

Data: 10100001 10100010 10100011 10100100 10100101

Final bit string: 0111 00001001 0100 00000101 10100001 10100010 10100011 10100100 10100101

See [14.3](#) for an example of transmission of this data following decoding.

7.4.3.3 Multiple extended channel interpretations

Refer to the ECI protocol defined in Reference [\[21\]](#) defining the effect of a subsequent ECI designator in an ECI data segment. For example, data to which a character set ECI has been applied can also be subject to encryption or compaction using a transformation ECI which will co-exist with the initial ECI, or a second character set ECI will have the effect of terminating the first ECI and starting a new ECI segment. Where any ECI designator appears in the data, it shall be encoded in the QR code symbol in accordance with [7.4.3.2](#) and shall commence a new mode segment.

7.4.3.4 ECIs and structured append

Any ECI(s) invoked shall apply subject to the rules defined in [7.4.3.3](#) and in the ECI protocol defined Reference [\[21\]](#) until the end of the encoded data or a change of ECI (signalled by mode indicator 0111). If the encoded data in the ECI(s) extends through two or more symbols in the structured append mode, it is necessary to provide

an ECI header consisting of ECI mode indicator and ECI designator number for each ECI in force, immediately following the structured append header, in subsequent symbols in which the ECI continues in force.

7.4.4 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 bits or 7 bits respectively. The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the numeric mode has either 4 bits for QR code symbols or the number of bits defined in [Table 2](#) for micro QR code symbols, and the character count indicator has the number of bits defined in [Table 3](#). The number of input data characters is converted to its binary equivalent and added as the character count indicator after the mode indicator and before the binary data sequence.

EXAMPLE 1 For a version 1-H symbol:

Input data:	01234567
a) The input data is divided into groups of three digits:	012 345 67
b) Each group is converted to its binary equivalent:	012 → 0000001100 345 → 0101011001 67 → 1000011
c) The binary data is connected in sequence:	0000001100 0101011001 1000011
d) The character count indicator is converted to binary (10 bits for version 1-H):	Number of input data characters: 8 → 0000001000
e) The mode indicator 0001 and character count indicator are added to binary data:	0001 0000001000 0000001100 0101011001 1000011

EXAMPLE 2 For a micro QR code version M3-M symbol:

Input data:	0123456789012345
a) The input data is divided into groups of three digits:	012 345 678 901 234 5
b) Each group is converted to its binary equivalent:	012 = 0000001100 345 = 0101011001 678 = 1010100110 901 = 1110000101 234 = 0011101010 5 = 0101
c) The binary data is connected sequentially:	0000001100 0101011001 1010100110 1110000101 0011101010 0101
d) The character count indicator is converted to binary (5 bits for version M3-M):	Number of input data characters: 16 = 10000
e) The mode indicator (00 for version M3-M) and character count indicator are added to binary data:	00 10000 0000001100 0101011001 1010100110 1110000101 0011101010 0101

For any number of data characters, the length of the bit stream in the numeric mode is given by [Formula \(1\)](#):

$$B = M + C + 10(D \div 3) + R \quad (1)$$

where

- B is the number of bits in bit stream;
- M is the number of bits in mode indicator (4 for QR code symbols or as shown in [Table 2](#) for micro QR code symbols);
- C is the number of bits in character count indicator (from [Table 3](#));
- D is the number of input data characters;
- R is 0 if $(D \bmod 3) = 0$, or is 4 if $(D \bmod 3) = 1$, or is 7 if $(D \bmod 3) = 2$.

7.4.5 Alphanumeric mode

Each input data character is assigned a character value V from 0 to 44 according to [Table 5](#).

Table 5 — Encoding and decoding table for the alphanumeric mode

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

Input data characters are divided into groups of two characters which are encoded as 11-bit binary codes. The character value of the first character is multiplied by 45 and the character value of the second digit is added to the product. The sum is then converted to an 11-bit binary number. If the number of input data characters is not a multiple of two, the character value of the final character is encoded as a 6-bit binary number. The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the alphanumeric mode has either 4 bits for QR code symbols or the number of bits defined in [Table 2](#) for micro QR code symbols, and the character count indicator has the number of bits defined in [Table 3](#). The number of input data characters is converted to its binary equivalent and added as the character count indicator after the mode indicator and before the binary data sequence.

In FNC1 mode, separator characters can be required as data fields are concatenated. Refer to [7.4.9.2](#), [7.4.9.3](#) and [14.4](#) for details on encoding and transmission.

EXAMPLE For a version 1-H symbol:

- Input data: AC-42
- a) Character values are determined according to [Table 5](#): AC-42 → (10,12,41,4,2)
 - b) The result is divided into groups of two decimal values: (10,12) (41,4) (2)
 - c) Each group is converted to its 11-bit binary equivalent:
 - (10,12) $10 \times 45 + 12 \rightarrow 462 \rightarrow 00111001110$
 - (41,4) $41 \times 45 + 4 \rightarrow 1849 \rightarrow 11100111001$
 - (2) $\rightarrow 2 \rightarrow 000010$
 - d) The binary data is connected in sequence: 00111001110 11100111001 000010
 - e) The character count indicator is converted to binary (9 bits for version 1-H):
 - Number of input data characters: 5 → 000000101
 - f) The mode indicator 0010 and character count indicator are added to binary data: 0010 000000101 00111001110 11100111001 000010

For any number of data characters, the length of the bit stream in alphanumeric mode is given by the formula:

$$B = M + C + 11(D \div 2) + 6(D \bmod 2)$$

where

B is the number of bits in bit stream;

M is the number of bits in mode indicator (4 for QR code symbols or as shown in [Table 2](#) for micro QR code symbols);

C is the number of bits in character count indicator (from [Table 3](#));

D is the number of input data characters.

7.4.6 Byte mode

In byte mode, one 8-bit codeword directly represents the byte value of the input data character, i.e. a density of 8 bits/character. [Table 6](#) shall apply for encoding and decoding byte mode characters.

Table 6 — Encoding and decoding table for the byte mode character set

Byte		Char.	Byte		Char.	Byte		Char.	Byte		Char.	Byte		Char.	Byte		Char.	Byte		Char.	Byte		Char.
Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
0	00	NUL	32	20	space	64	40	@	96	60	`	128	80		160	A0	NBSP	192	C0	À	224	E0	à
1	01	SOH	33	21	!	65	41	A	97	61	a	129	81		161	A1	ı	193	C1	Á	225	E1	á
2	02	STX	34	22	"	66	42	B	98	62	b	130	82		162	A2	¢	194	C2	Â	226	E2	â
3	03	ETX	35	23	#	67	43	C	99	63	c	131	83		163	A3	£	195	C3	Ã	227	E3	ã
4	04	EOT	36	24	\$	68	44	D	100	64	d	132	84		164	A4	¤	196	C4	Ä	228	E4	ä
5	05	ENQ	37	25	%	69	45	E	101	65	e	133	85		165	A5	¥	197	C5	Å	229	E5	å
6	06	ACK	38	26	&	70	46	F	102	66	f	134	86		166	A6	¦	198	C6	Æ	230	E6	æ
7	07	BEL	39	27	'	71	47	G	103	67	g	135	87		167	A7	§	199	C7	Ç	231	E7	ç
8	08	BS	40	28	(	72	48	H	104	68	h	136	88		168	A8	¨	200	C8	È	232	E8	è
9	09	HT	41	29	)	73	49	I	105	69	i	137	89		169	A9	©	201	C9	É	233	E9	é
10	0A	LF	42	2A	*	74	4A	J	106	6A	j	138	8A		170	AA	ª	202	CA	Ê	234	EA	ê
11	0B	VT	43	2B	+	75	4B	K	107	6B	k	139	8B		171	AB	«	203	CB	Ë	235	EB	ë
12	0C	FF	44	2C	,	76	4C	L	108	6C	l	140	8C		172	AC	¬	204	CC	Ì	236	EC	ì
13	0D	CR	45	2D	-	77	4D	M	109	6D	m	141	8D		173	AD	SHY	205	CD	Í	237	ED	í
14	0E	SO	46	2E	.	78	4E	N	110	6E	n	142	8E		174	AE	®	206	CE	Î	238	EE	î
15	0F	SI	47	2F	/	79	4F	O	111	6F	o	143	8F		175	AF	¯	207	CF	Ï	239	EF	ï
16	10	DLE	48	30	0	80	50	P	112	70	p	144	90		176	B0	°	208	D0	Ð	240	F0	ð
17	11	DC1	49	31	1	81	51	Q	113	71	q	145	91		177	B1	±	209	D1	Ñ	241	F1	ñ
18	12	DC2	50	32	2	82	52	R	114	72	r	146	92		178	B2	2	210	D2	Ò	242	F2	ò
19	13	DC3	51	33	3	83	53	S	115	73	s	147	93		179	B3	3	211	D3	Ó	243	F3	ó
20	14	DC4	52	34	4	84	54	T	116	74	t	148	94		180	B4	´	212	D4	Ô	244	F4	ô
21	15	NAK	53	35	5	85	55	U	117	75	u	149	95		181	B5	µ	213	D5	Õ	245	F5	õ
22	16	SYN	54	36	6	86	56	V	118	76	v	150	96		182	B6	¶	214	D6	Ö	246	F6	ö
23	17	ETB	55	37	7	87	57	W	119	77	w	151	97		183	B7	·	215	D7	×	247	F7	÷
24	18	CAN	56	38	8	88	58	X	120	78	x	152	98		184	B8	,	216	D8	Ø	248	F8	ø
25	19	EM	57	39	9	89	59	Y	121	79	y	153	99		185	B9	1	217	D9	Ù	249	F9	ù
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z	154	9A		186	BA	²	218	DA	Ú	250	FA	ú
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{	155	9B		187	BB	»	219	DB	Û	251	FB	û
28	1C	FS	60	3C	<	92	5C	\	124	7C		156	9C		188	BC	¼	220	DC	Ü	252	FC	ü
29	1D	GS	61	3D	=	93	5D	]	125	7D	}	157	9D		189	BD	½	221	DD	Ý	253	FD	ý
30	1E	RS	62	3E	>	94	5E	^	126	7E	~	158	9E		190	BE	¾	222	DE	Þ	254	FE	þ
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL	159	9F		191	BF	¿	223	DF	ß	255	FF	ÿ

NOTE 1 In the JIS8 character set (see [Table H.1](#)), byte values 80_{HEX} to 9F_{HEX} and E0_{HEX} to FF_{HEX} are not assigned but are reserved values. Some of those values are used as the first byte in the shift JIS character set (see [Figure H.1](#)) and can be used to distinguish between JIS8 and shift JIS character sets, or to enable Kanji mode compaction. JIS X 0208 gives details on shift coded representation.

NOTE 2 Byte values 00_{HEX} to 7F_{HEX} in the JIS8 character set correspond to ISO/IEC 8859-1 and ISO/IEC 646 IRV, except values 5C_{HEX} and 7E_{HEX}.

NOTE 3 The set of characters shown in [Table G.1](#) does not fully comply with ISO/IEC 8859-1. ISO/IEC 8859-1 contains only graphic characters with hexadecimal values 20-7E and A0-FF.

NOTE 4 The control functions given in [Table G.1](#) in positions with decimal values 00-31 (hexadecimal values 00-1F) correspond to the CO set of ISO/IEC 6429 where the characters with decimal values 28-31 (hexadecimal values 1C-1F) IS4, IS3, IS2, IS1 are replaced, respectively, by characters FS, GS, RS, US. The character DEL at the position with decimal value 127 (hexadecimal 7F) corresponds to the GO set of ISO/IEC 646.

The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the byte mode has either 4 bits for QR code symbols or the number of bits defined in [Table 2](#) for micro QR code symbols, and the character count indicator has the number of bits defined in [Table 3](#). The number of input data characters is converted to its binary equivalent and added after the mode indicator and before the binary data sequence.

For any number of data characters, the length of the bit stream in byte mode is given by [Formula \(2\)](#):

$$B = M + C + 8D \quad (2)$$

where

- B is the number of bits in bit stream;
- M is the number of bits in mode indicator (4 for QR code symbols or as shown in [Table 2](#) for micro QR code symbols);
- C is the number of bits in character count indicator (from [Table 3](#));
- D is the number of input data characters.

7.4.7 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 0208 values. JIS X 0208 gives details of the shift coded representation. Input data characters in Kanji mode are compacted to 13-bit binary codewords as defined in this subclause. The binary data is then concatenated and prefixed with the mode indicator and the character count indicator. The mode indicator in the numeric mode has either 4 bits for QR code symbols or the number of bits defined in [Table 2](#) for micro QR code symbols, and the character count indicator has the number of bits defined in [Table 3](#). The number of input data characters is converted to its binary equivalent and added as the character count indicator after the mode indicator and before the binary data sequence.

- 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 b), and
 - 4) convert result to a 13-bit binary string.
- b) For characters with shift JIS values from E040_{HEX} to EBBF_{HEX}
 - 1) subtract C140_{HEX} from shift JIS value,

- 2) multiply most significant byte of result by $C0_{\text{HEX}}$,
- 3) add least significant byte to product from b), and
- 4) convert result to a 13-bit binary string.

EXAMPLE

Input character	“点”	“茗”
(Shift JIS value):	935F	E4AA
i) 8140 or C140 is subtracted:	$935F - 8140 = 121F$	$E4AA - C140 = 236A$
ii) The most significant byte is multiplied by $C0$:	$12 \times C0 = D80$	$23 \times C0 = 1A40$
iii) The least significant byte is added:	$D80 + 1F = D9F$	$1A40 + 6A = 1AAA$
iv) It is converted to 13-bit binary:	$0D9F \rightarrow 0\ 1101\ 1001\ 1111$	$1AAA \rightarrow 1\ 1010\ 1010\ 1010$

- c) For all characters, prefix binary sequence representing input data characters with mode indicator (from [Table 2](#)) and character count indicator binary equivalent (number of bits defined in [Table 3](#)).

For any number of data characters, the length of the bit stream in Kanji mode is given by the formula:

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

where

- B is the number of bits in bit stream;
- M is the number of bits in mode indicator (4 for QR code symbols or as shown in [Table 2](#) for micro QR code symbols);
- C is the number of bits in character count indicator (from [Table 3](#));
- D is the number of input data characters.

7.4.8 Mixing modes

There is the option for a symbol to contain sequences of data in one mode and then to change modes if the data content requires it, or in order to increase the density of encoding. Refer to [Annex I](#) for guidance. Each segment of data is encoded in the appropriate mode as indicated in [7.4.3](#) to [7.4.7](#), with the basic structure mode indicator/character count indicator/data and followed immediately by the mode indicator commencing the next segment. [Figure 13](#) illustrates the structure of data containing n segments.

Segment 1			Segment 2			...	Segment n			Terminator
mode indicator 1	character count indicator	Data	mode indicator 2	character count indicator	Data	...	mode indicator n	character count indicator	Data	

Figure 13 — Format of mixed mode data

7.4.9 FNC1 modes

7.4.9.1 General

In QR code symbols, there are two mode indicators which are used cumulatively with those defined in [7.3.2](#) to [7.3.8](#) and [7.4.3](#) to [7.4.8](#) to identify symbols encoding messages formatted according to specific predefined industry or application specifications. These (together with any associated parameter data) precede the mode indicator(s) used to encode the data efficiently. When these mode indicators are used, it is necessary for the decoder to transmit the symbology identifier as defined in [14.2](#) and [Annex F](#).

7.4.9.2 FNC1 in first position

NOTE "First position" is not used in a literal sense but is a historical reference to the position of the FNC1 symbol character in Code 128 symbols.

This mode indicator identifies GS1 QR code symbols encoding data with GS1 application identifiers formatted according to the GS1 General Specifications. GS1 QR code symbology is a subset of QR code symbology. For this purpose, it shall only be used once in a symbol and shall be placed immediately before the first mode indicator used for efficient data encoding (numeric, alphanumeric, byte or Kanji), and after any ECI or structured append header. The FNC1 mode requires the use of a separator character to distinguish concatenated data fields from each other. See [14.4](#) for the detail of transmission of the data field separator characters.

EXAMPLE 1

Input data:	0104912345123459 (application identifier 01 = GS1 article number, fixed length; data: 04912345123459) 15970331 (application identifier 15 = "Best before" date YYMMDD, fixed length; data: 31 March 1997) 30128 (application identifier 30 = quantity, variable length; data: 128) (requires separator character) 10ABC123 (application identifier 10 = batch number, variable length; data: ABC123)
Data to be encoded:	01049123451234591597033130128%10ABC123
Bit sequence in symbol:	0101 (mode indicator, FNC1 implied in 1 st position) 0001 (mode indicator, numeric mode) 0000011101 (character count indicator, 29) <data bits for 01049123451234591597033130128> 0010 (mode indicator, alphanumeric mode) 000001001 (character count indicator, 9) <data bits for %10ABC123>
Transmitted data (see 14.2 and Annex F):	JQ301049123451234591597033130128<1D _{HEX} >10ABC123

EXAMPLE 2 Encoding and transmission of percentage character in data:

Input data:	123 %
Encoded as:	123 %%
Transmitted as:	123 %

7.4.9.3 FNC1 in second position

NOTE 1 "Second position" is not used in a literal sense but is a historical reference to the position of the FNC1 symbol character in Code 128 symbols.

This mode indicator identifies symbols formatted according to a specific industry or application specification listed in ISO/IEC 15424:—, Annex E. It is immediately followed by a one-byte codeword the value of which is that of the application indicator assigned to identify the specification concerned.

For this purpose, it shall only be used once in a symbol and shall be placed immediately before the first mode indicator used for efficient data encoding (numeric, alphanumeric, byte or Kanji), and after any ECI or structured append header. An application indicator can take the form of any single Latin alphabetic character from the set {a - z, A - Z} (represented by the ISO/IEC 646 IRV value of the character plus 100) or a two-digit number (represented by its numeric value directly) and shall be transmitted by the decoder as the first one or two characters immediately preceding the data. The FNC1 mode and some other application specifications can require the use of a separator character to distinguish concatenated data fields from each other. See [14.4](#) for details on the transmission of data field separator characters.

NOTE 2 At the time of the publication of this document, application indicator 37 has not been assigned to any organisation and the data content of the example is purely arbitrary.