INTERNATIONAL STANDARD

ISO/IEC 16022

First edition
2000-05-01

Information technology — International symbology specification — Data matrix

Technologies de l'information — Spécification internationale des symboles — Matrice de données

Reference number
ISO/IEC 16022:2000(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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 International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 16022 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

International Standard ISO/IEC 16022 was prepared by AIM International (as ANSI/AIM BC11) and was adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Annexes A to P form a normative part of this International Standard. Annexes Q to V are for information only.

Information technology — International symbology specification — Data matrix

Introduction

Data Matrix is a two-dimensional matrix symbology which is made up of square modules arranged within a perimeter finder pattern. Though primarily shown and described in this document as a dark symbol on light background, Data Matrix symbols can also be printed to appear as light on dark.

Manufacturers of bar code equipment and users of the technology require publicly available standard symbology specifications to which they can refer when developing equipment and application standards. The publication of Symbology Specifications is designed to achieve this.

1 Scope

This specification defines the requirements for the symbology known as Data Matrix. It specifies the Data Matrix symbology characteristics, data character encodation, symbol formats, dimensions and print quality requirements, error correction rules, decoding algorithm, and user-selectable application parameters.

2 Normative References

This specification incorporates provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed below. The latest edition of the publication referred to applies.

EN796	Bar Coding : Symbology Identifiers
EN1556	Bar Coding : Terminology
ANSI X3.182	Bar Code Print Quality - Guideline (Same as EN1635 - Bar Coding : Test Specifications for Bar Code Symbols)
ANSI X3.4	Coded Character Sets - 7-bit American National Standard Code for Information Interchange (7-bit ASCII) (equivalent to the US national version of ISO 646)
ISO/IEC 8859-1	Information Processing - 8-bit Single-byte Coded Graphic Character Sets - Part 1 (Latin Alphabet Number 1)
ECI Assignments Document- AIM International	

3 Definitions and Mathematical Symbols

3.1 Definitions

For the purposes of this specification, the following definitions in EN 1556 shall apply:

algorithm, application standard, ASCII, autodiscrimination, binary, bit, CCD, code page, code set, data character, data codeword, data region, data separator character, decode algorithm, decoder, error correction, finder pattern, human readable character, latch character, leading zeros, matrix symbology, modulo, numeric, omnidirectional, orientation pattern, overhead, pad character, pixel, quiet zone, reference decode algorithm, Reed-Solomon error correction, scanner, shift characters, structured append, symbol character, symbology, symbology identifier, X-dimension

The following definitions also apply to this specification. Although some of the terms below are defined in EN1556, the definitions which follow below are more appropriate for this specification.

3.1.1 Alignment Pattern

A unique pattern in larger ECC 200 Data Matrix symbols, made up of a solid line of contiguous dark cells abutting a line of alternating dark and light cells. The alignment patterns run horizontally and vertically within the symbols.

3.1.2 Codeword

A symbol character value. An intermediate level of coding between source data and the graphical encodation in the symbol.

3.1.3 Module

A single cell in a matrix symbology used to encode one bit of data. In Data Matrix the module is a square shape.

3.1.4 Convolutional Coding

An Error Checking and Correcting (ECC) algorithm that processes a set of input bits into a set of output bits that can recover from damage. The encoding process consists of breaking the input bits into blocks, then convolving each input block with the contents of a multi-stage shift register to produce protected output blocks. These encoders can be constructed in hardware using input and output switches, shift registers, and exclusive-or (XOR) gates.

3.1.5 Cyclic Redundancy Check (CRC)

An error detection algorithm which performs binary modulo 2 long division of a binary input stream by a fixed binary bit stream and produces as output the remainder of this division. The fixed binary bit stream is usually represented by a polynomial. The remainder of the division is the CRC value. The division can be implemented in hardware by shift registers and XOR gates. The purpose of a CRC is to provide error detection allowing validation of the user data after correction by the ECC process.

3.1.6 Extended Channel Interpretation (ECI)

A protocol used by some symbologies that allows the output data stream to have interpretations other than that of the default character set.

3.1.7 Pattern Randomizing

A procedure which converts an original bit pattern to another bit pattern by inverting selected bits. The resulting bit stream is less likely to have repeating patterns.

3.2 Mathematical Symbols and Operation

For the purposes of this specification, the mathematical symbols which follow shall apply globally unless defined locally:

d number of error correction codewords
e number of erasures
k (for ECC 000 - 140) the number of bits in a complete segment input to the state machine to generate the convolutional code
(for ECC 200) total number of error correction codewords
m the memory order of the convolutional code
n (for ECC 000 - 140) the number of bits in a complete segment generated by the state machine producing the convolutional code
(for ECC 200) total number of data codewords
N the numerical base in a encodation scheme
p number of misdecode protection codewords
S symbol character
t number of errors
u the input bit segment to the state machine, taken k bits at a time
v the output bit segment from the state machine, generated n bits at a time
X horizontal and vertical width of a module
ϵ error correction codeword

For the purposes of this specification, the notations and mathematical operations which follow shall apply:

div is the integer division operator
mod is the integer remainder after division
XOR is the exclusive-or logic function whose output is one only when its two inputs are not equivalent.
LSB Least significant bit
MSB Most significant bit

4 Symbol Description

4.1 Basic Characteristics

Data Matrix is a two-dimensional matrix symbology. There are two types: ECC 000 - 140 with several available levels of convolutional error correction and ECC 200 which uses Reed-Solomon error correction. For new applications ECC 200 is recommended. ECC 000 - 140 should only be used in closed applications where a single party controls both the production and reading of the symbols and is responsible for overall system performance. The characteristics of Data Matrix are:

a. Encodable character set:

1. values 0 - 127 in accordance with ANSI X3.4, i.e. all 128 ASCII characters

(equivalent to the U.S. national version of ISO 646)
2. values 128 - 255 in accordance with ISO 8859-1; Latin Alphabet No. 1. These are referred to as extended ASCII

b. Representation of data: A dark module is a binary one and a light module is a zero.

c. Symbol size in modules (not including quiet zone):

ECC 000 - 140	ECC 200
9 by 9	10 by 10
to 49 by 49	to 144 by 144
Odd Only	Even Only

d. Data characters per symbol (for maximum symbol size in ECC200):

1. Alphanumeric data: up to 2335 characters
2. 8-bit byte data: 1556 characters
3. Numeric data: 3116 digits

e. Selectable error correction:
ECC 000 - 140: Four levels of convolutional error correction, plus the option to apply only error detection
ECC 200: Reed-Solomon error correction

f. Code type: Matrix

g. Orientation independence: Yes

4.2 Summary of Additional Features

The following summarises additional features which are inherent or optional in Data Matrix:

a. Reflectance reversal: (Inherent) Symbols are intended to be read when marked so that the image is either dark on light or light on dark (see Figure 1).

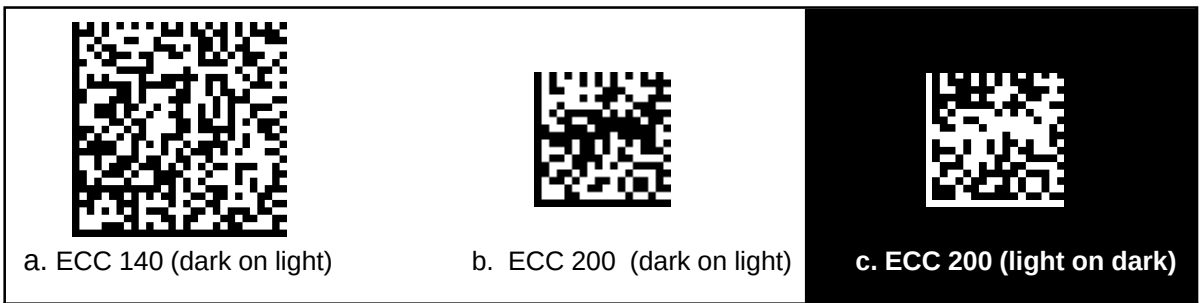
b. Extended Channel Interpretations: (ECC 200 only, optional) This mechanism enables characters from other character sets (e.g. Arabic, Cyrillic, Greek, Hebrew) and other data interpretations or industry-specific requirements to be represented.

c. Rectangular symbols: (ECC 200 only, optional) Six symbol formats are specified in a rectangular form.

d. Structured append: (ECC 200 only, optional) This allows files of data to be represented in up to 16 Data Matrix symbols. The original data can be correctly reconstructed regardless of the order in which the symbols are scanned.

4.3 Symbol Structure

Each Data Matrix symbol consists of data regions which contain nominally square modules set out in a regular array. In larger ECC 200 symbols, data regions are separated by alignment patterns. The data region is surrounded by a finder pattern, and this shall be surrounded on all four sides by a quiet zone border. Figure 1 illustrates an ECC 140 and two representations of an ECC 200 symbol.



a. ECC 140 (dark on light) b. ECC 200 (dark on light) c. ECC 200 (light on dark)

Figure 1: ECC 140 (a) and ECC 200 (b & c) encoding "A1B2C3D4E5F6G7H8I9J0K1L2"

4.3.1 Finder Pattern

The finder pattern is a perimeter to the data region and is one module wide. Two adjacent sides, the left and lower sides, forming the L boundary, are solid dark lines; these are used primarily to determine physical size, orientation and symbol distortion. The two opposite sides are made up of alternating dark and light modules. These are used primarily to define the cell structure of the symbol, but also can assist in determining physical size and distortion.

4.3.2 Symbol Sizes and Capacities

ECC 000 - 140 symbols have an odd number of rows and an odd number of columns. Symbols are square with sizes from 9 x 9 to 49 x 49 (modules) not including quiet zones. These symbols can be recognized by the upper right corner module being dark. The complete attributes of ECC 000 - 140 symbols are given in Annexe A.

ECC 200 symbols have an even number of rows and an even number of columns. Some symbols are square with sizes from 10 x 10 to 144 x 144 not including quiet zone. Some symbols are rectangular with sizes from 8 x 18 to 16 x 48 not including quiet zone. All ECC 200 symbols can be recognized by the upper right corner module being light. The complete attributes of ECC 200 symbols are given in Table 11 in Section 6.5.

5 ECC 000 - 140 Requirements

For new applications or open systems ECC 200 is recommended (See Section 6).

5.1 Encode Procedure Overview

This section provides an overview of the encoding procedure. Following sections will provide more details. An example encode for ECC 050 is given in Annexe Q. The following steps convert user data to an ECC 000 - 140 symbol:

Step 1: Data Encodation
The user data is analyzed to identify the variety of different characters to be encoded. For maximum compaction efficiency, the lowest level encodation scheme capable of encoding the data should be selected. If the user does not specify the matrix size, then choose the smallest size that accommodates the data. The result of this step is called the Encoded Data Bit Stream.

Step 2: Data Prefix Construction
A Data Prefix Bit Stream is constructed from the Format ID, CRC, and Data Length bit fields. This Data Prefix Bit Stream is prefixed to the Encoded Data Bit Stream to produce the Unprotected Bit Stream.

Step 3: Error Checking and Correcting
The Unprotected Bit Stream is processed by the user specified convolutional coding encode algorithm to produce the Protected Bit Stream. This step is omitted for ECC 000.

Step 4: Header and Trailer Construction
A header containing only the ECC bit field is prefixed to the Protected Bit Stream. A trailer containing pad bits (zeros) is appended to the Protected Bit Stream. The Protected Bit Stream with the header and trailer added is called the Unrandomized Bit Stream.

Step 5: Pattern Randomizing
The Unrandomized Bit Stream is processed by the pattern randomizing algorithm and produces the Randomized Bit Stream.

Encodation Scheme	Characters	Bits per Data Character
Base 11	Numeric data	3.5
Base 27	Upper-case alphabetic	4.8
Base 37	Upper-case alphanumeric	5.25
Base 41	Upper-case alphanumeric and punctuation	5.5
ASCII	Full 128 ASCII set	7
8-bit Byte	User defined	8

Table 1: Encodation Schemes

Step 6: Module Placement in Matrix

Modules are placed in a matrix to construct the finder pattern. The Randomized Bit Stream is placed into the matrix one module at a time according to the data module placement algorithm given in Annexe B.

Figure 2 shows the various bit streams during the encode process.

5.2 Data Encodation

The data shall be encoded using one of six encodation schemes (see Table 1).

The encodation scheme is fixed for the entire symbol, and thus the selection of the most appropriate encodation scheme can have a considerable effect on the number of bits required to encode any given data.

The same data may be represented in ECC 000 - 140 symbols in different ways through the use of the different encodation schemes. The character sets of all the encodation schemes, except the 8-bit byte scheme, are given in Annexe C. The 8-bit byte scheme is user definable.

The most efficient scheme to use is the lowest base number scheme which is capable of encoding all the characters in the message. Thus if all the characters can be encoded in Base 27, it is not efficient to use Base 37, Base 41 or ASCII.

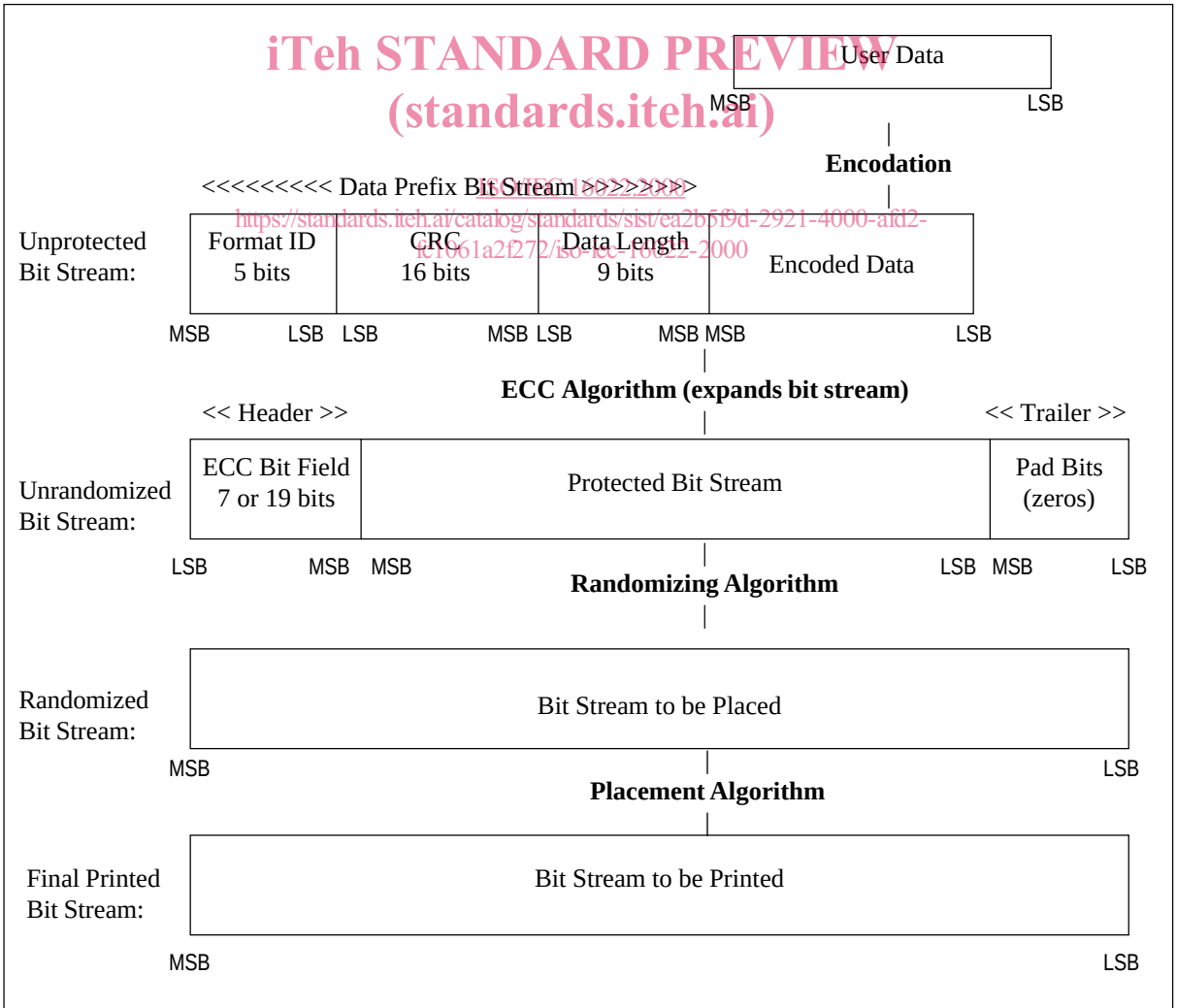


Figure 2: ECC 000-140 Encode Process Bit Streams

To determine the appropriate encodation scheme, the data to be encoded should be analyzed. The character sets of each of the Base N encodation schemes should be compared with the data character set to be encoded starting with the Base 11 character set. If this is suitable then it should be used, if not the comparisons should continue with Base 27, Base 37 and Base 41 until the appropriate lowest level encodation scheme is found. If data characters beyond the capability of Base 41 need to be encoded, the ASCII set should be used, unless characters are beyond this; in which case the 8-bit byte set should be used.

For all encodation schemes, each compressed sequence of 4-24 bits is placed into the Encoded Bit Stream in reverse order (LSB first). This means that each individual compressed sequence is composed, then reversed, and output immediately to the Encoded Bit Stream. This does not mean that a complete compressed bit stream is formed, then reversed.

The details of each encodation scheme are given in the following clauses.

5.2.1 Base 11 - Numeric Encodation

The Base 11 (Numeric) Encodation scheme encodes 6 data characters as 21 bits, achieving a encodation of 3.5 bits per data character. The Base 11 code set enables the following 11 characters to be encoded:

0 to 9
space

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 11 code values, as given in Annexe C.

In the second phase, the Base 11 code values shall be compacted using a Base 11 to Base 2 conversion according to the procedures defined in Annexe C.1.

5.2.2 Base 27 - Upper-case Alphabetic Encodation

The Base 27 (Upper-case Alphabetic) Encodation scheme encodes 5 data characters as 24 bits, achieving a encodation of 4.8 bits per data character. The Base 27 code set enables the following 27 characters to be encoded:

A to Z
space

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 27 code values, as given in Annexe C.

In the second stage, the Base 27 code values shall be compacted using a Base 27 to Base 2 conversion according to the procedures defined in Annexe C.2.

5.2.3 Base 37 - Upper-case Alphanumeric Encodation

The Base 37 (Upper-case Alphanumeric) Encodation scheme encodes 4 data characters as 21 bits, achieving a encodation of 5.25 bits per data character. The Base 37 code set enables the following 37 characters to be encoded:

A to Z
0 to 9
space

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 37 code values, as given in Annexe C.

In the second stage, the Base 37 code values shall be compacted using a Base 37 to Base 2 conversion according to the procedures defined in Annexe C.3.

5.2.4 Base 41 - Upper-case Alphanumeric plus Punctuation Encodation

The Base 41 (Upper-case Alphanumeric plus Punctuation) Encodation scheme encodes 4 data characters as 22 bits, achieving a encodation of 5.5 bits per data character. The Base 41 code set enables the following 41 characters to be encoded:

A to Z
0 to 9
space
. (period)
, (comma)
- (minus or hyphen)
/ (forward slash or solidus)

The data is encoded in two stages. In the first stage, the actual data characters shall be replaced by their Base 41 code values, as given in Annexe C.

In the second stage, the Base 41 code values shall be compacted using a Base 41 to Base 2 conversion according to the procedures defined in Annexe C.4.

5.2.5 ASCII Encodation

The ASCII Encodation scheme enables all 128 characters from ANSI X3.4 to be encoded. Each data character shall be encoded as a 7-bit byte equivalent to the decimal value shown in the ASCII column of Table C1 of Annexe C.

5.2.6 8-bit Byte Encodation

The 8-bit Byte Encodation scheme shall be used for closed applications, where the data interpretation shall be determined by the user. Each data character shall be encoded as an 8-bit byte.

5.3 User Selection of Error Correction Level

5.3.1 Selection of Error Correction Level

ECC 000 - 140 symbols offer five levels of error correction using convolutional code error correction, as set out in Table 2. In an application, it is important to understand that these error correction levels result in the generation of a proportional increase in the number of bits in the message (and hence increase in the size of the symbol), and offer different levels of error recovery.

Error Correction Code Level	Maximum % Damage	Increase in User Bits from ECC 0
000	none	none
050	28	33
080	5.5	50
100	12.6	100
140	25	300

Table 2: Error Correction Error Recovery and Overhead Percentages

5.3.2 Additional Error Correction Levels Based on Convolutional Codes

Additional levels of error correction, based on convolutional code algorithms, are possible in Data Matrix. Symbols conforming to these constructions may be found in systems implemented prior to the publication of this specification. Information on these additional Error Correction levels is available from AIM USA.

5.4 Constructing the Unprotected Bit Stream

Figure 2 illustrates that the Unprotected Bit Stream has the Data Prefix Bit Stream as a prefix to the encoded data bits. The component parts of the Data Prefix Bit Stream are defined below.

5.4.1 Format ID Bit Field

The format ID defines the data encodation scheme. The format ID has a decimal value for the purposes of definition and a 5-bit segment value for encoding as defined in Table 3.

Format ID	Encodation Scheme	Binary Segment Value MSB LSB
1	Base 11	00000
2	Base 27	00001
3	Base 41	00010
4	Base 37	00011
5	ASCII	00100
6	8-bit Byte	00101

Table 3: Encoding the Format ID

5.4.2 CRC Bit Field

The CRC Bit Field is generated by the CRC algorithm. The CRC Value is generated from the original user data as 8-bit bytes before encodation and so produces an independent error check on the user data. Annexe D describes the complete procedure for generating the CRC Value.

5.4.3 Data Length Bit Field

The Data Length Bit Field is 9 bits in length and represents, as a binary value, the number of user data characters being encoded.

5.4.4 Data Prefix Construction

The Data Prefix Bit Stream is constructed as 30 bits as illustrated in Figure 3.

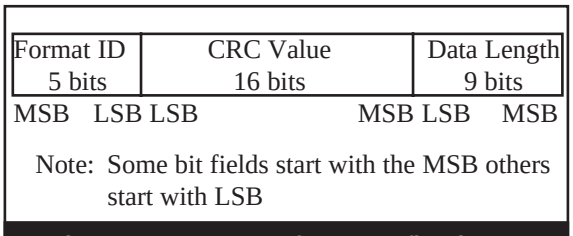


Figure 3: Structure of Data Prefix Bit Stream

5.4.5 Completing the Unprotected Bit Stream

The encoded data bits are added as a suffix to the Data Prefix Bit Stream to construct the Unprotected Bit Stream.

5.5 Constructing the Unrandomized Bit Stream

Figure 2 illustrates that the Unrandomized Bit