INTERNATIONAL STANDARD

ISO/IEC 21471

First edition
2020-02

Information technology — Automatic identification and data capture techniques — Extended rectangular data matrix (DMRE) bar code symbology specification

Technologies de l'information – Techniques de l'identification et de saisie de données automatiques – Data Matrix Rectangulaire Etendu (DMRE) spécification de symbologie de code à barres

Reference number
ISO/IEC 21471:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

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.

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

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.

Introduction

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

This document is an extension of ISO/IEC 16022, to which it adds rectangular formats. Maximum compatibility is a design goal. In consequence, most clauses of ISO/IEC 16022 are identical to those of this document, including the module placement algorithm and the reference decode algorithm.

This document is published separately because existing equipment supporting ISO/IEC 16022 will not recognize DMRE symbols. Only equipment that is enabled and configured to support DMRE will be capable of printing and scanning the new rectangular formats. To avoid user confusion due to this fact, a separate and complete document was developed.

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 standardized symbology specifications is designed to achieve this.

Information technology — Automatic identification and data capture techniques — Extended rectangular data matrix (DMRE) bar code symbology specification

1 Scope

This document defines the requirements for the symbology known as extended rectangular data matrix (DMRE). It specifies the DMRE code symbology characteristics, data character encodation, symbol formats, dimensions and print quality requirements, error correction rules, decoding algorithm, and user-selectable application parameters.

It applies to all DMRE code symbols produced by any printing or marking technology.

Original data matrix code sizes are not covered by this document but defined in ISO/IEC 16022 using the same matrix placement, decoding and error correction algorithm.

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 646, *Information technology — ISO 7-bit coded character set for information interchange*

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

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

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

ISO/IEC 29158:—[1], *Information technology — Automatic identification and data capture techniques — Direct Part Mark (DPM) Quality Guideline*

3 Terms, definitions, symbols and abbreviated terms and mathematical/logical notations

3.1 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 terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

1) Under preparation. Stage at the time of publication: ISO/IEC/DIS 29158:2020.

3.1.1
codeword
symbol character value
intermediate level of coding between source data and the graphical encodation in the symbol

3.1.2
module
single cell in a matrix symbology used to encode one bit of data

Note 1 to entry: In DMRE, the module is nominally a square shape.

3.1.3
pattern randomising
procedure which converts an original bit pattern to another bit pattern by inverting selected bits

Note 1 to entry: The resulting bitstream is less likely to have repeating patterns.

3.2 Symbols and abbreviated terms

d number of error correction codewords

e number of erasures

N the numerical base in an encodation scheme

p number of codewords reserved for error detection

S symbol character

t number of errors

X horizontal and vertical width of a module

ε error correction codeword

DMRE extended rectangular data matrix

ECI extended channel interpretation

DPM direct part marking

3.3 Mathematical/logical notations

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

LSB least significant bit

MSB most significant bit

4 Symbol description

4.1 Basic characteristics

DMRE is a two-dimensional matrix symbology.

The characteristics of DMRE are:

a) Encodable character set:

   1) values 0 to 127 in accordance with ISO/IEC 646, i.e. all 128 ASCII characters (equivalent to the U.S. national version of ISO 646);

   2) values 128 to 255 in accordance with ISO 8859-1. 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.

   This document specifies DMRE symbols in terms of dark modules marked on a light background. However, 4.2 provides that symbols may also be produced with the module's colours reversed. In such symbols, dark modules would be a binary zero, and light modules would be a binary 1.

c) Symbol size in modules (not including quiet zone): 8 × 48 to 26 × 64 even values only (see Table 7).

   Symbol sizes 8 × 18, 8 × 32, 12 × 26, 12 × 36, 16 × 36 and 16 × 48 are defined by ISO/IEC 16022 and are not covered by this document. These rectangular data matrix sizes are fully compatible with this document.

d) Data characters per symbol (for maximum symbol size):

   1) alphanumeric data: up to 175 characters;

   2) 8-bit byte data: 116 characters;

   3) numeric data: 236 digits.

e) Code type: rectangular matrix.

f) Orientation independence: yes.

g) Error detection and correction: ECC 200 Reed Solomon. No support for ECC 000 to ECC 140.

4.2 Summary of additional features

The following summarizes additional features which are inherent or optional in DMRE:

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). The specifications in this document are based on dark images on a light background, therefore references to dark or light modules should be taken as references to light or dark modules respectively in the case of symbols produced with reflectance reversal.

b) Extended channel interpretations (ECI) (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) Structured append (optional): this allows files of data to be represented in up to 16 rectangular data matrix symbols. The original data can be correctly reconstructed regardless of the order in which the symbols are scanned.

4.3 Symbol structure

4.3.1 General

Each DMRE symbol consists of data regions which contain nominally square modules set out in a regular array. In larger symbols, data regions are separated by alignment patterns (as illustrated in Figures C.1 and C.2). 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 two representations of a rectangular data matrix symbol.

a) DMRE, dark on light

b) DMRE, light on dark

Figure 1 — DMRE "A1B2C3D4E5F6G7H8I9J0K1L2"

4.3.2 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. The extent of the quiet zone is indicated by the corner marks in Figure 1.

4.3.3 Symbol sizes and capacities

DMRE code symbols have an even number of rows and an even number of columns. The symbols are rectangular with sizes from 8 × 48 to 26 × 64 not including quiet zones. For all rectangular data matrix code symbols the upper right corner module is light in the dark on light version (see Figure 1). The complete attributes are given in Table 7.

5 DMRE code requirements

5.1 Encoding procedure overview

This subclause provides an overview of the encoding procedure. The following subclauses provide more details. An encoding example for DMRE code is given in Annex H. The following steps convert user data to a DMRE code symbol:

Step 1: Data encodation

As DMRE code includes various encodation schemes that allows a defined set of characters to be converted into codewords more efficiently than the default scheme, analyse the data stream to identify the variety of different characters to be encoded. Insert additional codewords to switch between the encodation schemes and to perform other functions. Add pad characters as needed to fill the required number of codewords. If the user does not specify the matrix size, then choose the smallest size that accommodates the data. A complete list of matrix sizes is shown in Table 7.

Table 1 — Encodation schemes for rectangular data matrix code

Encodation scheme	Characters	Bits per data character
ASCII	Double digit numerics	4
	ASCII values 0 to 127	8
	Extended ASCII values 128 to 255	16

[a] Encoded as two C40 values as a result of the use of a shift character.

[b] Encoded as two Text values as a result of the use of a shift character.

Table 1 *(continued)*

Encodation scheme	Characters	Bits per data character
C40	Upper-case alphanumeric	5,33
	Lower case and special characters	10,66[a]
Text	Lower-case alphanumeric	5,33
	Upper case and special characters	10,66[b]
X12	ANSI X12 EDI data set	5,33
EDIFACT	ASCII values 32 to 94	6
Base 256	All byte values 0 to 255	8

[a] Encoded as two C40 values as a result of the use of a shift character.

[b] Encoded as two Text values as a result of the use of a shift character.

Step 2: Error checking and correcting codeword generation

Generate the error correction codewords for the result codeword stream from above step. The result of this process expands the codeword stream by the number of error correction codewords. Place the error correction codewords after the data codewords.

Step 3: Module placement in matrix

Place the codeword modules in the matrix. Insert the alignment pattern modules, if any, in the matrix. Add the finder pattern modules around the matrix.

NOTE Table 1 of this document is identical to ISO/IEC 16022, Table 1.

5.2 Data encoding

5.2.1 Overview

The data may be encoded using any combination of six encodation schemes (see Table 1). ASCII encodation is the basic scheme. All other encodation schemes are invoked from ASCII encodation and return to this scheme. The compaction efficiencies given in Table 1 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 switching between encodation schemes and between code sets within an encodation scheme (see Annex I). It should also be noted that even if the number of codewords is minimized, the codeword stream sometimes needs to be expanded to fill a symbol. This fill process is done using pad characters.

5.2.2 Default character interpretation

The default character interpretation for character values 0 to 127 shall conform to ISO/IEC 646. The default character interpretation for character values 128 to 255 shall conform to ISO 8859-1. The graphical representation of data characters shown throughout this document complies with the default interpretation. This interpretation can be changed using extended channel interpretation (ECI) escape sequences, see 5.4. The default interpretation corresponds to ECI 000003.

5.2.3 ASCII encodation

ASCII encodation is the default encodation scheme for the first symbol character in all symbol sizes. It encodes ASCII data, double density numeric data and symbology control characters. Symbology control characters include function characters, the pad character and the switches to other code sets. ASCII data is encoded as codewords 1 to 128 (ASCII value plus 1). Extended ASCII (data values 128 to 255) is encoded using the upper shift symbology control character (see 5.2.4.3). The digit pairs 00 to 99 are encoded with codewords 130 to 229 (numeric value plus 130). The ASCII code assignments are shown in Table 2.

Table 2 — ASCII encodation values

Codeword	Data or function
1 to 128	ASCII data (ASCII value + 1)
129	Pad (see 5.2.4.4)
130 to 229	2-digit data 00 to 99 (numeric value + 130)
230	Latch to C40 encodation
231	Latch to Base 256 encodation
232	FNC1
233	Structured append
234	Reader programming
235	Upper shift (shift to Extended ASCII)
236	05 Macro
237	06 Macro
238	Latch to ANSI X12 encodation
239	Latch to Text encodation
240	Latch to EDIFACT encodation
241	ECI character
242 to 255	Not to be used in ASCII encodation

5.2.4 Symbology control characters

5.2.4.1 General

DMRE code symbols have several special symbology control characters, which have particular significance to the encodation scheme. These characters shall be used to instruct the decoder to perform certain functions or to send specific data to the host computer as described in 5.2.4.2 to 5.2.4.10. These symbology control characters, with the exception of values from 242 through 255, are found in the ASCII encodation set (see Table 2).

5.2.4.2 Latch characters

A Latch character shall be used to switch from ASCII encodation to one of the other encodation schemes. All codewords which follow a Latch character shall be compacted according to the new encodation scheme. The encodation schemes have different methods for returning to the ASCII encodation set.

5.2.4.3 Upper shift character

The Upper shift character is used in combination with an ASCII value (1 to 128) to encode an extended ASCII character (129 to 255). An extended ASCII character encoded in the ASCII, C40, or Text encodation scheme requires a preceding Upper shift character and the extended ASCII character value decreased by 128 is then encoded according to the rules of the encodation scheme. In ASCII encodation, the Upper shift character is represented by codeword 235. The reduced data value (i.e. ASCII value minus 128) is transformed into its codeword value by adding 1. For example, to encode ¥ (Yen currency symbol) (ASCII value 165), an Upper shift character (codeword 235) is followed by value 37 (165 - 128), which is encoded as codeword 38. If there are long data strings of characters from the extended ASCII range, a Latch to Base 256 encodation should be more efficient.

5.2.4.4 Pad character

If the encoded data, irrespective of the encodation scheme in force, does not fill the data capacity of the symbol, pad characters (value 129 in ASCII encodation) shall be added to fill the remaining data capacity of the symbol. The pad characters shall only be used for this purpose. Before inserting pad characters, it is necessary to return to ASCII encodation if in any other encodation mode. The 253-State pattern randomising algorithm shall be applied to the pad characters starting at the second pad character and continuing to the end of the symbol (see A.2).

5.2.4.5 ECI character

An ECI character[9] is used to change from the default interpretation used to encode data. The ECI protocol is common across a number of symbologies and its application to rectangular data matrix code is defined in more detail in 5.4. The ECI character shall be followed by one, two, or three codewords which identify the ECI being invoked. The new ECI remains in place until the end of the encoded data, or until another ECI character is used to invoke another interpretation.

5.2.4.6 Shift characters in C40 and Text encodation

In C40 and Text encodation, three special characters, called shift characters, are used as a prefix to one of 40 values to encode about three quarters of the ASCII characters. This allows the remaining ASCII characters to be encoded in a more condensed way with single values.

5.2.4.7 FNC1 alternate data type identifier

To encode data to conform to specific industry standards as authorized by AIM Inc., a FNC1 character shall appear in the first or second symbol character position (or in the fifth or sixth data positions of the first symbol of Structured Append). FNC1 encoded in any other position is used as a field separator and shall be transmitted as $^{G}{}_{S}$ control character (ASCII value 29).

5.2.4.8 Macro characters

DMRE provides a means of abbreviating an industry specific header and trailer in one symbol character. This feature exists to reduce the number of symbol characters needed to encode data in a symbol using certain structured formats. A Macro character shall be in the first character position of a symbol. They shall not be used in conjunction with Structured append and their functions are defined in Table 3. The header shall be transmitted as a prefix to the data stream and the trailer shall be transmitted as a suffix to the data stream. The symbology identifier, if used, shall precede the header.

Table 3 — Macro functions

Macro codeword	Name	Interpretation	
		Header	Trailer
236	05 Macro	[)>$^{R}{}_{S}$05$^{G}{}_{S}$	R_S E_{oT}
237	06 Macro	[)>$^{R}{}_{S}$06$^{G}{}_{S}$	R_S E_{oT}

5.2.4.9 Structured append character

A Structured append character is used to indicate that the symbol is part of a Structured append sequence according to the rules defined in 5.6.

5.2.4.10 Reader programming character

A Reader programming character indicates that the symbol encodes a message used to program the reader system. The Reader programming character shall appear as the first codeword of the symbol and Reader programming shall not be used with Structured append.

5.2.5 C40 encodation

5.2.5.1 General

The C40 encodation scheme is designed to optimize the encoding of upper-case alphabetic and numeric characters but also enables other characters to be encoded by the use of shift characters in conjunction with the data character.

C40 characters are partitioned into 4 subsets. Characters of the first set, called the basic set, are the three special shift characters, the space character, and the ASCII characters A-Z and 0-9. They are assigned to a single C40 value. Characters of the other sets are assigned to one of the three shift characters, pointing to one of the 3 remaining subsets, followed by one of the C40 values (use Table B.1).

As a first stage, each data character is converted into a single C40 value or a pair of C40 values. The complete string of C40 values is then decomposed into groups of three values (special rules apply if one or two values remain at the end, see 5.2.5.3). Each triplet (C1, C2, C3) is then encoded into a 16-bit value according to the formula: (1600*C1) + (40*C2) + C3 + 1. Each 16-bit value is then separated into 2 codewords by taking the most significant 8 bits and the least significant 8 bits.

5.2.5.2 Switching to and from C40 encodation

It is possible to switch to C40 encodation from ASCII encodation using the appropriate latch codeword (230). Codeword 254 immediately following a pair of codewords in C40 encodation acts as an Unlatch codeword to switch back to ASCII encodation. Otherwise, the C40 encodation remains in effect to the end of the data encoded in the symbol.

5.2.5.3 C40 encodation rules

Each pair of codewords represents a 16-bit value where the first codeword represents the most significant 8 bits. Three C40 values (C1, C2, C3) shall be encoded as:

$$(1600 * C1) + (40 * C2) + C3 + 1$$

which produces a value from 1 to 64000. Figure 2 illustrates three C40 values compacted into two codewords. Characters in the Shift 1, Shift 2 and Shift 3 sets shall be encoded by first encoding the appropriate shift character, and then the C40 value for the data. C40 encodation may be in effect at the end of the symbol's codewords which encode data.

The following rules apply when only one or two symbol characters remain in the symbol before the start of the error correction codewords:

a) If two symbol characters remain and three C40 values remain to be encoded (which may include both data and shift characters), encode the three C40 values in the last two symbol characters. A final Unlatch codeword is not required.

b) If two symbol characters remain and two C40 values remain to be encoded (the first C40 value may be a shift or data character but the second shall represent a data character), encode the two remaining C40 values followed by a pad C40 value of 0 (Shift 1) in the last two symbol characters. A final Unlatch codeword is not required.

c) If two symbol characters remain and only one C40 value (data character) remains to be encoded, the first symbol character is encoded as an Unlatch character and the last symbol character is encoded with the data character using the ASCII encodation scheme.

d) If one symbol character remains and one C40 value (data character) remains to be encoded, the last symbol character is encoded with the data character using the ASCII encodation scheme. The Unlatch character is not encoded, but is assumed, before the last symbol character.

In all other cases, either an Unlatch character is used to exit the C40 encodation scheme before the end of the symbol, or a larger symbol size is required to encode the data.

Data characters	AIM
C40 values	14, 22, 26
Calculate 16-bit value	(1600 * 14) + (40 * 22) + 26 + 1 = 23307
1st codeword: (16-bit value) div 256	23307 div 256 = 91
2nd codeword: (16-bit value) mod 256	23307 mod 256 = 11
Codewords	91, 11

Figure 2 — Example of C40 encoding

5.2.5.4 Use of Upper shift with C40

In C40 encodation, the Upper shift character is not a symbology function character but a shift within the encodation set. When a data character from the extended ASCII character range is encountered, three or four values in C40 encodation need to be encoded according to the following rule:

IF [ASCII value – 128] is in the Basic Set, then:

[1(Shift 2)] [30(Upper Shift)] [V(ASCII value – 128)]

ELSE

[1(Shift 2)] [30(Upper Shift)] [0, 1, or 2(Shift 1, 2, or 3)] [V(ASCII value – 128)]

In the rule the number in [ ] equates to the C40 values from Table B.1; V has been used to indicate the appropriate C40 value.

5.2.6 Text encodation

5.2.6.1 General

Text encodation is designed to encode normal printed text, which is predominantly lowercase characters. It is similar in structure to the C40 encodation set, except that lower-case alphabetic characters are directly encoded (i.e. without using a shift). Upper-case alphabetic characters are preceded by a Shift 3. The full Text encodation character set assignments are shown in Table B.2.

5.2.6.2 Switching to and from Text encodation

It is possible to switch to Text encodation from ASCII encodation using the appropriate latch codeword (239). Codeword 254 immediately following a pair of codewords in text encodation acts as an Unlatch codeword to switch back to ASCII encodation. Otherwise, the Text encodation remains in effect to the end of the data encoded in the symbol.

5.2.6.3 Text encodation rules

The rules for C40 encodation apply.

5.2.7 ANSI X12 encodation

5.2.7.1 General

ANSI X12 encodation is used to encode the standard ANSI X12 electronic data interchange characters, which are compacted three data characters to two codewords in a manner similar to C40 encodation. It encodes upper-case alphabetic characters, numerics, space and the three standard ANSI X12 terminator