МЕЖГОСУДАРСТВЕННЫЙ СОВЕТ ПО СТАНДАРТИЗАЦИИ, МЕТРОЛОГИИ И СЕРТИФИКАЦИИ
(МГС)
INTERSTATE COUNCIL FOR STANDARDIZATION, METROLOGY AND CERTIFICATION
(ISC)

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

ГОСТ ISO/IEC 16022— 2025

Информационные технологии

ТЕХНОЛОГИИ АВТОМАТИЧЕСКОЙ ИДЕНТИФИКАЦИИ И СБОРА ДАННЫХ

Спецификация символики штрихового кода Data Matrix

(ISO/IEC 16022:2024, IDT)

Издание официальное

Москва
Российский институт стандартизации
2026

gost.e-ecolog.ru

Предисловие

Цели, основные принципы и общие правила проведения работ по межгосударственной стандартизации установлены ГОСТ 1.0 «Межгосударственная система стандартизации. Основные положения» и ГОСТ 1.2 «Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Правила разработки, принятия, обновления и отмены»

Сведения о стандарте

1 ПОДГОТОВЛЕН Ассоциацией автоматической идентификации «ЮНИСКАН/ГС1 РУС» (ГС1 РУС) (Российская Федерация) совместно с Государственным предприятием «Центр систем идентификации» НАН Беларуси (ГП ЦСИ) (Беларусь), GS1 Kyrgyzstan (Киргизия), ГС1 Туркменистан (Туркмения) и GS1 Uzbekistan (Узбекистан) в рамках Межгосударственного технического комитета МТК 517 «Технологии автоматической идентификации и сбора данных» на основе собственного перевода ГС1 РУС на русский язык англоязычной версии стандарта, указанного в пункте 5

2 ВНЕСЕН Федеральным агентством по техническому регулированию и метрологии

3 ПРИНЯТ Межгосударственным советом по стандартизации, метрологии и сертификации по результатам голосования в АИС МГС (протокол от 31 октября 2025 г. № 190-П)

За принятие проголосовали:

Краткое наименование страны по МК (ИСО 3166) 004—97	Код страны по МК (ИСО 3166) 004—97	Сокращенное наименование национального органа по стандартизации
Армения	AM	ЗАО «Национальный орган по стандартизации и метрологии» Республики Армения
Беларусь	BY	Госстандарт Республики Беларусь
Россия	RU	Росстандарт
Узбекистан	UZ	Узбекское агентство по техническому регулированию

4 Приказом Федерального агентства по техническому регулированию и метрологии от 16 декабря 2025 г. № 1731-ст межгосударственный стандарт ГОСТ ISO/IEC 16022—2025 введен в действие в качестве национального стандарта Российской Федерации с 1 октября 2026 г.

5 Настоящий стандарт идентичен международному стандарту ISO/IEC 16022:2024 «Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация символики штрихового кода Data Matrix» («Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification», IDT).

Международный стандарт разработан подкомитетом ISO/IEC JTC 1/SC 31 «Технологии автоматической идентификации и сбора данных» совместного технического комитета по стандартизации ISO/IEC JTC 1 «Информационные технологии» Международной организации по стандартизации (ISO) и Международной электротехнической комиссии (IEC).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им межгосударственные стандарты, сведения о которых приведены в дополнительном приложении ДА.

Дополнительные сноски по тексту стандарта, выделенные курсивом, приведены для пояснения текста оригинала

6 ВВЕДЕН ВПЕРВЫЕ

7 Некоторые положения международного стандарта могут быть объектами патентных прав. Международная организация по стандартизации (ISO) и Международная электротехническая комиссия (IEC) не несут ответственности за идентификацию подобных патентных прав

© ISO, 2024
© IEC, 2024
© Оформление. ФГБУ «Институт стандартизации», 2026



В Российской Федерации настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 1
4 Обозначения 2
5 Математические или логические обозначения 2
6 Описание символов 2
6.1 Основные параметры 2
6.2 Обзор дополнительных свойств 3
6.3 Структура символов 3
7 Требования к кодированию Data Matrix 4
7.1 Основные положения процедуры кодирования 4
7.2 Кодирование данных 5
7.3 Интерпретация в расширенном канале (ECI) 11
7.4 Атрибуты символа Data Matrix 13
7.5 Структурированное соединение 13
7.6 Обнаружение и исправление ошибок 16
7.7 Формирование символа 18
8 Размеры символов 18
9 Качество печати символов 19
9.1 Общие положения 19
9.2 Параметры качества символа 19
9.3 Измерения в процессе контроля 19
10 Рекомендуемый алгоритм декодирования для символики Data Matrix 20
11 Рекомендации для пользователей 30
11.1 Интерпретация для визуального чтения 30
11.2 Способность к автоматическому распознаванию 30
11.3 Системные подходы 30
12 Передаваемые данные 30
12.1 Общие положения 30
12.2 Протокол для знака FNC1 30
12.3 Протокол для знака FNC1 во второй позиции 30
12.4 Протокол для знаков Macro в первой позиции 31
12.5 Протокол для интерпретаций ECI 31
12.6 Идентификатор символики 31
12.7 Пример передаваемых данных 31
Приложение А (обязательное) Процесс чередования в символике Data Matrix 33
Приложение В (обязательное) Шаблонная рандомизация в символике Data Matrix 37
Приложение С (обязательное) Наборы кодируемых знаков символики Data Matrix 38
Приложение D (обязательное) Направляющие шаблоны символов Data Matrix 42
Приложение Е (обязательное) Алгоритм обнаружения и исправления ошибок Рида—Соломона для символики Data Matrix 44
Приложение F (обязательное) Размещение знаков символа 47
Приложение G (обязательное) Качество печати символов Data Matrix — аспекты, связанные с особенностями символики 62
Приложение Н (обязательное) Идентификатор символики 71
Приложение I (справочное) Пример кодирования 72
Приложение J (рекомендуемое) Кодирование данных с использованием минимального числа знаков символа 74
Приложение К (рекомендуемое) Возможность автоматического распознавания 77
Приложение L (рекомендуемое) Системный подход 78
Приложение М (рекомендуемое) Рекомендации пользователям 79
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов межгосударственным стандартам 80
Библиография 81

Введение

Data Matrix — двухмерная матричная символика, состоящая из модулей номинально квадратной формы, упорядоченных внутри периметра шаблона поиска. В настоящем стандарте представление символа и его описание приведено, главным образом, для темного символа на светлом фоне. Тем не менее, символы Data Matrix также могут быть напечатаны в виде светлых модулей на темном фоне.

Изготовителям оборудования для штрихового кодирования и пользователям указанной технологии необходима общедоступная стандартная спецификация символики, на которую они могли бы ссылаться при разработке оборудования и стандартов по применению. С этой целью разработан настоящий стандарт.

М Е Ж Г О С У Д А Р С Т В Е Н Н Ы Й С Т А Н Д А Р Т

Информационные технологии

ТЕХНОЛОГИИ АВТОМАТИЧЕСКОЙ ИДЕНТИФИКАЦИИ И СБОРА ДАННЫХ

Спецификация символики штрихового кода Data Matrix

Information technology. Automatic identification and data capture techniques.
Data Matrix bar code symbology specification

Дата введения — 2026—10—01

1 Область применения

Настоящий стандарт устанавливает требования к символике Data Matrix, а также параметры символики, кодирование знаков данных, форматы символов, требования к размерам и качеству печати, правила исправления ошибок, алгоритм декодирования и прикладные параметры, выбираемые пользователем.

Настоящий стандарт распространяется на все символы символики Data Matrix, напечатанные или нанесенные каким-либо другим способом.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты [для датированных ссылок применяют только указанное издание ссылочного стандарта, для недатированных — последнее издание (включая все изменения)]:

ISO/IEC 19762, Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary [Информационные технологии. Технологии автоматической идентификации и сбора данных (АИСД). Гармонизированный словарь]

ISO/IEC 15415, Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Two-dimensional symbols (Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация испытаний качества печати символов штрихового кода. Двумерные символы)

ISO/IEC 646, Information technology — ISO 7-bit coded character set for information interchange (Информационные технологии. Набор 7-битовых кодированных знаков ИСО для обмена информацией)

ISO/IEC 8859-1, Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet № 1 (Информационные технологии. Наборы 8-битовых однобайтных кодированных графических знаков. Часть 1. Латинский алфавит № 1)

ISO/IEC 29158, Information technology — Automatic identification and data capture techniques — Direct Part Mark (DPM) Quality Guideline [Информационные технологии. Технологии автоматической идентификации и сбора данных. Рекомендации по контролю качества маркировки при прямом маркировании изделий (ПМИ)]

3 Термины и определения

В настоящем стандарте применены термины и определения по ISO/IEC 19762, а также следующие термины с соответствующими определениями.

Издание официальное

ISO и IEC поддерживают терминологические базы данных для использования в стандартизации по следующим адресам:

- платформа онлайн-просмотра ISO: доступна по адресу gost.e-ecolog.ru ;
- Электропедия IEC: доступна по адресу gost.e-ecolog.ru .

3.1 **кодовое слово** (codeword); значение знака символа (symbol character value): Промежуточный уровень кодирования в процессе преобразования исходных данных в их графическое представление в символе.

3.2 **модуль** (module): Одиночная ячейка элемента в символе матричной символики, используемая для кодирования одного бита кодового слова (3.1).

3.3 **шаблонная рандомизация** (pattern randomising): Процедура, с помощью которой исходный набор битов превращают в другой набор битов путем преобразования выбранных битов.

Примечание 1 — Полученный битовый поток с меньшей вероятностью будет иметь повторяющиеся наборы.

4 Обозначения

e — число стираний;

k — общее число кодовых слов исправления ошибок;

n — общее число кодовых слов данных;

N — числовое основание в схеме кодирования;

p — число кодовых слов, зарезервированных для обнаружения ошибок;

S — знак символа;

t — число ошибок;

X — горизонтальный и вертикальный размеры модуля;

ε — кодовое слово исправления ошибок.

5 Математические или логические обозначения

div — оператор деления на целое число;

mod — остаток при делении на целое число;

XOR — исключающее-или (exclusive-or) — логическая функция или операция, результатом которой является единица только в случае неэквивалентности двух входов.

6 Описание символов

6.1 Основные параметры

Data Matrix представляет собой двумерную матричную символику.

Символика Data Matrix имеет следующие параметры:

a) кодируемый набор знаков:

1) знаки с десятичными значениями 0—127 в соответствии с набором знаков IRV (International Reference Version, международная ссылочная версия) по ISO/IEC 646, т. е. все 128 знаков набора ASCII;
2) знаки с десятичными значениями 128—255 в соответствии с набором знаков по ISO/IEC 8859-1 (далее в ссылках называемые знаками расширенного набора ASCII);
3) дополнительные знаки, которые могут быть закодированы с использованием возможностей интерпретации в расширенном канале (ECI);

b) представление данных: темный модуль соответствует двоичной единице, светлый — двоичному нулю.

Настоящий стандарт определяет символы Data Matrix как темные модули, расположенные на светлом фоне. Несмотря на это, в 6.2 предусматривается, что символы могут изготавливаться с модулями с обратными значениями коэффициентов отражения. В таких символах темные модули соответствовали бы двоичному нулю, а светлые модули — двоичной единице;

c) размер символа в модулях (без учета свободных зон), который находится в пределах от 10 × 10 модулей до 144 × 144 модуля для символов квадратной формы и от 8 × 18 модулей до 16 × 48 модулей для вариантов символов прямоугольной формы (см. таблицу 10).

Примечание — Дополнительные размеры для символов прямоугольной формы установлены ISO/IEC 21471 (см. [4]);

d) число знаков данных в символе (для символа максимального размера):
1) алфавитно-цифровые данные: до 2335 знаков,
2) данные в 8-битовых байтах: 1555 знаков,
3) числовые данные: 3116 числовых разрядов;
e) тип штрихового кода: матричный;
f) независимость от ориентации: присутствует;
g) обнаружение и исправление ошибок: по коду Рида—Соломона.

6.2 Обзор дополнительных свойств

Символика Data Matrix обладает следующими дополнительными, неотъемлемыми или устанавливаемыми по выбору свойствами:

a) замена на изображение с обратными значениями коэффициентов отражения (неотъемлемое свойство): темные символы на светлом фоне или светлые на темном фоне (см. рисунок 1). Положения настоящего стандарта установлены для темного изображения на светлом фоне, следовательно, указания о темных или светлых модулях должны рассматриваться как указания о светлых или темных модулях соответственно для символов, изготовленных с обратными значениями коэффициентов отражения изображения;

b) интерпретации в расширенном канале (ECI) (неотъемлемое свойство): данный механизм позволяет использовать знаки из иных наборов (например, знаки арабского, кириллического, греческого, еврейского алфавитов) и иных различных интерпретаций данных или представлять их в соответствии с особыми отраслевыми требованиями;

c) прямоугольная форма символов (свойство по выбору): установлены шесть форматов символа прямоугольной формы.

Примечание — Дополнительные форматы для символов прямоугольной формы приведены в ISO/IEC 21471 (см. [4]);

d) структурированное соединение: (свойство по выбору): позволяет представить файлы данных в виде до 16 символов Data Matrix. Исходные данные могут быть восстановлены вне зависимости от порядка сканирования символов. Если данное свойство не реализовано, устройство считывания не должно передавать данные при наличии символа структурированного соединения.

6.3 Структура символов

6.3.1 Общие положения

Каждый символ Data Matrix состоит из областей данных, составленных из номинально квадратных модулей, структурированных в регулярную матрицу. В символах большого размера области данных отделены направляющими шаблонами. Область данных или набор областей данных с направляющими шаблонами окружены шаблоном поиска, вокруг которого со всех четырех сторон должна присутствовать свободная зона. На рисунке 1 представлены два символа Data Matrix: темное изображение на светлом фоне и изображение с обратными значениями коэффициентов отражения.



a) Data Matrix — темное изображение на светлом фоне



b) Data Matrix — светлое изображение на темном фоне (изображение с обратными значениями коэффициентов отражения)

Рисунок 1 — Символ Data Matrix, кодирующий "A1B2C3D4E5F6G7H8I9J0K1L2"

6.3.2 Шаблон поиска

Шаблоном поиска является периметр области данных шириной в один модуль. Две смежные стороны — левая и нижняя — являются сплошными линиями и формируют L-образную границу. Они используются, прежде всего, для определения реального размера, ориентации и искажений символа. Две противоположные стороны состоят из чередующихся темных и светлых модулей. Они используются, прежде всего, для определения структуры символа, состоящей из ячеек, но также могут применяться для определения физического размера и искажений символа. Наличие свободной зоны обозначено на рисунке 1 угловыми метками.

6.3.3 Размеры и емкость символов

Символы Data Matrix состоят из четного числа строк и четного числа столбцов. Символы могут быть квадратной формы с размерами (в модулях) от 10 × 10 до 144 × 144 без учета свободных зон либо прямоугольной формы размерами (в модулях) от 8 × 18 до 16 × 48 без учета свободных зон. Для всех символов штрихового кода Data Matrix модуль, расположенный в верхнем правом углу, имеет коэффициент отражения со значением, противоположным (т. е. быть светлым или темным) по отношению к значению коэффициента отражения L-образной части шаблона поиска (см. рисунок 1). Полный перечень атрибутов приведен в таблице 10.

7 Требования к кодированию Data Matrix

7.1 Основные положения процедуры кодирования

7.1.1 Общие положения

В 7.1 приведены общие сведения о процедуре кодирования. В следующих разделах приведено более детальное рассмотрение указанной процедуры. Пример кодирования приведен в приложении I. Преобразование данных пользователя в символ штрихового кода Data Matrix происходит в следующей последовательности.

7.1.2 Этап 1: кодирование данных

Поскольку символы Data Matrix могут содержать различные схемы кодирования, которые позволяют более эффективно преобразовать найденные множества знаков в кодовые слова по сравнению со схемой кодирования, принятой по умолчанию, анализируют поток данных для определения разнообразия типов различных знаков, подлежащих кодированию. Для переключения между схемами кодирования и для выполнения других функций вводят дополнительные кодовые слова. Для образования требуемого числа кодовых слов добавляют необходимое количество знаков-заполнителей. Если пользователь не установил размер матрицы, то выбирают наименьший размер, в котором могут быть размещены данные. Полный перечень размеров матриц приведен в таблице 10.

Таблица 1 — Схемы кодирования по правилам Data Matrix

Наименование схемы кодирования	Знаки	Число битов на один знак данных
ASCII	Сдвоенные разряды чисел	4
	Знаки набора знаков ASCII с десятичными значениями от 0 до 127	8
	Знаки расширенного набора знаков ASCII с десятичными значениями от 128 до 255	16
C40	Цифры и прописные латинские буквы	5,33
	Специальные знаки и строчные латинские буквы	10,66[a]
Text	Цифры и строчные латинские буквы	5,33
	Специальные знаки и прописные латинские буквы	10,66[b]
X12	Набор знаков данных для электронного обмена данными по ANSI X12	5,33
EDIFACT	Знаки набора знаков ASCII с десятичными значениями от 32 до 94	6
Base 256	Любые байты с десятичными значениями от 0 до 255	8

[a] Кодируют как два значения в схеме кодирования C40 с использованием знака регистра (Shift).
[b] Кодируют как два значения в схеме кодирования Text с использованием знака регистра (Shift).

7.1.3 Этап 2: формирование кодовых слов проверки и исправления ошибок

Для символов, содержащих более 255 кодовых слов, поток кодовых слов подразделяют на чередующиеся блоки, чтобы дать возможность обработки алгоритмами исправления ошибок, как показано в приложении А. Для каждого блока формируют кодовые слова исправления ошибок. Результатом этого процесса является удлинение потока кодовых слов на число кодовых слов исправления ошибок. Кодовые слова исправления ошибок помещают после кодовых слов данных.

7.1.4 Этап 3: размещение модулей в матрице

Модули кодовых слов размещают в матрице. В матрицу вставляют модули направляющих шаблонов (при их наличии). Вокруг матрицы добавляют модули шаблона поиска.

7.2 Кодирование данных

7.2.1 Общие положения

Данные можно кодировать с использованием любой комбинации из шести схем кодирования (см. таблицу 1), при этом схема кодирование ASCII является основной схемой. Остальные схемы кодирования вызывают из схемы кодирования ASCII с последующим возвратом к этой же схеме. Следует учитывать эффективность уплотнения (число битов на знак данных), приведенную в таблице 1. Лучшей схемой для выбранного набора данных может оказаться не та, у которой на знак данных приходится наименьшее число битов. Если требуется наибольшая степень уплотнения, то следует принимать в расчет служебную информацию для переключения между схемами кодирования и наборами знаков внутри одной схемы кодирования (см. приложение J). Следует также учитывать, что даже если число кодовых слов минимизировано, поток кодовых слов может нуждаться в расширении для полного заполнения символа. Дополнение осуществляют путем использования знаков-заполнителей.

7.2.2 Интерпретация знаков по умолчанию

Интерпретация знаков по умолчанию для знаков с десятичными значениями от 0 до 127 должна соответствовать набору знаков IRV по ISO/IEC 646, а для знаков с десятичными значениями от 128 до 255 — набору знаков по ISO/IEC 8859-1. Графические представления знаков данных, приведенных в настоящем стандарте, соответствуют интерпретации по умолчанию. Эта интерпретация может быть изменена с помощью переключающих последовательностей интерпретации в расширенном канале, см. 7.3. Интерпретацией по умолчанию является интепретация ECI 000003.

7.2.3 Схема кодирования ASCII

Схема кодирования ASCII представляет собой набор знаков по умолчанию для первого знака символа в символах любого размера. С помощью указанной схемы кодируют знаки набора знаков ASCII, числовые данные двойной плотности и управляющие знаки символики. Управляющие знаки символики включают в себя функциональные знаки, знак-заполнитель и знаки-переключатели на другие кодовые наборы. Знаки набора данных ASCII кодируют как кодовые слова со значениями от 1 до 128 (десятичное значение знака набора знаков ASCII плюс 1). Знаки расширенного набора знаков ASCII с десятичными значениями от 128 до 255 кодируют с использованием управляющего знака верхнего регистра (Upper Shift) (см. 7.2.4.3). Пары цифр от 00 до 99 кодируют кодовыми словами со значениями от 130 до 229 (числовое значение плюс 130). Присвоенные значения кодовых слов для схемы кодирования ASCII приведены в таблице 2.

Примечание — Схема кодирования ASCII является наименованием набора знаков в символике Data Matrix. Не следует путать с набором знаков IRV по ISO/IEC 646.

Таблица 2 — Значения кодовых слов в схеме кодирования ASCII

Значение кодового слова	Знак данных или функция
0	Не подлежит использованию в схеме кодировании ASCII
1—128	Знаки данных набора знаков ASCII (десятичное значение знака + 1)
129	Знак-заполнитель (Pad)
130—229	Пары цифр от 00 до 99 (числовое значение + 130)
230	Знак фиксации (Latch) схемы кодирования C40
231	Знак фиксации (Latch) схемы кодирования Base 256

Окончание таблицы 2

Значение кодового слова	Знак данных или функция
232	Знак FNC1
233	Знак структурированного соединения (Structured Append)
234	Знак программирования устройства считывания (Reader Programming)
235	Знак верхнего регистра (Upper Shift) (переход к расширенному набору знаков ASCII)
236	Знак Макро 05 (Macro 05)
237	Знак Макро 06 (Macro 06)
238	Знак фиксации (Latch) схемы кодирования X12
239	Знак фиксации (Latch) схемы кодирования Text
240	Знак фиксации (Latch) схемы кодирования EDIFACT
241	Знак интерпретации ECI
242—255	Не подлежит использованию в схеме кодирования ASCII

7.2.4 Управляющие знаки символики

7.2.4.1 Общие положения

В символах Data Matrix существуют несколько специальных управляющих знаков символики, имеющих особое значение для схемы кодирования. Эти знаки должны использоваться для сообщения команды декодеру на выполнение определенных функций или передачи управляющему компьютеру специальных данных, приведенных в 7.2.4.2 — 7.2.4.10. Указанные управляющие знаки символики, за исключением знаков со значениями от 242 до 255, присутствуют в схеме кодирования ASCII (см. таблицу 2).

7.2.4.2 Знаки фиксации (Latch)

Для переключения из схемы кодирования ASCII в любую иную схему кодирования используют знаки фиксации (latch character). Все кодовые слова после знака фиксации должны кодироваться в соответствии с новой схемой кодирования. Различные схемы кодирования имеют свои способы возврата к схеме кодирования ASCII.

7.2.4.3 Знак верхнего регистра (Upper Shift character)

Знак верхнего регистра (Upper Shift) используется в комбинации со значением знака набора ASCII (от 1 до 128) для кодирования знака расширенного набора знаков ASCII с десятичными значениями (от 129 до 255). Знак расширенного набора знаков ASCII, кодируемый в схемах кодирования ASCII, C40 или Text, требует наличия предшествующего знака верхнего регистра (Upper Shift), после которого следует знак расширенного набора знаков ASCII, значение которого уменьшено на 128. Эту пару кодируют в соответствии с правилами схемы кодирования. В схеме кодирования ASCII знак верхнего регистра (Upper Shift) представлен кодовым словом со значением 235. Уменьшенное значение знака данных (т. е. значение знака расширенного набора знаков ASCII минус 128) преобразуют в значение кодового слова путем прибавления к его значению единицы. Например, для кодирования знака ¥ [(денежный знак иены), значение которого в расширенном наборе знаков ASCII равно 165] после знака верхнего регистра (Upper Shift) (с кодовым словом со значением 235) следует поставить знак набора знаков ASCII с десятичным значением 37 (165 – 128), которое кодируется как кодовое слово со значением 38. При наличии протяженных последовательностей знаков данных расширенного набора знаков ASCII более эффективное кодирование может быть достигнуто путем использования знака фиксации схемы кодирования Base 256.

7.2.4.4 Знак-заполнитель (Pad)

Если кодируемых данных, независимо от используемой схемы кодирования, не хватает для полного заполнения емкости символа для данных, то необходимо добавить знаки-заполнители (со значением кодового слова 129 в схеме кодирования ASCII) для заполнения оставшейся емкости символа для данных. Знаки-заполнители должны использоваться исключительно для указанной цели. Перед вводом знака-заполнителя необходимо вернуться к схеме кодирования ASCII из любой другой используемой схемы кодирования. Алгоритм рандомизации с шаблоном из 253 состояний применяют к знакам-заполнителям, ко всей последовательности знаков-заполнителей, начиная со второго знака-заполнителя и до конца символа, в соответствии с приложением В.

7.2.4.5 Знак интерпретации в расширенном канале (ECI)

Знак интерпретации в расширенном канале (далее знак интерпретации ECI) используют для смены интерпретации, принятой по умолчанию, на иную интерпретацию, применяемую для кодирования данных. Протокол интерпретаций в расширенном канале (ECI) является общим для многих символик, и его применение к символике Data Matrix более полно определено в 7.3. После знака интерпретации ECI обязательно должны следовать одно, два или три кодовых слова, которые идентифицируют конкретную активизируемую интерпретацию ECI. Новая интерпретация ECI действует до конца кодируемых данных или до тех пор, пока другой знак интерпретации ECI не вызовет иную интерпретацию. См. также приложение М.

7.2.4.6 Знаки регистра (Shift) в схемах кодирования C40 и Text

В схемах кодирования C40 и Text используют три специальных знака, называемые знаками регистра (Shift), в качестве префикса к одному из 40 значений для кодирования примерно трех четвертей набора знаков ASCII. Это позволяет добиться более компактного кодирования оставшихся знаков ASCII с помощью одиночных значений.

7.2.4.7 Знак FNC1 как идентификатор альтернативного типа данных

Для кодирования данных, соответствующих специальным международным отраслевым стандартам, одобренных AIM Inc, знак FNC1 должен присутствовать в позиции первого или второго знака символа (либо пятой или шестой позиции данных в первом символе структурированного соединения символов). Знак FNC1, кодируемый в позиции любого иного знака символа, используют как разделитель полей, и он подлежит передаче как управляющий знак G_S (знак набора ASCII с десятичным значением 29).

7.2.4.8 Знаки Макро (Macro)

Символика Data Matrix обеспечивает представление специальных международных отраслевых головной (header) и конечной (trailer) меток в одном знаке символа, которое сокращает число знаков символа, необходимых для кодирования данных в символе при использовании установленных структурированных форматов. Любой знак Макро (Macro) применяют только в позиции первого знака символа. Эти знаки не должны использоваться вместе со структурированным соединением (Structured Append). Функции знаков Макро (Macro) приведены в таблице 3. Головная метка (header) должна быть включена в передаваемый поток данных в виде префикса, а конечная метка (trailer) — суффикса. Если используют идентификатор символики (symbology identifier), то он должен предшествовать головной метке (header).

Таблица 3 — Функции знаков Макро (Macro)

Значение кодового слова знака Макро (Macro)	Обозначение знака	Интерпретация	
		Головная метка (header)	Конечная метка (trailer)
236	Макро 05 (Macro 05)	$[)>{}^{R}_{S}05{}^{G}_{S}$	${}^{R}_{S}\,{}^{E}O_{T}$
237	Макро 06 (Macro 06)	$[)>{}^{R}_{S}06{}^{G}_{S}$	${}^{R}_{S}\,{}^{E}O_{T}$

7.2.4.9 Знак структурированного соединения (Structured Append)

Знак структурированного соединения используют для указания того, что символ является частью последовательности символов структурированного соединения в соответствии с 7.5.

7.2.4.10 Знак программирования устройства считывания

Знак программирования устройства считывания (Reader Programming) указывает на то, что в символе закодировано сообщение, предназначенное для программирования устройства считывания. Знак программирования устройства считывания (Reader Programming) должен быть первым кодовым словом символа и не подлежит использованию совместно со структурированным соединением.

7.2.5 Схема кодирования C40

7.2.5.1 Общие положения

Схему кодирования C40 применяют для оптимизации кодирования данных, состоящих из последовательности, включающей прописные буквы латинского алфавита и числа. Данная схема позволяет также кодировать и другие знаки путем использования знаков регистра (shift) в комбинации с другими знаками данных.

Знаки в схеме кодирования C40 разделены на четыре набора. Знаки из первого набора, называемого основным набором (basic set), содержат три специальных знака регистра (shift), знак Пробел и знаки набора знаков ASCII с A по Z и с 0 по 9. Каждому знаку данных соответствует единственное значение схемы кодирования C40 (далее — значение C40). Знаки других наборов присваивают одному из трех знаков регистра (shift), которые указывают на один из трех оставшихся наборов и сопровождаются одним из значений C40 (см. таблицу С.1).

В результате первого этапа кодирования каждый знак данных преобразуют в одно значение С40 или в пару значений С40. Затем полную строку значений С40 разбивают на группы по три значения (если в конце данных остается одно или два значения, то применяют специальные правила, приведенные в 7.2.5.3). После этого три значения (*C1*, *C2*, *C3*) кодируют как одно 16-битовое значение по формуле (1600×*C1*) + (40×*C2*) + *C3* + 1. В завершение каждое 16-битовое значение кодируют в двух кодовых словах, представляющих собой восемь старших битов и восемь младших битов.

7.2.5.2 Переключение на схему кодирования С40 и обратно

На схему кодирования С40 можно переключиться из схемы кодирования ASCII, используя соответствующее кодовое слово фиксации (latch) схемы кодирования С40 со значением 230. Кодовое слово со значением 254, непосредственно следующее за парой кодовых слов в схеме кодирования С40, действует как кодовое слово отказа от фиксации (unlatch) для возврата к схеме кодирования ASCII. В противном случае кодирование по схеме кодирования С40 сохраняется до окончания данных, закодированных в символе.

7.2.5.3 Правила кодирования С40

Каждая пара кодовых слов представляет собой 16-битовое значение, в котором первое кодовое слово соответствует восьми старшим битам, а второе — восьми младшим битам. Три значения С40 (*C1*, *C2*, *C3*) кодируют по формуле

$$(1600 \times C1) + (40 \times C2) + C3 + 1$$

В результате получают значения от 1 до 64 000. Уплотнение трех значений С40 в два кодовых слова представлено в таблице 4. Для кодирования знаков, принадлежащих наборам Регистр 1 (Shift 1), Регистр 2 (Shift 2) и Регистр 3 (Shift 3), сначала следует закодировать соответствующий знак регистра (shift), а затем — значение С40 для данных. Кодирование по схеме С40 может действовать до окончания кодовых слов символа, кодирующих данные.

Если в символе остается только один или два знака символа до начала следования кодовых слов исправления ошибки, то необходимо придерживаться следующих правил:

a) если остаются два знака символа и кодированию подлежат три оставшихся значения С40 [которые могут включать как знаки данных, так и знаки регистра (shift)], то эти три значения С40 кодируют в двух последних знаках символа. Заключительного кодового слова отказа от фиксации (unlatch) схемы кодирования не требуется;

b) если остаются два знака символа и кодированию подлежат два оставшихся значения С40 [первое значение С40 может быть знаком регистра (shift) или знаком данных, а второе значение представляет знак данных], добавляют к вводу знак Регистр 1 (Shift 1) (со значением С40 — 0), чтобы увеличить его до трех знаков и обрабатывать его как обычно, как ввод с любыми тремя знаками С40. Кодового слова отказа от фиксации схемы кодирования (unlatch) также не требуется;

c) если остаются два знака символа для кодирования одного оставшегося значения С40 (знака данных), то в первом из двух оставшихся знаков символа (предпоследнем знаке символа) кодируют отказ от фиксации схемы кодирования (unlatch), а в последнем знаке символа кодируют знак данных по схеме кодирования ASCII;

d) если остается один знак символа для кодирования одного оставшегося значения С40 (знака данных), то в последнем знаке символа кодируют знак данных по схеме кодирования ASCII. Знак отказа от фиксации схемы кодирования (unlatch) не кодируют, его наличие подразумевается перед последним знаком символа.

Во всех остальных случаях либо используют знак отказа от фиксации схемы кодирования (unlatch) для выхода из схемы кодирования С40 перед окончанием символа, либо применяют символ большего размера для кодирования данных.

Таблица 4 — Пример кодирования по схеме кодирования С40

Исходные знаки данных	AIM
Полученные значения С40	14, 22, 26
Вычисление 16-битового значения	(1600 × 14) + (40 × 22) + 26 + 1 = 23307
Определение первого кодового слова: (16-битовое значение) div 256	23307 div 256 = 91
Определение второго кодового слова: (16-битовое значение) mod 256	23307 mod 256 = 11
Итоговые кодовые слова	91, 11

7.2.5.4 Использование знака Upper Shift в схеме кодирования С40

В схеме кодирования С40 знак верхнего регистра (Upper Shift) не является функциональным знаком символики, а используется как знак регистра (shift) внутри данного кодового набора. При обнаружении знака данных расширенного набора ASCII необходимо закодировать три или четыре значения по схеме кодирования С40 в соответствии со следующим правилом.

Если [десятичное значение знака расширенного набора ASCII минус 128] принадлежит основному набору (basic set), то используют запись:

[1 (значение знака Регистр 2/Shift 2)] [30 (значение знака верхнего регистра UperShift)] [V (десятичное значение знака расширенного набора ASCII минус 128)].

В противном случае запись приобретает следующий вид:

[1 (значение знака Регистр 2/Shift 2)] [30 (значение знака верхнего регистра/UperShift)] [0, 1 или 2 (значения знаков Регистр 1/Shift 1, Регистр 2/Shift 2 или Регистр 3/Shift 3 соответственно)] [V (десятичное значение знака расширенного набора ASCII минус 128)].

В данных записях число, приведенное в квадратных скобках, соответствует значению согласно таблице С.1, соответствующее значение С40 обозначено V.

7.2.6 Схема кодирования Text

7.2.6.1 Общие положения

Схема кодирования Text предназначена для кодирования обычного печатного текста, состоящего в основном из знаков нижнего регистра. По структуре она похожа на кодовый набор, используемый в схеме кодирования С40, за исключением того, что строчные буквы нижнего регистра кодируют напрямую [т. е. без использования знака регистра (shift)]. Знаки верхнего регистра предваряют знаком Регистр 3 (Shift 3). Полный кодовый набор знаков схемы кодирования Text приведен в таблице С.2.

7.2.6.2 Переключение на схему кодирования Text и обратно

На схему кодирования Text можно переключиться из схемы кодирования ASCII, используя соответствующее кодовое слово фиксации схемы кодирования (latch) со значением 239. Кодовое слово со значением 254, непосредственно следующее за парой кодовых слов в схеме кодирования Text, действует как кодовое слово отказа от фиксации (unlatch) для возврата в схему кодирования ASCII. В противном случае схема кодирования Text действует до окончания данных, кодируемых в символе.

7.2.6.3 Правила кодирования в схеме кодирования Text

Применяют те же правила, что и в схеме кодирования С40.

7.2.7 Схема кодирования ANSI X12

7.2.7.1 Общие положения

Схему кодирования ANSI X12 применяют для кодирования знаков, используемых при стандартном электронном обмене данными по ANSI X12, где три знака данных размещают с уплотнением в двух кодовых словах способом, в некоторой степени, подобным схеме кодирования С40. Схема кодирования ANSI X12 позволяет кодировать буквы верхнего регистра (прописные латинские буквы), цифры, знак Пробел и три ограничительных и разделительных знака в соответствии со стандартом ANSI X12. Соответствие кодовых значений по ANSI X12 приведено в таблице 5. В наборе кодируемых знаков по ANSI X12 отсутствуют знаки регистра (shift).

Таблица 5 — Набор кодируемых знаков по ANSI X12

Значение знака X12	Кодируемые знаки	Десятичные значения знаков набора знаков ASCII
0	Знак-ограничитель сегмента X12 (X12 segment terminator) <CR>	13
1	Знак-разделитель сегмента X12 (X12 segment separator) * (звездочка)	42
2	Знак-разделитель подэлементов X12 (X12 sub-element separator) > (больше)	62
3	Пробел	32
4—13	От 0 до 9	48—57
14—39	От A до Z	65—90

7.2.7.2 Переключение на схему кодирования ANSI X12 и обратно

На схему кодирования ANSI X12 можно переключиться из схемы кодирования ASCII, используя соответствующее кодовое слово фиксации схемы кодирования (latch) (значение 238). Кодовое слово со

значением 254, непосредственно следующее за парой кодовых слов схемы кодирования ANSI X12, действует как кодовое слово отказа от фиксации (unlatch) для возврата в схему кодирования ASCII. В противном случае схема кодирования ANSI X12 действует до окончания данных, кодируемых в символе.

7.2.7.3 Правила кодирования по схеме кодирования ANSI X12

Применяют правила, установленные для схемы кодирования C40. Исключение составляет окончание кодирования данных ANSI X12. Если знаки данных не полностью заполняют пары кодовых слов, то сразу за последней полной парой кодовых слов следует использовать переключение в схему кодирования ASCII с помощью кодового слова значением 254 и продолжить использование схемы кодирования ASCII за исключением случая, когда остается единственный конечный знак символа (кодовое слово) перед первым кодовым словом исправления ошибок. Этот единственный знак символа кодируется по схеме кодирования ASCII без использования кодового слова отказа от фиксации (unlatch).

7.2.8 Схема кодирования EDIFACT

7.2.8.1 Общие положения

Схема кодирования EDIFACT включает в себя значения 63 знаков набора ASCII (с десятичными значениями от 32 до 94), а также знак отказа от фиксации (unlatch) (с двоичным значением 011111) для возврата в схему кодирования ASCII. Схема кодирования EDIFACT позволяет кодировать четыре знака данных в трех кодовых словах. Знаки данных включают в себя все цифры, буквы латинского алфавита и специальные графические знаки (знаки пунктуации/punctuation character), определенные в наборе знаков «EDIFACT Level A» без знаков регистра (shift), используемых в схеме кодирования C40.

7.2.8.2 Переключение на схему кодирования EDIFACT и обратно

На схему кодирования EDIFACT можно переключиться из схемы кодирования ASCII, используя соответствующее кодовое слово фиксации схемы кодирования (latch) со значением 240. Знак отказа от фиксации (latch) в схеме кодирования EDIFACT следует использовать в качестве знака-ограничителя окончания схемы кодирования EDIFACT для возврата в схему кодирования ASCII.

7.2.8.3 Правила кодирования в соответствии со схемой кодирования EDIFACT

Набор знаков в схеме кодирования EDIFACT приведен в таблице C.3. Существует простое соответствие между 6-битовыми значениями знаков по EDIFACT и 8-битовыми байтами знаков расширенного набора знаков ASCII. При построении 6-битового значения знака по EDIFACT исключают два бита старших разрядов 8-битового байта в соответствии с таблицей 6. Строки из четырех знаков со значениями по EDIFACT кодируют в три кодовых слова. В процессе простого кодирования два бита старших разрядов удаляют из 8-битового байта. Оставшийся 6-битовый байт является значением по EDIFACT и должен быть непосредственно закодирован в кодовом слове в соответствии с таблицей 7. Когда кодирование EDIFACT завершается знаком отказа от фиксации схемы кодирования (unlatch), любые биты, оставшиеся в одиночном знаке символа, следует заполнять нулями. Схема кодирования ASCII начинается со следующего знака символа. Если схема кодирования EDIFACT действует до конца символа, и до первого знака исправления ошибок осталось закодировать только одно или два кодовых слова, оставшихся за последним триплетом кодовых слов по схеме кодирования EDIFACT, их следует кодировать по схеме кодирования ASCII без использования знака отказа от фиксации (unlatch).

Таблица 6 — Соответствие значений знаков по EDIFACT и значений 8-битовых байтов

Знак данных	Значение знака набора ASCII		Значение знака по EDIFACT
	Десятичное значение	8-битовый байт	
A	65	01000001	000001
9	57	00111001	111001
Примечание — В процессе декодирования, если начальный бит (6-й разряд) равен 1, то для построения 8-битового байта требуется вставить в качестве префикса биты 00. Если начальный бит (6-й разряд) равен нулю, то для построения 8-битового байта следует вставить в качестве префикса биты 01. Исключением является знак со значением по EDIFACT 011111, который является управляющим знаком символики с функцией отказа от фиксации (unlatch) для возврата в схему кодирования ASCII.			

Таблица 7 — Пример кодирования по EDIFACT

Знаки данных	D			A			T			A		
Исходные двоичные значения (по таблице C.3)	00	01	00	00	00	01	01	01	00	00	00	01

Окончание таблицы 7

Знаки данных	D				A				T				A			
Разделение по три 8-битовых байта	00	01	00	00	00	01	01	01	00	00	00	01				
Итоговые значения кодовых слов	16				21				1							

7.2.9 Схема кодирования Base 256

7.2.9.1 Общие положения

Схему кодирования Base 256 используют для кодирования любых данных, представленных 8-битовыми байтами, включая интерпретации в расширенном канале (ECI), и двоичных данных. Интерпретация, используемая по умолчанию, определена в 7.2.2. Алгоритм рандомизации с шаблоном из 255 состояний применяют к каждой последовательности по схеме кодирования Base 256, встречающейся в закодированных данных (см. B.2). Схема начинает действовать после знака фиксации (latch) схемы кодирования Base 256 и заканчивается на последнем знаке, определенном длиной поля в схеме кодирования Base 256.

7.2.9.2 Переключение на схему кодирования Base 256 и обратно

На схему кодирования Base 256 можно переключиться из схемы кодирования ASCII, используя соответствующее кодовое слово фиксации схемы кодирования (latch) со значением 231. По окончании данных, закодированных в соответствии со схемой кодирования Base 256, возврат к схеме кодирования ASCII осуществляется автоматически. Обращение к интерпретации в расширенном канале (ECI), отличающейся от принятой по умолчанию, должно быть выполнено до переключения на схему кодирования Base 256. Последовательность интерпретации ECI не требуется располагать непосредственно перед переключением в схему кодирования Base 256.

7.2.9.3 Правила кодирования по схеме кодирования Base 256

После переключения на схему кодирования Base 256 первые одно (*d1*) или два (*d1*, *d2*) кодовых слова устанавливают длину поля данных в байтах. Определение степени соответствия между длиной поля и значениями *d1* и *d2* приведено в таблице 8. Далее все кодирование должно быть представлено значениями в байтах.

Таблица 8 — Длина поля в схеме кодирования Base 256

Длина поля	Значения *d1*, *d2*	Допустимые значения *d*
До окончания символа	*d1* = 0	*d1* = 0
От 1 до 249	*d1* = заданная длина	*d1* = от 1 до 249
От 250 до 1555	*d1* = (заданная длина DIV 250) +249	*d1* = от 250 до 255
	d2 = заданная длина MOD 250	*d2* = от 0 до 249

7.3 Интерпретация в расширенном канале (ECI)

7.3.1 Общие положения

Протокол интерпретаций в расширенном канале (далее — интерпретации ECI) позволяет включать в выходной поток данных знаки различных интерпретаций, отличающиеся от набора знаков по умолчанию. Протокол интерпретации ECI единообразно определен для ряда символик. В символике Data Matrix поддерживаются четыре распространенных типа интерпретаций:

a) международные наборы знаков (или кодовые страницы);

b) интерпретации общего назначения, такие как шифрование и уплотнение;

c) определяемые пользователем интерпретации для замкнутых систем применения;

d) управляющая информация для структурированного соединения в небуферизованном режиме.

Протокол интерпретаций в расширенном канале полностью установлен в [1]. Протокол обеспечивает последовательный метод установления специфических интерпретаций значений байтов перед печатью и после декодирования. Конкретную интерпретацию ECI идентифицируют с помощью 6-разрядного числа, которое в символике Data Matrix кодируют знаком интерпретации ECI, за которым следует от одного до трех кодовых слов. Специальные интерпретации приведены в документе AIM Inc. «Регистр наборов знаков интерпретаций в расширенном канале» («Extended Channel Interpretations Character Set Register»). Интерпретация ECI может использоваться только с устройствами считывания,

позволяющими передавать идентификаторы символики. Устройства считывания, которые не могут передавать идентификаторы символики, не обеспечивают передачу данных из любого символа, содержащего интепретацию ECI. Исключение может быть сделано только в случае, если интерпретация(и) ECI может (могут) быть полностью обработана(ы) самим устройством считывания.

Заданная интерпретация в расширенном канале может быть вызвана в любом месте закодированного сообщения.

7.3.2 Кодирование интерпретаций ECI

Разнообразные схемы кодирования символики Data Matrix (установленные в таблице 1) могут применяться при любых интерпретациях ECI. Вызов интерпретации ECI может быть осуществлен только из схемы кодирования ASCII, после этого допускается переключение между любыми схемами кодирования. Используемый способ кодирования строго определен 8-битовыми значениями данных и не зависит от действующей интерпретации в расширенном канале. Например, последовательность знаков с десятичными значениями в диапазоне от 48 до 57 может быть наиболее эффективно закодирована в цифровом режиме, даже если они не будут интерпретироваться как числа. Присвоенный номер интерпретации ECI вводят с помощью кодового слова значением 241 (знак интерпретации ECI) в схеме кодирования ASCII. Одно, два или три дополнительных кодовых слова используют для кодирования присвоенного номера интерпретации ECI (ECI Assignment number). Правила кодирования приведены в таблице 9.

Следующие примеры приведены для иллюстрации кодирования:

Присвоенный номер интерпретации ECI = 015000

Кодовые слова:

[241] [(15000 – 127) div 254 + 128] [(15000 – 127) mod 254 + 1]

= [241] [58 + 128] [141 + 1]

= [241] [186] [142]

Присвоенный номер интерпретации ECI = 090000

Кодовые слова:

[241] [(90000 – 16383) div 64516 + 192] [((90000 – 16383) div 254) mod 254 + 1] [(90000 – 16383) mod 254 + 1]

= [241] [1 + 192] [289 mod 254 + 1] [211 + 1]

= [241] [193] [36] [212]

Таблица 9 — Кодирование присвоенных номеров интерпретаций ECI в символике штрихового кода Data Matrix[1)]

Присвоенный номер интерпретации ECI	Последовательность кодовых слов	Значения кодовых слов	Область значений
От 000000 до 000126	C_0	241	
	C_1	ECI_no + 1	C_1 = (от 1 до 127)
От 000127 до 016382	C_0	241	
	C_1	(ECI_no – 127) div 254 + 128	C_1 = (от 128 до 191)
	C_2	(ECI_no – 127) mod 254 + 1	C_2 = (от 1 до 254)
От 0016383 до 999999	C_0	241	
	C_1	(ECI_no – 16383) div 64516 +192	C_1 = (от 192 до 207)
	C_2	[(ECI_no – 16383) div 254] mod 254 + 1	C_2 = (от 1 до 254)
	C_3	(ECI_no – 16383) mod 254 + 1	C_3 = (от 1 до 254)

7.3.3 Интерпретации ECI и структурированное соединение

Интерпретации ECI могут появляться в любом месте сообщения, закодированного в одиночном символе или в символе структурированного соединения (Structured Append) (см. 7.5) набора символов Data Matrix. Любая активизированная интерпретация ECI сохраняет действие либо до конца закодированных данных, либо до появления другой интерпретации ECI. Таким образом, интерпретация в заданной ECI может распространяться на два или более символов.

[1)] *В настоящей таблице обозначение «ECI_no» соответствует присвоенному номеру интерпретации ECI, обозначения C_0, C_1, C_2, C_3 соответствуют значениям кодовых слов.*

7.3.4 Протокол после декодирования

Протокол передачи данных интерпретации ECI определен в 12.5. При применении интерпретаций ECI следует полностью внедрить использование идентификаторов символики (см. 12.6), а соответствующий идентификатор символики должен передаваться в качестве преамбулы перед декодированными данными.

7.4 Атрибуты символа Data Matrix

7.4.1 Размеры и емкости символов

В символике штрихового кода Data Matrix доступны 24 квадратных и 6 прямоугольных символов, указанных в таблице 10.

Примечание — Дополнительные форматы прямоугольных символов установлены в ISO/IEC 21471.

7.4.2 Включение направляющих шаблонов в символы большого размера

Согласно таблице 10 символы квадратной формы, размерами (в модулях) 32 × 32 и более, и четыре прямоугольных символа размерами (в модулях) 8 × 32, 12 × 36, 16 × 36 и 16 × 48 имеют две или более области данных (data regions). Эти области данных ограничивают направляющими шаблонами (alignment patterns) в соответствии с приложением D. Символы квадратной формы делят на 4, 16 или 36 областей данных (как показано на рисунках D.1, D.2 и D.3). Прямоугольные символы делят на две области данных (как показано на рисунке D.4). Чередующиеся темные модули направляющего шаблона (alignment pattern) должны быть расположены на верхней и правой границах области данных и идентифицировать четные столбцы и строки.

7.5 Структурированное соединение

7.5.1 Основные принципы

В структурированном формате может присутствовать до 16 символов Data Matrix. Символ является частью структурированного соединения (Structured Append), что отмечается кодовым словом значением 233 в позиции первого знака символа. Непосредственно за ним следуют три кодовых слова структурированного соединения. Первое кодовое слово является индикатором позиции символа в группе, второе и третье предназначены для идентификации файла.

7.5.2 Индикатор позиции символа

Кодовое слово индикатора позиции символа задает положение конкретного символа внутри группы (до 16) символов Data Matrix в формате структурированного соединения в виде значения «*m* из общего числа *n* символов». Первые четыре бита в данном кодовом слове указывают позицию данного символа как двоичное значение, равное (*m* – 1). Последние 4 бита задают общее число символов, подлежащих объединению в формате структурированного соединения как двоичное значение, равное (17 – *n*). 4-битовые комбинации должны соответствовать установленным в таблице 11.

Например кодирование кодового слова индикатора позиции символа для третьего символа в группе из семи символов проводят следующим образом:

значение битов в третьей позиции символа: 0010;

общее число символов 7: 1010;

комбинация битов: 00101010;

значение кодового слова: 42.

7.5.3 Идентификация файла

Идентификацию файла задают значениями двух кодовых слов. Каждое кодовое слово идентификации файла может иметь значение от 1 до 254, что допускает 64 516 различных вариантов идентификации файла. Идентификация файла предназначена для повышения вероятности того, что только логически связанные символы обрабатываются как часть единого сообщения.

7.5.4 Знак FNC1 и структурированное соединение

Если структурированное соединение (Structured Append) используется в сочетании со знаком FNC1 (см. 7.2.4.7), то первые четыре кодовых слова следует применять для структурированного соединения, а пятое и шестое доступны для использования знака FNC1. Знак FNC1 не должен повторяться в этих же позициях во втором и последующем символах, если только он не используется в качестве знака — разделителя полей.

Таблица 10 — Атрибуты символов Data Matrix

Размер символа[a]		Область данных		Размер координатной матрицы	Общее число кодовых слов		Число кодовых слов в блоке Рида—Соломона		Число чередующихся блоков	Максимальная емкость символа для данных			Кодовые слова исправления ошибок, %	Максимальное число восстановленных кодовых слов, ошибок/стираний[b]
число строк	число столбцов	размер	число областей данных		данных	исправления ошибок	данных	исправления ошибок		число числовых разрядов	число латинских букв и цифр[d]	число байтов		
10	10	8 × 8	1	8 × 8	3	5	3	5	1	6	3	1	62,5	2/0
12	12	10 × 10	1	10 × 10	5	7	5	7	1	10	6	3	58,3	3/0
14	14	12 × 12	1	12 × 12	8	10	8	10	1	16	10	6	55,6	5/7
16	16	14 × 14	1	14 × 14	12	12	12	12	1	24	16	10	50	6/9
18	18	16 × 16	1	16 × 16	18	14	18	14	1	36	25	16	43,8	7/11
20	20	18 × 18	1	18 × 18	22	18	22	18	1	44	31	20	45	9/15
22	22	20 × 20	1	20 × 20	30	20	30	20	1	60	43	28	40	10/17
24	24	22 × 22	1	22 × 22	36	24	36	24	1	72	52	34	40	12/21
26	26	24 × 24	1	24 × 24	44	28	44	28	1	88	64	42	38,9	14/25
32	32	14 × 14	4	28 × 28	62	36	62	36	1	124	91	60	36,7	18/33
36	36	16 × 16	4	32 × 32	86	42	86	42	1	172	127	84	32,8	21/39
40	40	18 × 18	4	36 × 36	114	48	114	48	1	228	169	112	29,6	24/45
44	44	20 × 20	4	40 × 40	144	56	144	56	1	288	214	142	28	28/53
48	48	22 × 22	4	44 × 44	174	68	174	68	1	348	259	172	28,1	34/65
52	52	24 × 24	4	48 × 48	204	84	102	42	2	408	304	202	29,2	42/78
64	64	14 × 14	16	56 × 56	280	112	140	56	2	560	418	277	28,6	56/106
72	72	16 × 16	16	64 × 64	368	144	92	36	4	736	550	365	28,1	72/132
80	80	18 × 18	16	72 × 72	456	192	114	48	4	912	682	453	29,6	96/180
88	88	20 × 20	16	80 × 80	576	224	144	56	4	1 152	862	573	28	112/212
96	96	22 × 22	16	88 × 88	696	272	174	68	4	1392	1042	693	28,1	136/260
104	104	24 × 24	16	96 × 96	816	336	136	56	6	1632	1222	813	29,2	168/318

Окончание таблицы 10

Размер символа[a]		Область данных		Размер координатной матрицы	Общее число кодовых слов		Число кодовых слов в блоке Рида—Соломона		Число чередующихся блоков	Максимальная емкость символа для данных			Кодовые слова исправления ошибок, %	Максимальное число восстановленных кодовых слов, ошибок/стираний[b]
число строк	число столбцов	размер	число областей данных		данных	исправления ошибок	данных	исправления ошибок		число числовых разрядов	число латинских букв и цифр[d]	число байтов		
120	120	18 × 18	36	108 × 108	1 050	408	175	68	6	2100	1573	1047	28	204/390
132	132	20 × 20	36	120 × 120	1 304	496	163	62	8	2608	1954	1301	27,6	248/472
144	144	22 × 22	36	132 × 132	1 558	620	156	62	8[c]	3116	2335	1555	28,5	310/590
							155	62	2[c]					
Символы прямоугольной формы														
8	18	6 × 16	1	6 × 16	5	7	5	7	1	10	6	3	58,3	3/0
8	32	6 × 14	2	6 × 28	10	11	10	11	1	20	13	8	52,4	5/0
12	26	10 × 24	1	10 × 24	16	14	16	14	1	32	22	14	46,7	7/11
12	36	10 × 16	2	10 × 32	22	18	22	18	1	44	31	20	45,0	9/15
16	36	14 ×16	2	14 × 32	32	24	32	24	1	64	46	30	42,9	12/21
16	48	14 × 22	2	14 × 44	49	28	49	28	1	98	72	47	36,4	14/25

[a] В размер символа не включена свободная зона.

[b] По 7.6.3.

[c] В символе наибольшего размера (144 × 144) первые восемь блоков по Риду—Соломону включают в 218 кодовых слов, кодирующих 156 кодовых слов данных. Последние два блока включают 217 кодовых слов (155 слов данных). Все блоки содержат 62 кодовых слова исправления ошибок.

[d] В соответствии с правилами схем кодировании Text или C40 без использования знаков фиксации и регистра; при кодировании с помощью других схем это значение может значительно варьироваться в зависимости от сочетания и группирования наборов знаков.

Таблица 11 — Значения битов позиции символа в структурированном соединении

Позиция символа	Значения битов с 1 по 4	Общее число символов	Значения битов с 5 по 8
1	0000		
2	0001	2	1111
3	0010	3	1110
4	0011	4	1101
5	0100	5	1100
6	0101	6	1011
7	0110	7	1010
8	0111	8	1001
9	1000	9	1000
10	1001	10	0111
11	1010	11	0110
12	1011	12	0101
13	1100	13	0100
14	1101	14	0011
15	1110	15	0010
16	1111	16	0001

7.5.5 Буферизованные и небуферизованные операции

Сообщение, содержащееся в рамках последовательности структурированного соединения (Structured Append), может быть целиком накоплено в буфере устройства считывания до своего полного ввода и передано после того, как считаны все символы. В качестве альтернативы устройство считывания может передавать декодированные данные из каждого символа по мере их считывания. В этой небуферизованной операции протокол интерпретаций ECI для структурированного соединения (установленный в стандарте AIM ITS 04/001, часть 1)[1)] определяет управляющий блок, который должен вставляться в качестве префикса перед началом передаваемых данных каждого символа.

7.6 Обнаружение и исправление ошибок

7.6.1 Исправление ошибок Рида—Соломона

В символах Data Matrix используют исправление ошибок Рида—Соломона. Для символов Data Matrix с общим числом кодовых слов менее 255 кодовые слова исправления ошибки вычисляют с помощью кодовых слов данных без процедуры чередования. Для символов Data Matrix с общим числом кодовых слов более 255 кодовые слова исправления ошибки вычисляют с помощью кодовых слов данных с использованием процедуры чередования, приведенной в приложении А. Каждый символ Data Matrix характеризуется особым числом кодовых слов данных и исправления ошибок, которые разделены в определенном числе блоков как указано в таблице 10 и к которым применяется процедура чередования согласно приложению А.

Полиномиальные арифметические вычисления для символов Data Matrix должны проводиться с использованием побитового арифметического подсчета для битовых операций по модулю 2 и арифметического подсчета для байтовых операций по модулю 100101101 (десятичное значение 301). Это поле Галуа 2^8, где 100101101 соответствует простому минимальному многочлену поля $x^8 + x^5 + x^3 + x^2 + 1$. Используют 16 различных порождающих многочленов для вычисления соответствующих кодовых слов исправления ошибок, приведенных в Е.1.

Примечание — Пример реализации алгоритма исправления ошибок приведен в [2].

---

1) *См. [1].*

7.6.2 Генерация кодовых слов исправления ошибок

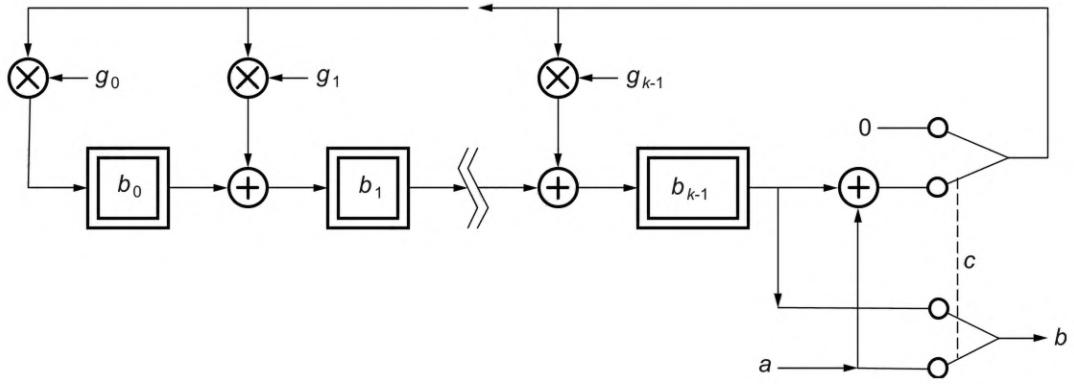
Кодовые слова исправления ошибок являются остатком от деления кодовых слов данных на полиномиальную функцию $g(x)$, используемую для кодов Рида—Соломона (см. Е.1).

Если это вычисление выполняется методом «длинного деления», то полином данных символа сначала должен быть умножен на x^k.

Кодовые слова данных являются коэффициентами полинома с коэффициентом при наивысшей степени, равным первому кодовому слову данных, и с коэффициентом при низшей степени, равным последнему кодовому слову данных перед первым кодовым словом исправления ошибок. Коэффициент при наивысшей степени оставшейся части полинома является первым кодовым словом исправления ошибок, а коэффициент при нулевой степени является последним кодовым словом исправления ошибок и последним кодовым словом. Это может быть выполнено с помощью схемы деления, приведенной на рисунке 2. Регистры от b_0 до b_{k-1} инициализируют нулями. Существуют две стадии генерации кодирования. На первой стадии при положении ключа в нижней позиции кодовые слова данных передаются как на выход, так и на схему. Первая стадия завершается за n синхронизирующих импульсов. На второй стадии ($n + 1$, $n + k$ синхронизирующих импульсов), при положении ключа в верхнем положении, кодовые слова исправления ошибок $\varepsilon_{k-1}, ..., \varepsilon_0$ генерируются путем выдачи значений из всех регистров по порядку с сохранением нулевых данных на входе. Кодовые слова на выходе регистра сдвига (shift register) должны быть расположены в порядке, в котором они будут размещены в символе.

Примечание 1 — При использовании процедуры чередования указанные кодовые слова не будут размещены последовательно в знаках символа (см. приложение А).

Примечание 2 — n и k определены в разделе 4 как число кодовых слов данных и число кодовых слов исправления ошибок соответственно.



⊕ — GF(256) сложение; ⊗ — GF(256) умножение; *a* — вход; *b* — выход; *c* — переключатель

Рисунок 2 — Схема генерации кодовых слов исправления ошибки

7.6.3 Возможности исправления ошибок

Кодовые слова исправления ошибок позволяют исправлять два типа ошибочных кодовых слов: стирания (ошибочные кодовые слова с известными позициями) и ошибки (ошибочные кодовые слова с неизвестными позициями). Стирание представляет собой не сканированный или не подающийся декодированию знак символа. Ошибка представляет собой неправильно декодированный знак символа. Число стираний и ошибок, которые одновременно можно исправить, вычисляют по формуле (1):

$$e + 2t \le d - p, \tag{1}$$

где e — число стираний;

t — число ошибок;

d — число кодовых слов исправления ошибок;

p — число кодовых слов, зарезервированных для обнаружения ошибок.

В общем случае $p = 0$. Однако, если большая часть возможности исправления ошибок, свойственных коду, используется для исправления стираний, то возрастает возможность необнаружения ошибки. Если число стираний больше половины числа кодовых слов исправления ошибок, то $p = 3$. Для символов небольших размеров (10 × 10, 12 × 12, 8 × 18, 8 × 32) не следует использовать исправление стираний ($e = 0$ и $p = 1$).

7.7 Формирование символа

7.7.1 Общие положения

При заданной последовательности кодовых слов, рассмотренной в предыдущих подразделах, символ Data Matrix формируется в следующей последовательности этапов:

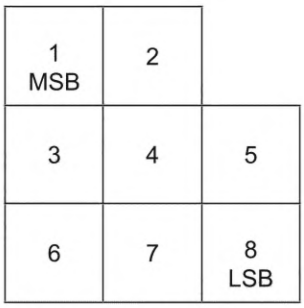
a) размещение модулей кодовых слов в координатной матрице;

b) подстановка модулей направляющего шаблона, при необходимости;

c) размещение модулей шаблона поиска по периметру символа.

7.7.2 Размещение знаков символа

Каждый знак символа должен включать в себя восемь номинально квадратных модулей, каждый из которых отображает один двоичный разряд. Темный модуль соответствует единице, светлый — нулю. Восемь модулей, упорядоченных слева-направо и сверху-вниз составляют форму знака символа, приведенную на рисунке 3. Вследствие того, что форма знака символа, приведенная на рисунке 3, не может точно вписаться в границы символа, некоторые знаки символа разбивают на части. Порядок размещения знаков символа определен программой на языке программирования С, указанной в F.1, приведенной в F.2 и проиллюстрированной в F.3. Следует использовать размещение знаков символа в соответствии с установленным в приложении F.



LSB — младший значащий разряд (Least significant bit); MSB — старший значащий разряд (Most significant bit)

Рисунок 3 — Представление кодового слова в знаке символа

7.7.3 Размещение модулей направляющего шаблона (Alinement Pattern)

Этот этап применяют только для крупных матриц:

- квадратной формы размерами (в модулях) от 32 × 32 и более;
- прямоугольной формы размерами (в модулях) от 8 × 32, 12 × 36 и более.

Для выбранного формата символа координатную матрицу разбивают на области данных размерами, установленными в таблице 10. Области данных отделяют друг от друга направляющими шаблонами (alignment pattern) шириной два модуля. В результате некоторые знаки символа будут разделены между двумя смежными областями данных. Для матриц квадратной формы направляющие шаблоны размещают между областями данных горизонтально и вертикально парами. Общее число пар направляющих шаблонов 2, 6 или 10 — в соответствии с представленным на рисунках D.1 — D.3. Для матриц прямоугольной формы между областями данных размещают только один вертикальный направляющий шаблон, как изображено на рисунке D.4.

7.7.4 Размещение модулей шаблона поиска (Finder Pattern)

Для формирования шаблона поиска модули следует размещать по периметру матрицы в соответствии с 6.3.2.

8 Размеры символов

Размеры символов Data Matrix должны соответствовать следующим требованиям:

- размер *X*: размер модуля должен быть установлен в стандарте по применению с учетом методов сканирования и технологии производства символа;
- шаблон поиска (finder pattern): ширина шаблона поиска должна быть равна размеру *X*;
- направляющий шаблон (alignment pattern): ширина направляющего шаблона должна быть равна 2*X*;
- свободная зона — минимальный размер свободной зоны вокруг символа Data Matrix должен быть равен размеру *X* со всех четырех сторон. Для прикладных систем, в которых необходимо уменьшить влияние областей, расположенных в непосредственной близости от символа, создающих повышенные отражающие помехи, рекомендуется оставлять свободную зону размером от 2*X* до 4*X*.

9 Качество печати символов

9.1 Общие положения

Качество печати символов Data Matrix должно оцениваться в соответствии с требованиями, установленными в ISO/IEC 15415 или ISO/IEC 29158 в соответствии с подходящей спецификацией по применению с учетом дополнений и изменений, приведенных в настоящем разделе.

Некоторые методы маркировки не позволяют наносить символы, полностью соответствующие требованиям настоящего стандарта, без применения специальных мер.

9.2 Параметры качества символа

9.2.1 Повреждение фиксированного шаблона

Методы измерения и оценивания параметра повреждения фиксированного шаблона установлены в приложении G.

9.2.2 Полный класс символа

Метод оценивания по ISO/IEC 15415 или ISO/IEC 29158 следует использовать в соответствии со спецификацией по применению.

9.2.3 Параметр декодирования (Decode)

Для получения класса параметра декодирования (Decode) следует использовать рекомендуемый алгоритм декодирования. Неспособность рекомендуемого алгоритма декодирования успешно декодировать символ оценивают как класс параметра декодирования (Decode) 0. Рекомендуемое декодирование создает фактическую сетку выборки, которая используется для выборки модулей символа с целью декодирования. В случае ошибочного кодирования (например, ненадлежащего следования требованиям 7.1, 7.2 и 7.3) оценивание символа также должно приводить к классу параметра декодирования (Decode) 0.

9.2.4 Параметр неоднородности сетки (Grid non-uniformity)

Идеальную сетку вычисляют, используя четыре угловые точки, находящиеся в окружающем фиксированном шаблоне/однородной темной области сетки выборки (см. рисунок 4), для каждой области данных и разделяя ее одинаково по обеим осям.

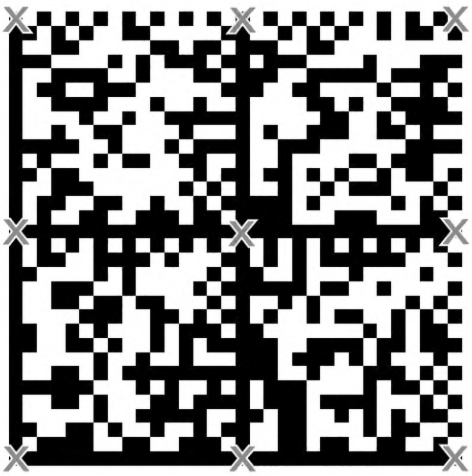


Рисунок 4 — Угловые точки в идеальной сетке

9.3 Измерения в процессе контроля

Для выполнения практичных измерений с целью мониторинга и контролирования процесса создания символов Data Matrix могут использоваться различные средства и методы. Указанные способы не устанавливают проверку качества печати изготавливаемых символов. Методом, необходимым для оценки качества печати символа, является метод, установленный в 9.2 и приложении G. Хотя иные способы по отдельности и в совокупности дают хорошие показатели того, создает ли процесс печати символов работоспособные символы.

10 Рекомендуемый алгоритм декодирования для символики Data Matrix

Данный рекомендуемый алгоритм декодирования позволяет находить на изображении символы Data Matrix и производить их декодирование следующим образом:

a) определяют значения параметров размеров и формируют цифровое изображение:

1) задают расстояние d_{min}, равное 7,5 диаметра апертуры, заданной в системе применения, которое считают минимальной длиной стороны L-образного шаблона поиска;
2) задают расстояние g_{max}, равное 7,5 диаметра апертуры. Это расстояние считают максимальным промежутком в L-образном шаблоне поиска, допускаемым алгоритмом поиска на этапе b);
3) задают расстояние m_{min}, равное 1,25 диаметра апертуры, которое считают номинальным наименьшим размером модуля, когда размер аппертуры составляет 80 % от размера *X* символа;
4) формируют черно-белое изображение, используя порог, определенный по методике, установленной в ISO/IEC 15415;

b) осуществляют поиск горизонтальной и вертикальной линий сканирования для двух внешних L-образных границ символа Data Matrix:

1) продлевают горизонтальную линию сканирования по обе стороны от центральной точки изображения и, производя выборку вдоль этой линии, находят все точки перехода черный/белый и белый/черный. Для каждой точки перехода, найденной на линии сканирования и приведенной к границе пикселя, производят следующие действия:
   i) следуют от этой точки вверх вдоль границы перехода черный/белый, пиксель за пикселем до достижения точки, расположенной на расстоянии $3{,}5m_{min}$ от пересечения линии сканирования и точки старта, или до той точки, в которой линия границы поворачивает вниз обратно к пересечению линии сканирования и границы перехода — точке старта;
   ii) следуют от точки старта вниз вдоль границы перехода черный/белый, пиксель за пикселем до достижения точки, расположенной на расстоянии $3{,}5m_{min}$ от пересечения линии сканирования и точки старта, или до той точки, в которой линия границы поворачивает вверх обратно к пересечению линии сканирования и границы перехода — точке старта;
   iii) если при движении вверх достигли точки, отстоящей на $3{,}5m_{min}$ от точки старта, выполняют следующие этапы:
      I) проводят линию A, соединяющую конечные точки вертикальной границы перехода;
      II) проверяют, чтобы отклонение промежуточных точек границы находилось в пределах $0{,}5m_{min}$ от прямой линии A. Если это условие выполняется, то продолжают выполнение этапа, указанного в перечислении III). В противном случае переходят к выполнению этапа, указанного в перечислении 1)iv), до достижения края границы перехода в противоположное направление;
      III) продолжают следовать вдоль границы перехода вверх до расстояния $0{,}5m_{min}$ от линии A или до тех пор, пока граница перехода не повернет назад на расстояние не менее $0{,}5m_{min}$. Возвращаются в ближайшую точку границы перехода, находящуюся на расстоянии по евклидовой дистанции, большем или равным m_{min} от последней точки границы перехода вдоль линии границы перехода, и сохраняют ее как конечную точку границы перехода. Эту точку границы перехода следует рассматривать как одно из предполагаемых положений границ внешнего края L-образных границ;
      IV) продолжают следовать вниз вдоль границы перехода до расстояния $0{,}5m_{min}$ от линии A или до тех пор, пока граница перехода не повернет назад на расстояние не менее $0{,}5m_{min}$. Возвращаются в ближайшую точку границы перехода, находящуюся на расстоянии по евклидовой дистанции, большем или равным m_{min} от последней точки границы перехода вдоль линии границы перехода, и сохраняют ее как конечную точку границы перехода. Эту точку границы перехода следует рассматривать как одно из предполагаемых положений границы внешнего края L-образных границ;
      V) вычисляют новую откорректированную линию A1, которая является «наиболее приближенной» линией для границы перехода, определенной на двух предыдущих этапах. «Наиболее приближенную» линию вычисляют с использованием алгоритма линейной регрессии (используя конечные точки для выбора зависимой оси, т. е. если они ближе к горизонтальной оси, зависимая ось — ось *x*) для каждой точки. На «наиболее приближенной» прямой линии отмечают отрезок, ограниченный точками p1 и p2, которые являются ближайшими к найденным выше конечным точкам границы перехода;

VI) сохраняют две конечные точки отрезка линии A1 — p1 и p2. Так же сохраняют значение цвета левой стороны края границы перехода, видимое при движении от p1 к p2;

iv) если этап, указанный в перечислении iii), закончился неудачей или невозможно продолжить движение вниз на $3{,}5m_{min}$ на этапе, указанном в перечислении iii) IV), проверяют, достигнута ли снизу граница перехода на расстоянии $3{,}5m_{min}$ от точки старта. Если да, повторяют операции этапа, указанные в перечислении iii), но не при движении вверх, а при движении вниз;

v) если ни один из этапов, указанных в перечислениях iii) или iv), не оказался успешным, проверяют, находятся ли верхняя и нижняя границы перехода на расстоянии не менее $2m_{min}$ от точки старта. Если достигнуты верхняя и нижняя границы перехода, то включают в формируемую границу перехода отрезки вверх и вниз на расстоянии $2m_{min}$ и повторяют операции этапа, указанные в перечислении iii), но с добавлением границы перехода;

vi) повторяют вышеуказанный процесс для следующей точки перехода на линии сканирования, начиная с этапа, указанного в перечислении i), до достижения края изображения;

2) продлевают линию сканирования вертикально в обоих направлениях от центральной точки изображения. Находят линию отрезков с использованием той же логической процедуры, что и на этапе, указанном в перечислении 1), одновременно следуя от каждой границы перехода символа влево, а затем вправо;

3) среди сохраненных отрезков линий A1 осуществляют поиск пар отрезков, удовлетворяющих четырем критериям:

i) если две линии имеют одинаковые направления p1 к p2, проверяют, чтобы ближайшее из расстояний между линиями p1 к p2 было менее g_{max}. Если две линии имеют противоположные направления p1 к p2, проверяют, чтобы ближайшее из расстояний между линиями p1 к p1 или p2 к p2 было менее g_{max};

ii) подтверждают, что две вышеуказанных линии являются параллельными с отклонением не более 5°;

iii) подтверждают, что две вышеуказанных линии имеют одинаковый сохраненный цвет, если их направления от p1 к p2 совпадают, или что сохраненные цвета противоположны, если их направления от p1 до p2 противоположны друг другу;

iv) формируют две временные линии, продолжая каждую из двух рассматриваемых линий по достижению на их продолжении точки, ближайшей к конечной точке отрезка другой линии. Проверяют, чтобы обе временные линии были отделены менее чем на $0{,}5m_{min}$ от любой иной точки между двумя продолженными линиями;

4) для каждой пары линий, соответствующих требованиям критерия, указанного в перечислении 3), заменяют эту пару отрезков линий на один удлиненный отрезок линии A1 путем выбора «наиболее приближенной» линии по четырем конечным точкам пары рассматриваемых коротких отрезков линий. Также сохраняют значение цвета левой стороны границы перехода новой удлиненной линии, рассматриваемой от конечной точки p1 до конечной точки p2;

5) повторяют этапы, указанные в перечислениях 3) и 4), до тех пор, пока не останется ни одной пары линий A1, которые можно будет скомбинировать;

6) выбирают отрезки линии, длиннее d_{min}. Помечают эти линии как предполагаемые L-образные стороны;

7) находят среди полученных пар предполагаемых L-образных сторон две линии, которые соответствуют трем критериям;

i) подтверждают, что ближайшие точки каждой линии находятся друг от друга на расстоянии менее $1{,}5g_{max}$;

ii) подтверждают, что эти две линии должны были взаимно перпендикулярны с погрешностью до 5°;

iii) подтверждают, что внутри L-образной структуры, образованной двумя линиями, присутствует один и тот же сохраненный цвет. Следует иметь ввиду, что если одна или обе линии простираются в обе стороны от точки их пересечения, то два или четыре образованных L-образных шаблона должны быть проверены на соответствие цвету и минимальной длине d_{min} для укороченной стороны или сторон, прежде чем они могут стать предполагаемыми L-образными сторонами;

8) для каждой пары линий — предполагаемых L-образных сторон, найденных на этапе, указанном в перечислении 7), формируют предполагаемую L-образную структуру путем продолжения отрезков до точки их пересечения;

9) если предполагаемая L-образная структура была сформирована из отрезков линий белого цвета внутри угла L-образной структуры, формируют инвертированное по цвету изображение для декодирования. Предпринимают попытки декодировать символ, начиная соответствующего нормального или инвертированного изображения, выбирая в качестве начального этап, указанный в перечислении d), используя каждую предполагаемую L-образную структуру, определенную на этапе, указанном в перечислении 8), как L-образную часть шаблона поиска. Если декодирование не удалось выполнить, переходят к этапу, указанному в перечислении c);

c) продолжают подбирать отрезки линий A1 и стороны предполагаемой L-образной структуры, полученных из предыдущих этапов, также продолжают поиски предполагаемых L-образных структур, используя горизонтальное и вертикальное смещение линий сканирования по отношению к предыдущим линиям сканирования:

1) используя новую горизонтальную линию сканирования, проведенную на расстоянии $3m_{min}$ выше от центральной горизонтальной линии сканирования, повторяют в том же порядке действия этапа, указанного в перечислении b)1), исключая действия, при которых процесс начинается из центральной точки изображения, и этапы, указанные в перечислениях от b)3) до b)9). Если декодирование не удалось выполнить, переходят к следующему этапу, указанному в перечислении 2);
2) используя новую вертикальную линию сканирования, проведенную на расстоянии $3m_{min}$ слева от центральной вертикальной линии сканирования, повторяют действия этапа, указанного в перечислении b)2), исключая действия, при которых процесс начинается из центральной точки изображения, и этапы, указанные в перечислениях от b)3) до b)9). Если декодирование не удалось выполнить, переходят к следующему этапу, указанному в перечислении 3);
3) повторяют действия этапа, указанного в перечислении 1), используя новую горизонтальную линию сканирования, расположенную на расстоянии $3m_{min}$ ниже центральной горизонтальной линии сканирования. Если декодирование не удалось выполнить, повторяют действия этапа, указанного в перечислении 2), но со сдвигом новой вертикальной линии сканирования на $3m_{min}$ вправо от центральной вертикальной линии сканирования. Если декодирование не удалось выполнить, переходят к этапу, указанному в перечислении 4);
4) продолжают обрабатывать горизонтальные и вертикальные линии сканирования, как это предусмотрено на этапах, указанных в перечислениях 1) — 3), на $3m_{min}$ вверх, затем влево, затем вниз, затем от ранее произведенных линий сканирования до успешного декодирования символа или до достижения края изображения;

d) первоначально считают, что область-кандидат содержит символ квадратной формы. Если декодировать область как символ квадратной формы не удается, пытаются найти и декодировать символ прямоугольной формы, начиная с этапа, указанного в перечислении j). Для символа квадратной формы сначала формируют нормализованную схему переходов для равных сторон области-кандидата, чтобы найти часть шаблона поиска с чередующимися модулями (alternating module finder pattern):

1) проецируют через область-кандидат линию, делящую пополам внутренний угол, образованный сторонами L-образной структуры, как показано на рисунке 5. Определяют две равные области, образованные этой разделительной линией (биссектрисой), как левую и правую области со стороны угла L-образной структуры;
2) для каждой стороны формируют так называемую линию поиска, расположенную на расстоянии d_{min} от вершины угла L-образной структуры и параллельную другой ее стороне, и продолжают эту линию до биссектрисы согласно рисунку 5;
3) сдвигают каждую линию поиска от вершины угла L-образной структуры согласно рисунку 5, удлиняя каждую линию поиска, чтобы они всегда начинались от стороны угла L-образной структуры и заканчивались на биссектрисе, сохраняя линии поиска параллельными противоположным сторонам угла L-образной структуры. По мере перемещения каждой стороны на размер пикселя изображения подсчитывают число переходов черный/белый и белый/черный, начиная и заканчивая подсчет переходами от цвета стороны L-образной структуры к противоположному цвету. Переход от одного цвета к другому подлежит подсчету только тогда, когда на двух соседних параллельных линиях поиска (предыдущей и следующей) в соответствующей позиции обнаруживается переход того же типа (от черного к белому или от белого к черному). Осуществляют построение графика числа переходов, умноженного на длину самой длинной стороны L-образной структуры, поделенную на текущую длину линии поиска, измеренную между двумя ограничивающими линиями, как указано в формуле

$$T = N \cdot L_{max}/L_s, \quad (2)$$

где N — число переходов;

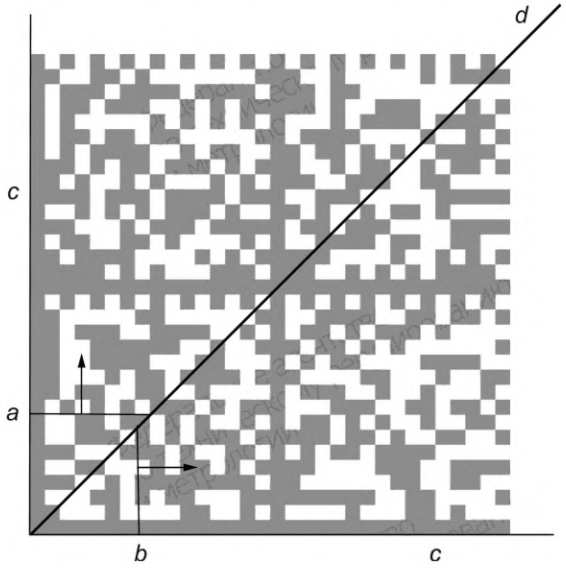
L_{max} — максимальная длина стороны L-образной структуры;

L_s — длина линии поиска;

T — нормализованное число переходов.

Эта формула нормализует значение T, предупреждая его увеличение по мере увеличения длины линии поиска.

Продолжают вычислять значения T до тех пор, пока линия поиска не будет длиннее наибольшей оси предполагаемой области-кандидата символа на 50 %;



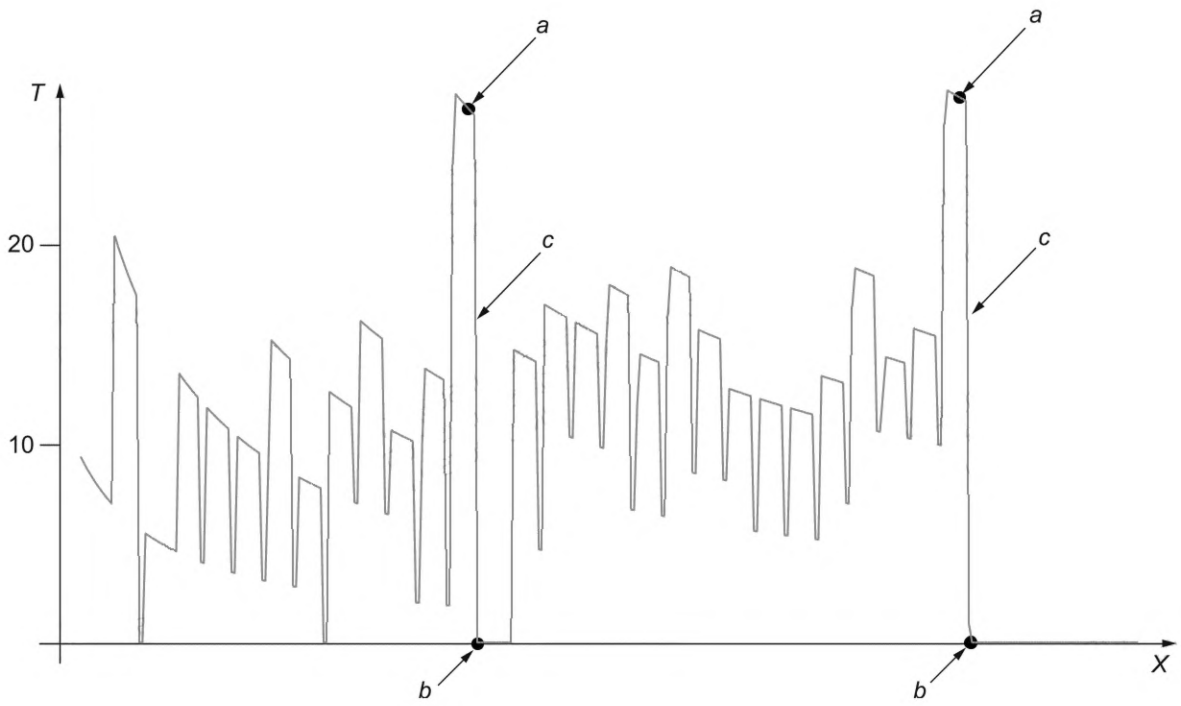
a — левая линия поиска; *b* — правая линия поиска; *c* — сторона L-образной структуры; *d* — биссектриса угла

Рисунок 5 — Направления движения линий поиска

4) строят график зависимости значений T для каждой стороны, где на оси ординат Y указано значение T, а на оси абсцисс X — расстояние линии поиска от вершины угла L-образной структуры. Пример графика приведен на рисунке 6;

5) начиная со значения T с наименьшим расстоянием X на графике для правой стороны и затем увеличивая расстояние X, находят первый экземпляр значения T_S (T_S = максимум нуля и $T - 1$), которое на 25 % менее предыдущего локального максимального значения T, при условии, что значение T более 1. Увеличивают это значение расстояния X до тех пор, пока число переходов не перестанет уменьшаться. Если число переходов не увеличивается, увеличивают значение X еще раз. Называют это значение X значением для впадины. Увеличивают значение X локального максимума до тех пор, пока число переходов не уменьшится, и называют это значение X значением для пика. Называют среднее значение значений X для пика и впадины значением X нисходящей линии. Линия поиска в точке пика может соответствовать стороне шаблона поиска с чередующимися модулями (alternating finder pattern side). Линия поиска во впадине может соответствовать внутренней части однородной темной линии или светлой свободной зоне;

6) находят пик и впадину на графике для левой стороны, значение нисходящей линии X которых в наибольшей степени соответствует значению нисходящей линии X для правого пика и впадины. Если возвращаются к этому этапу с более позднего этапа, рассматривают дополнительные левые пики и впадины, упорядоченные по степени их соответствия правому пику и впадине. Однако, любой рассматриваемый левый пик и впадина должны быть проверены, чтобы убедиться в том, что абсолютная величина разности между значениями X для правого и левого пика составляет менее 15 % от среднего значения значений X для двух пиков и что абсолютная величина разности между значениями X для правой и левой впадин составляет менее 15 % от среднего значения X для двух впадин. Значение, равное 15 %, соответствует максимально разрешенному сокращению;



X — расстояние от вершины угла L-образной структуры; *T* — значения *T*; *a* — пик; *b* — впадина; *c* — предполагаемый переход «пик-впадина»

Рисунок 6 — Пример графика зависимости *T* от увеличения длины линии поиска

7) линия поиска, соответствующая впадине на графике для правой стороны, линия поиска, соответствующая впадине на графике для левой стороны, и две стороны угла L-образной структуры очерчивают возможную область данных символа Data Matrix. Проводят обработку этой области данных согласно этапу, указанному в перечислении e). Если декодирование не удалось, находят следующий левый пик и впадину согласно действиям этапа, приведенного в перечислении d)6). После того, как все левые пики и впадины будут забракованы, отбрасывают правый пик и впадину и продолжают поиск, начиная с этапа, указанного в перечислении d)5), для нахождения следующего правого пика и впадины;

e) для каждой из двух сторон шаблона с чередующимися модулями находят линию, проходящую через центры чередующихся темных и светлых модулей, в следующей последовательности:

1) для каждой стороны формируют прямоугольную область, ограниченную линиями поиска для пика и впадины в качестве двух длинных сторон прямоугольной области и стороной L-образной структуры и линией для впадины для другой стороны в качестве коротких сторон прямоугольной области, как приведено на рисунке 7;

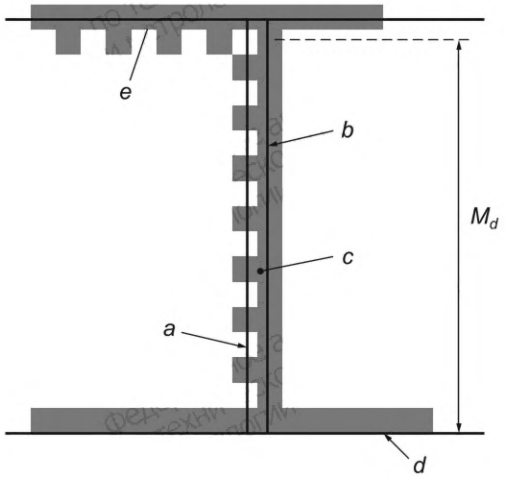
2) в пределах этой прямоугольной области, находят линии в наибольшей степени подходящие для внешних границ в следующей последовательности:

i) проходят все проверочные линии, начиная с линии для впадины и параллельно ей, и продолжая до достижения линии для пика, осуществляют поиск переходов к противоположному цвету, обычно перпендикулярных к проверочной линии. Выбирают только переходы, которые являются переходами либо от темного к светлому, либо от светлому к темному, в зависимости от того какой первоначальный цвет совпадает с преобладающим цветом изображения вдоль линии для впадины;

ii) создают график, который отображает расстояние от линии для пика до первого перехода, найденного в каждой позиции по линиям поиска. Если переход не найден, используют ноль, чтобы пометить расстояние.

Для каждого отрезка этого графика, ограниченного нулем с обеих сторон:

a) находят точку на указанном графике, которая находится в центре данного отрезка и точкой границы, соответствующей этой точке на графике;



a — линия для пика; *b* — линия для впадины; *c* — прямоугольная область; *d* — граница L-образной структуры; *e* — линия для впадины с другой стороны

Рисунок 7 — Построение прямоугольной области

b) находят все точки переходов, которые находятся в пределах области, ограниченной радиусом R с центром в точке на границе, где величина R соответствует 25 % расстояния между двумя точками с нулевыми значениями на указанном графике;

c) собирают данные по всем точкам и продолжают действия применительно к следующему отрезку;

iii) вычисляют с применением линейной регрессии наиболее приближенную линию, используя данные по собранным ранее точкам. Эта линия должна проходить вдоль внешней стороны шаблона чередующихся модулей (указана на рисунке 8 как наиболее приближенная линия);

3) для каждой стороны осуществляют построение линии, параллельной линии, найденной на этапе e)2), которая смещена в направлении угла L-образной структуры на перпендикулярное расстояние от угла L-образной структуры до линии поиска для пика, деленное на удвоенное число переходов в линии поиска для пика плюс один, согласно формуле

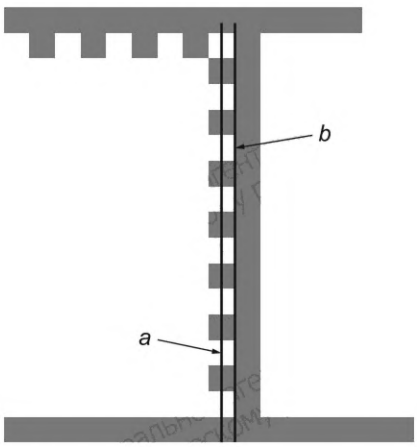
$$O = d / ((n + 1) \cdot 2), \quad (3)$$

где O — смещение;

d — длина линии для пика;

n — число переходов.

Каждая из двух построенных линий должна соответствовать линии центров модулей шаблона чередующихся модулей на такой стороне, см. рисунок 8;



a — линия центров модулей шаблона чередующихся модулей; *b* — наиболее приближенная линия

Рисунок 8 — Линия центров модулей шаблона чередующихся модулей

f) для каждой стороны измеряют расстояния от края до подобного края в шаблоне чередующихся модулей в следующей последовательности:

1) ограничивают линию, проходящую через центры модулей (среднюю линию) шаблона чередующихся модулей, сформированную на этапе e)3), с одной стороны границей L-образной структуры и с другой стороны линией, проходящей через центры модулей другого шаблона чередующихся модулей, определенной на этапе e)3). Длину этой линии обозначают *M*d (см. рисунок 7);
2) вдоль ограниченной средней линии измеряют расстояния от края до края между всеми подобными краями всех пар, состоящих из двух элементов, т. е. пар элементов темный/светлый и светлый/темный. Начинают и завершают измерения от края до подобного края с переходом краев от цвета L-образной структуры к противоположному цвету;
3) вычисляют среднеарифметическое значение всех измеренных расстояний от края до подобного края и устанавливают текущую оценку расстояния от края до подобного края EE_Dist как искомое значение;
4) бракуют все пары элементов, у которых измеренные расстояния от края до подобного края отличаются более чем на 25 % от значения EE_Dist;

g) для каждой стороны находят центральные точки модулей в шаблоне чередующихся модулей;

1) используя измеренные расстояния оставшихся (незабракованных) пар элементов на этапе, указанном в перечислении f)4), вычисляют среднее приращение/сокращение краски при печати (по вертикали или горизонтали в зависимости от стороны сегмента) как среднее приращение/сокращение краски при печати для пары элементов, в которой штрих — это ширина темного элемента, а пробел — ширина светлого элемента в оставшейся паре элементов) согласно формуле

$$i = \text{Average}\left((w_b - ((w_b + w_s) / 2)) / ((w_b + w_s) / 2)\right), \quad (4)$$

где i — среднее приращение/сокращение краски при печати;
w_b — ширина темного элемента (штриха);
w_s — ширина светлого элемента (пробела);

2) вычисляют центр темного элемента (штриха) в паре элементов, занимающих среднее положение, используя следующее смещение по формуле (5) в сторону темного элемента (штриха) от внешнего края темного элемента (штриха) в паре элементов, занимающих среднее положение:

$$o = (EE_Dist \cdot (1 + i)) / 4, \quad (5)$$

где o — смещение;
i — среднее приращение/сокращение краски при печати.

Если присутствует более одной пары элементов, занимающих среднее положение, выбирают одну пару в соответствии со следующей процедурой:

i) располагают края (за исключением края L-образной структуры шаблона поиска) на их расстоянии от края L-образной структуры шаблона поиска. Присутствует нечетное число этих краев, потому что края начинаются и заканчиваются на переходе от темного элемента к светлому в направлении от L-образной структуры шаблона поиска;
ii) обращаются к краю, занимающему среднее положение, в списке центрального края;
iii) вычисляют (нечетное число) расстояний от края до подобного края пар элементов и находят их среднее положение *EE_Dist*;
iv) выбирают одну или несколько пар элементов с длиной *EE_Dist*;
v) среди этих пар определяют одну или две пары краев элементов, которые имеют край, ближайший к центральному краю;
vi) если по прежнему возникает препятствие, выбирают пару элементов, у которых внешний край темного элемента (штриха) является ближайшим к центральному краю;
vii) если по прежнему возникает препятствие, выбирают пару элементов, имеющих внутренний край, который является ближайшим к L-образной структуре шаблона поиска;

3) начиная от центра темного элемента (штриха) пары элементов, занимающих среднее положение, из этапа, указанного в перечислении f)3), и продолжая процесс в направлении светлого элемента (пробела) из пары элементов, до конца ограниченной средней линии, вычисляют центр каждого элемента, выделенного на рисунке 9 серым цветом, с выполнением следующих действий.

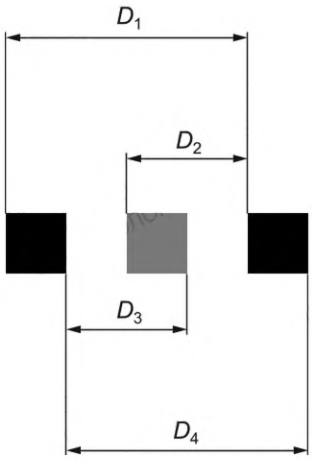


Рисунок 9 — Измерения расстояний от края до подобного края для определения центра каждого элемента

На рисунке 9 показаны три темных (штриха) и два светлых элемента (пробела). Если элемент, центр которого вычисляют, является светлым (пробелом), то на схеме должны быть представлены три светлых элемента (пробела) вместо темных элементов (штрихов) и два темных элемента (штриха) вместо светлых (пробелов). Для светлых элементов, прилегающих к элементу в конце средней линии, некоторые из измерений от D_1 до D_4 опущены, поскольку они выпадают за пределы измеримых краев элементов символа или сегмента;

i) вычисляют точку p1, находящуюся на средней линии на расстоянии EE_Dist/2 от предыдущего вычисленного центра элемента в направлении нового элемента;

ii) вычисляют значения от d_1 до d_4, где:

$d_1 = D_1 / 2$,
$d_2 = D_2$,
$d_3 = D_3$,
$d_4 = D_4 / 2$;

iii) если одно из значений d_1 — d_4 находится в пределах 25 % от EE_Dist, выбирают одно из значений, ближайшее к EE_Dist, и устанавливают новое значение EE_Dist как среднее между текущим значением EE_Dist и указанным выбранным значением из диапазона d_1 — d_4.

I) если выбрано значение d_1 или d_4, определяют соответствующий край D_1 или D_4, ближайший к элементу, центр которого необходимо вычислить. Сдвигают этот край на расстояние $(i/2) \cdot (EE_Dist/2)$ в соответствующем направлении [т. е. если приращение ширины темного элемента (штриха) i положительная величина, смещение края должно быть в сторону светлого элемента (пробела), заключенного в пределах значений D_1 или D_4, и, если отрицательная величина, смещение должно быть в противоположную сторону от светлого элемента]. Вычисляют точку p2, находящуюся на средней линии на расстоянии 0,75 выбранного значения d_1 или d_4 от этого смещенного края в сторону элемента, центр которого должен быть вычислен;

II) если выбрано значение d_2 или d_3, определяют соответствующий край D_2 или D_3, ближайший к элементу, центр которого необходимо вычислить. Сдвигают этот край на расстояние $(i/2) \cdot$ (EE_Dist/2) в соответствующем направлении [т. е. если приращение ширины темного элемента (штриха) при печати является положительной величиной, смещение края должно быть проведено в сторону светлого элемента (пробела), заключенного в пределах значений D_2 или D_3, и, если отрицательной величиной, смещение должно быть в противоположную сторону от светлого элемента]. Вычисляют точку p2, находящуюся на средней линии на расстоянии 0,25 выбранного значения d_2 или d_3 от смещенного края в сторону элемента, центр которого следует вычислить;

III) считают, что центр элемента находится точно посередине между точками p1 и p2;

iv) в противном случае, если ни одно из значений d_1, d_2, d_3, d_4 не находится в пределах 25 % EE_Dist, оставляют текущее значение EE_Dist, используют p1 как центр нового элемента и переходят к определению следующего элемента;

4) чтобы вычислить окончательный центр элемента, продвигаются из текущего центра элемента на величину в $EE_Dist / 2$;

5) начиная с темного элемента в паре элементов, занимающих среднее положение, и продолжая в противоположном направлении по отношению к определенному на этапе, указанном в перечислении 3), вплоть до окончания ограниченной средней линии, вычисляют центры каждого элемента, используя порядок действий, установленный для этапа, указанного в перечислении для этапа 3);

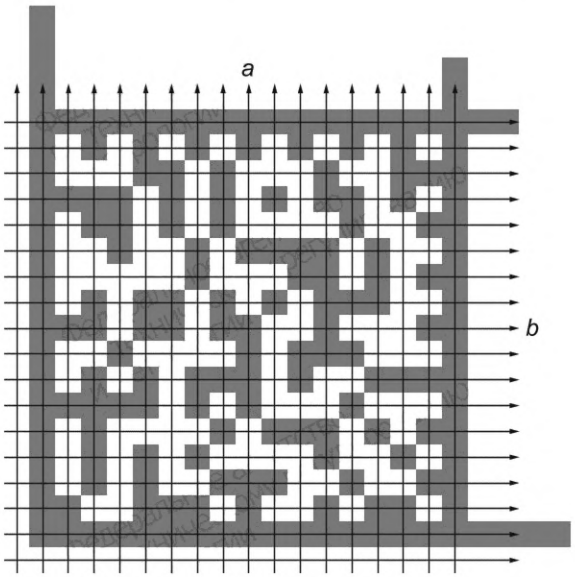
6) чтобы вычислить окончательный центр элемента, продвигаются из текущего центра элемента на величину в *EE_Dist* / 2;

h) если число модулей на каждой стороне не соответствует надлежащей первой области, продолжают поиски, начиная с этапа, указанного в перечислении d)6), для следующего левых пика и впадины. В противном случае осуществляют построение сетки выборки модулей данных в области данных, охватывая центры модулей шаблона чередующихся модулей следующим образом:

1) для каждой стороны продолжают каждую линию, построенную на этапе, указанном в перечислении e)3), и линию противоположной стороны L-образной структуры, для формирования точки схода двух почти параллельных линий или параллельных продленных линий;

2) из каждой точки схода проводят лучи, проходящие через центры модулей, построенные на этапе, указанном в перечислении g), в направлении, близком к перпендикулярному, к линии, полученной на этапе, указанном в перечислении e)3);

3) точки пересечения этих двух направлений лучей, близких к перпендикулярным, должны соответствовать центрам модулей данных в области данных как приведено на рисунке 10;



a — вверх от стороны точки схода; *b* — вправо от стороны точки схода

Рисунок 10 — Формирование сетки выборки модулей

i) продолжают заполнение остальных областей данных:

1) в процессе составления области данных формируют новую L-образную структуру для следующей части данных левее или выше, используя одну из двух следующих процедур;

i) если новая область данных по-прежнему ограничена с одной стороны исходной L-образной структурой, полученной на этапе, указанном в перечислении b), повторяют этап, указанный в перечислении c), устанавливая новую область данных и используя множество точек, выбранных на этапе, указанном в перечислении e)2), и множество точек на стороне L-образной структуры из этапа, указанного в перечислении b)2), которые находятся за пределами линии, полученной на этапе, указанной в перечислении e)2);

ii) если новая область данных ограничена с двух сторон другими областями данных, повторяют порядок действий с этапа, указанного в перечислении c), для определения новой области данных с помощью множества точек, выбранных на этапе, указанном в перечислении e)2), для каждой области данных, которая примыкает и ограничивает новую область данных с двух сторон;

2) если область данных не соответствует по числу модулей ранее полученным областям данных, символ корректируют путем его уменьшения до ближайшего большего числа областей, допускаемых правилами для символа;
3) декодируют символ с одной или несколькими областями данных, начиная с последовательности действий, установленных на этапе, указанном в перечислении k);
4) если в текущей области данных будут полностью исчерпаны пики и впадины, следует вернуться к предыдущей области данных и продолжить поиск с этапа, указанного в перечислении d)6), для следующего левого пика и впадины в этой области данных;

j) находят части данных прямоугольного символа:
1) проверяют длины обеих сторон L-образной структуры как предполагаемой пары и устанавливают одну из них как «более длинную», а другую как «более короткую» в рамках указанной предполагаемой пары;
2) для каждой стороны L-образной структуры передвигают линию, перпендикулярную к этой стороне, осуществляя сканирование по длине другой стороны L-образной структуры. Сохраняют параллельность каждой линии поиска линии другой стороны L-образной» структуры. По мере того, как происходит перемещение по каждой стороне на размер пикселя изображения, подсчитывают число переходов черный/белый и белый/черный, начиная и завершая подсчет переходами от цвета стороны L-образной структуры к противоположному цвету. Переход от одного цвета к другому будет учитываться только тогда, когда текущая линия поиска, а также линии поиска непосредственно выше и ниже имеют одинаковый цвет, противоположный ранее учтенному цвету перехода. По мере перемещения по каждой стороне на пиксель, отмечают число переходов T;
3) продолжают, пока параллельная линия не переместится далее за перпендикулярную сторону L-образной структуры на 50 % для более короткой стороны или на 25 % для более длинной стороны;
4) начиная с начала графика, для каждого направления находят первый экземпляр значения T_S (T_S = максимум нуля и $T - 1$), которое менее 15 % от предыдущего локального максимального значения T, при условии, что значение T более 1. Увеличивают это значение X до тех пор, пока значение T не перестанет уменьшаться. Если значение T не увеличивается, увеличивают значение X еще раз. Приписывают это значение X к впадине. Увеличивают значение X локального максимума до тех пор, пока значение T не уменьшится, и приписывают это значение X к пику. Относят среднеарифметическое значений X пика и впадины к значению X нисходящей линии. Линия впадины в этой точке может представлять сторону символа или области данных;
5) находят линии шаблонов чередующихся модулей для каждой стороны области подобно процедуре для этапа, указанного в перечислении e);
6) составляют пробную сетку модулей для области данных или символа аналогично этапам, указанным в перечислениях f), g) и h);
7) если найденная область данных не является надлежащим символом прямоугольной формы, пытаются сформировать новую область данных, используя следующие далее действительные переходы пик/впадина;
8) составляют все дополнительные области данных как предусмотрено на этапе, указанном в перечислении i);
9) если установлена одна или несколько действительных областей данных, предпринимают попытки декодировать символ в соответствии с перечислениями k) и l). Если область(и) не является(ются) действительной(ыми), или декодирование не удалось выполнить, игнорируют эту область как предполагаемую действительную;

k) если число модулей данных является четным числом или символ имеет надлежащую прямоугольную форму, выполняют его декодирование, используя алгоритм исправления ошибок Рида—Соломона, в следующей последовательности:
1) определяют модули данных в предполагаемых центрах сетки. Темный узел в центре соответствует единице, светлый узел — нулю;
2) преобразуют группы по восемь модулей по определенным шаблонам кодовых слов в 8-битовые значения знаков символа;
3) выполняют процедуру исправления ошибок Рида—Соломона с полученными значениями знаков символа;

4) декодируют знаки символа в знаки данных в соответствии с установленными схемами декодирования.

Примечание 1 — Указанный рекомендуемый алгоритм декодирования был разработан без учета декодирования символов, составленных из несоприкасающихся точек (например, при точечном иглоударном нанесении). В этом случае рекомендуемый алгоритм декодирования не выполнит свое назначение. Приложения могут использовать методы, приведенные в ISO/IEC 29158, чтобы уменьшить это ограничение.

11 Рекомендации для пользователей

11.1 Интерпретация для визуального чтения

Поскольку символы Data Matrix могут содержать тысячи знаков, интерпретация для визуального чтения знаков данных в виде расположенного рядом обычного текста может оказаться неудобной. Как альтернатива, символ может сопровождать краткий описательный текст, а не весь закодированный. Размер знаков и тип шрифта не устанавливаются, и такое сообщение может быть напечатано в любом месте вблизи символа Data Matrix. Текст интерпретации для визуального чтения не должен накладываться ни на символ Data Matrix, ни на свободные зоны вокруг него.

11.2 Способность к автоматическому распознаванию

Символика Data Matrix может наряду с некоторыми другими символиками использоваться в среде автоматического распознавания (см. приложение K).

11.3 Системные подходы

Прикладные системы применения Data Matrix должны рассматриваться в качестве целостных системных решений (см. приложение L).

12 Передаваемые данные

12.1 Общие положения

Данный раздел описывает типовой протокол передачи данных для совместимых устройств считывания. Такие устройства считывания могут быть запрограммированы на поддержку иных вариантов передачи. Все закодированные знаки данных должны быть включены в передаваемые данные. Управляющие знаки символики и знаки исправления ошибок не передаются. Более сложные вопросы интерпретации данных рассмотрены далее в настоящем разделе.

12.2 Протокол для знака FNC1

Когда знак FNC1 находится в позиции первого знака символа (или в позиции пятого знака символа в первом символе последовательности структурированного соединения), это является признаком соответствия данных типовому формату идентификаторов применения GS1 (GS1 Application Identifier, AI) GS1 (см. [3]). Знак FNC1 в любой более последующей позиции такого символа выполняет функцию разделителя полей. Данный протокол должен обеспечивать передачу идентификатора символики. Первый знак FNC1 не должен быть представлен в передаваемых данных, хотя на его присутствие указывает использование в идентификаторе символики знака модификации со значением 2 (см. 12.6).

Когда знак FNC1 используют как разделитель полей, он должен быть представлен в передаваемом сообщении как управляющий знак $<^{G}_{S}>$ (знак набора знаков ASCII с десятичным значением 29).

12.3 Протокол для знака FNC1 во второй позиции

Когда знак FNC1 находится в позиции второго знака символа (или в позиции шестого знака символа в первом символе последовательности структурированного соединения), это является признаком того, что данные соответствуют определенному международному отраслевому стандартному формату. Указанный протокол должен обеспечивать передачу идентификатора символики. Первый знак FNC1 не должен быть представлен в передаваемых данных, хотя на его присутствие указывает использование соответствующего знака-модификатора со значением 3 в идентификаторе символики (см. 12.6).

Данные, закодированные в первом знаке символа, должны передаваться как обычно в начале данных. Когда знак FNC1 используется как разделитель полей, он должен быть представлен в передаваемом сообщении как управляющий знак $<^{G}_{S}>$ ASCII (КОИ-7) с десятичным значением 29.

12.4 Протокол для знаков Macro в первой позиции

Данный протокол используют для кодирования с уплотнением двух специальных головных и конечных меток сообщения в символах Data Matrix.

Когда знак Macro находится в первой позиции, должны передаваться вводная и заключительная части сообщений. Если первый знак символа содержит кодовое слово со значением 236 (т. е. закодированный знак Macro 05), то кодируемым данным должна предшествовать вводная часть $[)^{R}_{S}05^{G}_{S}$, за которой следуют данные. Если первый знак символа содержит кодовое слово со значением 237 (т. е. закодированный знак Macro 06), то кодируемым данным должна предшествовать вводная часть $[)^{R}_{S}06^{G}_{S}$, за которой следуют данные. В обоих случаях после данных должна быть передана заключительная часть $^{R}_{S}{}^{E}_{O}{}_{T}$.

12.5 Протокол для интерпретаций ECI

В системах, которые поддерживают интерпретации ECI, необходимо использовать префиксы идентификаторов символики при каждой передаче данных [1]. Если присутствует кодовое слово ECI, его следует передавать как управляющий знак с десятичным значением 92_{DEC} (или с шестнадцатеричным значением $5C_{HEX}$), представляемый знаком «\» (обратная дробная черта) в интерпретации, принимаемой по умолчанию. Следующие кодовые слова преобразуют в 6-разрядные значения в соответствии с правилами преобразования, обратными приведенным в таблице 6. Полученные 6-разрядные значения передают как соответствующие десятичные значения знаков набора знаков ASCII (от 48 до 57). Прикладное программное обеспечение после распознавания последовательности \nnnnnn должно интерпретировать все последующие знаки как знаки интерпретации ECI, установленной в соответствии с указанным 6-разрядным значением. Эта интерпретация действует до окончания кодируемых данных или до появления другой последовательности ECI. Если знак Обратная дробная черта (байт с десятичным значением 92_{DEC}) должен быть использован в кодируемых данных, то его передача должна осуществляться следующим образом. В случае, когда знак набора знаков ASCII (с десятичным значением 92_{DEC}) встречается как знак данных, должны быть переданы два байта с этим же значением, таким образом при одиночном применении знак действует как управляющий знак, а появление сдвоенных знаков свидетельствует о появлении знака данных.

Примеры
Закодированные данные A\\B\C
Передаваемые данные A\\\\B\\C

Использование идентификатора символики обеспечивает правильность интерпретации управляющего знака в данном прикладном применении. Исключение может быть сделано, если интерпретацию(и) ECI можно полностью обработать в устройстве считывания.

12.6 Идентификатор символики

ISO/IEC 15424 предусматривает типовую процедуру указания символики, которая была считана, наряду с набором вариантов, установленных в декодере, и специальными свойствами, которые могут быть включены в символ. После того как структура данных (включая использование любых интерпретаций ECI) установлена, декодер должен добавить соответствующий идентификатор символики в виде префикса к передаваемым данным. Идентификатор символики также необходим, если одна или несколько интерпретаций в расширенном канале (ECI) появляются в символе (если только не является действительным условие исключения, указанное в 12.5) или при использовании знака FNC1 в соответствии с 12.2 и 12.3. Идентификаторы символики и значения возможных вариантов, которые допустимы для использования, приведены в приложении H.

12.7 Пример передаваемых данных

В данном примере сообщение, состоящее из двух знаков «¶Ж» должно быть закодировано с помощью схемы кодирования ASCII. Знак «¶» представляют байтом с десятичным значением 182 в наборе знаков по умолчанию Data Matrix (номер назначения интерпретации ECI 000003 соответствует набору знаков по ISO/IEC 8859-1). Буква кирилловского алфавита «Ж» отсутствует в интерпретации ECI 000003, но представлена в ISO/IEC 8859-5 (см. [6]) (номер назначения интерпретации ECI 000007) байтом с тем же десятичным значением 182. Полное сообщение, следовательно, может быть представлено путем вставки переключения к интерпретации ECI 000007 после первого знака следующим образом. Символ кодирует

сообщение <¶> <переключение к интерпретации ECI 000007> <Ж>, используя следующую последовательность кодовых слов Data Matrix: [Upper Shift] [55] [ECI] [8] [Upper Shift] [55] с десятичными значениями кодовых слов [235], [55], [241], [8], [235], [55].

Примечание 1 — Знак верхнего регистра (Upper Shift) с последующим кодовым словом, имеющим значение 55, кодирует байт с десятичным значением 182.

Примечание 2 — Интерпретации ECI в символе Data Matrix кодируют как номер назначения ECI+1.

Декодер передает байты со следующими значениями (включая префикс идентификатора символики, вариант символики, знак-модификатор 4, указывающий на использование протокола интерпретаций ECI):

93, 100, 52, 182, 92, 48, 48, 48, 48, 48, 55, 182.

В графических знаках эта запись будет выглядеть следующим образом в интерпретации по умолчанию:]d4¶\000007

Декодер отвечает за передачу сигнала переключения к ECI 000007, но не за интерпретацию результата. Программное обеспечение с поддержкой ECI в приемной прикладной системе удалит управляющую последовательность ECI \000007, и буква «Ж» будет представлена способом, принятым в системе (т. е. путем изменения шрифта в файле распечатки). Конечным результатом будет соответствие исходному сообщению с буквой «Ж».

Приложение А
(обязательное)

Процесс чередования в символике Data Matrix

А.1 Пояснительная схема

Рассмотрим пример символа размером 72 × 72 модулей, для которого необходимо четыре уровня чередования, чтобы закодировать 368 кодовых слов данных и 144 кодовых слова исправления ошибок. Кодовые слова делят на четыре блока по 92 кодовых слова данных и 36 кодовых слов исправления ошибок с общей длиной каждого блока — 128 кодовых слов, см. рисунок А.1.

Поток кодовых слов	Кодовые слова данных *d*										Кодовые слова исправления ошибок ε									
	1	2	3	4	…	…	365	366	367	368	1	2	3	4	…	...	141	142	143	144
Блок 1	Кодовые слова данных *d*										Кодовые слова исправления ошибок ε									
	1	5	…	…	…	361	365				1	5	…	…		137	141			
Блок 2		Кодовые слова данных *d*										Кодовые слова исправления ошибок ε								
		2	6	…	…	…	362	366				2	6	…	…		138	142		
Блок 3			Кодовые слова данных *d*										Кодовые слова исправления ошибок ε							
			3	7	…	…	…	363	367				3	7	…	…		139	143	
Блок 4				Кодовые слова данных *d*										Кодовые слова исправления ошибок ε						
				4	8	…	…	…	364	368				4	8	…	…	…	140	144

Рисунок А.1 — Схема процесса чередования для символа размером 72 × 72

Символ размером 144 × 144 модулей является особенным в следствии того, что в нем присутствуют размеры двух блоков 156 модулей для блоков с 1 до 8 и 155 модулей для блоков 9 и 10. 1558 байтов кодовых слов данных и 620 байтов кодовых слов исправления ошибок распределены в 10 блоках как представлено на рисунке А.2.

Вследствии неправильного понимания предыдущей редакции этого стандарта некоторые реализации ошибочно начинают кодовые слова исправления ошибок с блока 9 вместо блока 1. Эти реализации должны следовать положениям настоящего раздела.

Поток кодовых слов	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	1	2	3	…	1556	1557	1558	1	2	3	…	619	620
Блок 1	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	1	11	21	…	1541	1551		1	11	21	…	601	611
Блок 2	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	2	12	22	…	1542	1552		2	12	22	…	602	612
Блок 3	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	3	13	23	…	1543	1553		3	13	23	…	603	613
Блок 4	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	4	14	24	…	1544	1554		4	14	24	…	604	614
Блок 5	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	5	15	25	…	1545	1555		5	15	25	…	605	615
Блок 6	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	6	16	26	…	1546	1556		6	16	26	…	606	616
Блок 7	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	7	17	27	…	1547	1557		7	17	27	…	607	617
Блок 8	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	8	18	28	…	1548	1558		8	18	28	…	608	618
Блок 9	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	9	19	29	…	1549			9	19	29	…	609	619
Блок 10	Кодовые слова данных *d*							Кодовые слова исправления ошибок ε					
	10	20	30	…	1550			10	20	30	…	610	620

Рисунок А.2 — Пояснение чередования для символа размером 144 × 144 модулей

А.2 Начальная последовательность для чередования в символах различных размеров

Последовательность чередующихся кодовых слов данных и кодовых слов исправления ошибок приведена в таблице А.1.

Таблица А.1 — Последовательность кодовых слов данных и кодовых слов исправления ошибок для символов разных размеров

Размер символа	Блок Рида—Соломона	Последовательность кодовых слов данных			Последовательность кодовых слов исправления ошибок		
52 × 52	1	1, 3, 5	...	201, 203	1, 3, 5	...	81, 83
	2	2, 4, 6	...	202, 204	2, 4, 6	...	82, 84
64 × 64	1	1, 3, 5	...	277, 279	1, 3, 5	...	109, 111
	2	2, 4, 6	...	278, 280	2, 4, 6	...	110, 112
72 × 72	1	1, 5, 9	...	361, 365	1, 5, 9	...	137, 141
	2	2, 6, 10	...	362, 366	2, 6, 10	...	138, 142
	3	3, 7, 11	...	363, 367	3, 7, 11	...	139, 143
	4	4, 8, 12	...	364, 368	4, 8, 12	...	140, 144
80 × 80	1	1, 5, 9	...	449, 453	1, 5, 9	...	185, 189
	2	2, 6, 10	...	450, 454	2, 6, 10	...	186, 190
	3	3, 7, 11	...	451, 455	3, 7, 11	...	187, 191
	4	4, 8, 12	...	452, 456	4, 8, 12	...	188, 192
88 × 88	1	1, 5, 9	...	569, 573	1, 5, 9	...	217, 221
	2	2, 6, 10	...	570, 574	2, 6, 10	...	218, 222
	3	3, 7, 11	...	571, 575	3, 7, 11	...	219, 223
	4	4, 8, 12	...	572, 576	4, 8, 12	...	220, 224
96 × 96	1	1, 5, 9	...	689, 693	1, 5, 9	...	265, 269
	2	2, 6, 10	...	690, 694	2, 6, 10	...	266, 270
	3	3, 7, 11	...	691, 695	3, 7, 11	...	267, 271
	4	4, 8, 12	...	692, 696	4, 8, 12	...	268, 272
104 × 104	1	1, 7, 13	...	805, 811	1, 7, 13	...	325, 331
	2	2, 8, 14	...	806, 812	2, 8, 14	...	326, 332
	3	3, 9, 15	...	807, 813	3, 9, 15	...	327, 333
	4	4, 10, 16	...	808, 814	4, 10, 16	...	328, 334
	5	5, 11, 17	...	809, 815	5, 11, 17	...	329, 335
	6	6, 12, 18	...	810, 816	6, 12, 18	...	330, 336
120 × 120	1	1, 7, 13	...	1039, 1045	1, 7, 13	...	397, 403
	2	2, 8, 14	...	1040, 1046	2, 8, 14	...	398, 404
	3	3, 9, 15	...	1041, 1047	3, 9, 15	...	399, 405
	4	4, 10, 16	...	1042, 1048	4, 10, 16	...	400, 406
	5	5, 11, 17	...	1043, 1049	5, 11, 17	...	401, 407
	6	6, 12, 18	...	1044, 1050	6, 12, 18	...	402, 408
132 × 132	1	1, 9, 17	...	1289, 1297	1, 9, 17	...	481, 489
	2	2, 10, 18	...	1290, 1298	2, 10, 18	...	482, 490

Окончание таблицы А.1

Размер символа	Блок Рида—Соломона	Последовательность кодовых слов данных			Последовательность кодовых слов исправления ошибок		
	3	3, 11, 19	...	1291, 1299	3, 11, 19	...	483, 491
	4	4, 12, 20	...	1292, 1300	4, 12, 20	...	484, 492
	5	5, 13, 21	...	1293, 1301	5, 13, 21	...	485, 493
	6	6, 14, 22	...	1294, 1302	6, 14, 22	...	486, 494
	7	7, 15, 23	...	1295, 1303	7, 15, 23	...	487, 495
	8	8, 16, 24	...	1296, 1304	8, 16, 24	...	488, 496
144 × 144	1	1, 11, 21	...	1541, 1551	1, 11, 21	...	601, 611
	2	2, 12, 22	...	1542, 1552	2, 12, 22	...	602, 612
	3	3, 13, 23	...	1543, 1553	3, 13, 23	...	603, 613
	4	4, 14, 24	...	1544, 1554	4, 14, 24	...	604, 614
	5	5, 15, 25	...	1545, 1555	5, 15, 25	...	605, 615
	6	6, 16, 26	...	1546, 1556	6, 16, 26	...	606, 616
	7	7, 17, 27	...	1547, 1557	7, 17, 27	...	607, 617
	8	8, 18, 28	...	1548, 1558	8, 18, 28	...	608, 618
	9	9, 19, 29	...	1549	9, 19, 29	...	609, 619
	10	10, 20, 30	...	1550	10, 20, 30	...	610, 620

Приложение В
(обязательное)

Шаблонная рандомизация в символике Data Matrix

В.1 Общие положения

Алгоритмы шаблонной рандомизации преобразуют кодовое слово в заданной позиции на входе в новое рандомизированное (псевдослучайное) кодовое слово на выходе.

В.2 Алгоритм 253 состояний

В.2.1 Общие положения

Указанный алгоритм добавляет псевдослучайное число к значению кодового слова-заполнителя. Псевдослучайное число всегда будет в диапазоне от 1 до 253, а рандомизированное значение кодового слова-заполнителя будет в диапазоне от 1 до 254.

Переменная позиция кодового слова-заполнителя является номером кодового слова исходных данных от начала закодированных данных.

В.2.2 Алгоритм рандомизации 253 состояний

INPUT (Pad_codeword_value, Pad_codeword_position)

pseudo_random_number = ((149 * Pad_codeword_position) mod 253) + 1 temp_variable = Pad_codeword_value + pseudo_random_number

IF (temp_variable ≤ 254)

  OUTPUT (randomised_Pad_codeword_value = temp_variable)

ELSE

  OUTPUT (randomised_Pad_codeword_value = temp_variable − 254)

В.2.3 Алгоритм дерандомизации 253 состояний

INPUT (randomised_Pad_codeword_value, Pad_codeword_position)

pseudo_random_number = ((149 * Pad_codeword_position) mod 253) + 1

temp_variable = randomised_Pad_codeword_value − pseudo_random_number

IF (temp_variable ≥ 1)

  OUTPUT (Pad_codeword_value = temp_variable)

ELSE

  OUTPUT (Pad_codeword_value = temp_variable + 254)

В.3 Алгоритм 255 состояний

В.3.1 Общие положения

Указанный алгоритм добавляет псевдослучайное число к значению кодового слова в схеме кодирования по основанию 256. Псевдослучайное число всегда будет находиться в диапазоне от 1 до 255, а рандомизированное значение кодового слова в схеме кодирования по основанию 256 — в диапазоне от 0 до 255.

Переменная позиция кодового слова по основанию 256 (Base256_codeword_position) является номером кодового слова исходных данных от начала кодированных данных.

В.3.2 Алгоритм рандомизации 255 состояний

INPUT (Base256_codeword_value, Base256_codeword_position)

pseudo_random_number = ((149 * Base256_codeword_position) mod 255) + 1

temp_variable = Base256_codeword_value + pseudo_random_number

IF (temp_variable ≤ 255)

  OUTPUT (randomised_Base256_codeword_value = temp_variable)

ELSE

  OUTPUT (randomised_Base256_codeword_value = temp_variable − 256)

В.3.3 Алгоритм дерандомизации 255 состояний

INPUT (randomised_Base256_codeword_value, Base256_codeword_position)

pseudo_random_number = ((149 * Base256_codeword_position) mod 255) + 1

temp_variable=randomised_Base256_codeword_value − pseudo_random_number

IF (temp_variable ≥ 0)

  OUTPUT (Base256_codeword_value = temp_variable)

ELSE

  OUTPUT (Base256_codeword_value = temp_variable + 256)

Приложение С
(обязательное)

Наборы кодируемых знаков символики Data Matrix

Таблица С.1 — Набор знаков в схеме кодирования С40

Значение в схеме С40	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
0	Регистр 1 (Shift 1)		NUL	0	!	33	‘	96
1	Регистр 2 (Shift 2)		SOH	1	"	34	a	97
2	Регистр 3 (Shift 3)		STX	2	#	35	b	98
3	Пробел (Space)	32	ETX	3	$	36	c	99
4	0	48	EOT	4	%	37	d	100
5	1	49	ENQ	5	&	38	e	101
6	2	50	ACK	6	‘	39	f	102
7	3	51	BEL	7	(	40	g	103
8	4	52	BS	8	)	41	h	104
9	5	53	HT	9	*	42	i	105
10	6	54	LF	10	+	43	j	106
11	7	55	VT	11	,	44	k	107
12	8	56	FF	12	-	45	l	108
13	9	57	CR	13	.	46	m	109
14	A	65	SO	14	/	47	n	110
15	B	66	SI	15	:	58	o	111
16	C	67	DLE	16	;	59	p	112
17	D	68	DC1	17	<	60	q	113
18	E	69	DC2	18	=	61	r	114
19	F	70	DC3	19	>	62	s	115
20	G	71	DC4	20	?	63	t	116
21	H	72	NAK	21	@	64	u	117
22	I	73	SYN	22	[	91	v	118
23	J	74	ETB	23	\	92	w	119
24	K	75	CAN	24	]	93	x	120
25	L	76	EM	25	^	94	y	121
26	M	77	SUB	26	_	95	z	122

Окончание таблицы С.1

Значение в схеме С40	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
27	N	78	ESC	27	FNC1		{	123
28	O	79	FS	28			|	124
29	P	80	GS	29			}	125
30	Q	81	RS	30	Верхний регистр (Upper Shift)		~	126
31	R	82	US	31			DEL	127
32	S	83						
33	T	84						
34	U	85						
35	V	86						
36	W	87						
37	X	88						
38	Y	89						
39	Z	90						

Примечание — Соответствие между десятичным значением знака набора знаков ASCII и значением знака в схеме кодирования С40 остается неизменным вне зависимости от действующей интерпретации ECI.

Таблица С.2 — Набор знаков в схеме кодирования Text

Значение в схеме Text	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
0	Регистр 1 (Shift 1)	1	NUL	0	!	33	‘	96
1	Регистр 2 (Shift 2)	2	SOH	1	"	34	A	65
2	Регистр 3 (Shift 3)	3	STX	2	#	35	B	66
3	Пробел (Space)	32	ETX	3	$	36	C	67
4	0	48	EOT	4	%	37	D	68
5	1	49	ENQ	5	&	38	E	69
6	2	50	ACK	6	‘	39	F	70
7	3	51	BEL	7	(	40	G	71
8	4	52	BS	8	)	41	H	72
9	5	53	HT	9	*	42	I	73

Окончание таблицы С.2

Значение в схеме Text	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
10	6	54	LF	10	+	43	J	74
11	7	55	VT	11	,	44	K	75
12	8	56	FF	12	-	45	L	76
13	9	57	CR	13	.	46	M	77
14	a	97	SO	14	/	47	N	78
15	b	98	SI	15	:	58	O	79
16	c	99	DLE	16	;	59	P	80
17	d	100	DC1	17	<	60	Q	81
18	e	101	DC2	18	=	61	R	82
19	f	102	DC3	19	>	62	S	83
20	g	103	DC4	20	?	63	T	84
21	h	104	NAK	21	@	64	U	85
22	i	105	SYN	22	[	91	V	86
23	j	106	ETB	23	\	92	W	87
24	k	107	CAN	24	]	93	X	88
25	l	108	EM	25	^	94	Y	89
26	m	109	SUB	26	_	95	Z	90
27	n	110	ESC	27	FNC1		{	123
28	o	111	FS	28			|	124
29	p	112	GS	29			}	125
30	q	113	RS	30	Верхний регистр (Upper Shift)	Shift	~	126
31	r	114	US	31			DEL	127
32	s	115						
33	t	116						
34	u	117						
35	v	118						
36	w	119						
37	x	120						
38	y	121						
39	z	122						

Примечание — Соответствие между десятичным значением знака набора знаков ASCII и значением знака в схеме кодирования Text остается неизменным вне зависимости от действующей интерпретации ECI.

Таблица С.3 — Набор знаков в схеме кодирования EDIFACT

Знак данных			Двоичное значение по EDIFACT	Знак данных			Двоичное значение по EDIFACT
Знак	Десятичное значение	Двоичное значение		Знак	Десятичное значение	Двоичное значение	
@	64	01000000	000000	ПРОБЕЛ (space)	32	00100000	100000
A	65	01000001	000001	!	33	00100001	100001
B	66	01000010	000010	"	34	00100010	100010
C	67	01000011	000011	#	35	00100011	100011
D	68	01000100	000100	$	36	00100100	100100
E	69	01000101	000101	%	37	00100101	100101
F	70	01000110	000110	&	38	00100110	100110
G	71	01000111	000111	‘	39	00100111	100111
H	72	01001000	001000	(	40	00101000	101000
I	73	01001001	001001	)	41	00101001	101001
J	74	01001010	001010	*	42	00101010	101010
K	75	01001011	001011	+	43	00101011	101011
L	76	01001100	001100	,	44	00101100	101100
M	77	01001101	001101	-	45	00101101	101101
N	78	01001110	001110	.	46	00101110	101110
O	79	01001111	001111	/	47	00101111	101111
P	80	01010000	010000	0	48	00110000	110000
Q	81	01010001	010001	1	49	00110001	110001
R	82	01010010	010010	2	50	00110010	110010
S	83	01010011	010011	3	51	00110011	110011
T	84	01010100	010100	4	52	00110100	110100
U	85	01010101	010101	5	53	00110101	110101
V	86	01010110	010110	6	54	00110110	110110
W	87	01010111	010111	7	55	00110111	110111
X	88	01011000	011000	8	56	00111000	111000
Y	89	01011001	011001	9	57	00111001	111001
Z	90	01011010	011010	:	58	00111010	111010
[	91	01011011	011011	;	59	00111011	111011
\	92	01011100	011100	<	60	00111100	111100
]	93	01011101	011101	=	61	00111101	111101
^	94	01011110	011110	>	62	00111110	111110
Отказ от фиксации (Unlatch)		01011111	011111	?	63	00111111	111111

Примечание — Соответствие между десятичным значением знака набора знаков ASCII и значением по EDIFACT остается неизменным вне зависимости от действующей интерпретации ECI.

Приложение D
(обязательное)

Направляющие шаблоны символов Data Matrix

На рисунках D.1 — D.4 приведены примеры направляющих шаблонов.

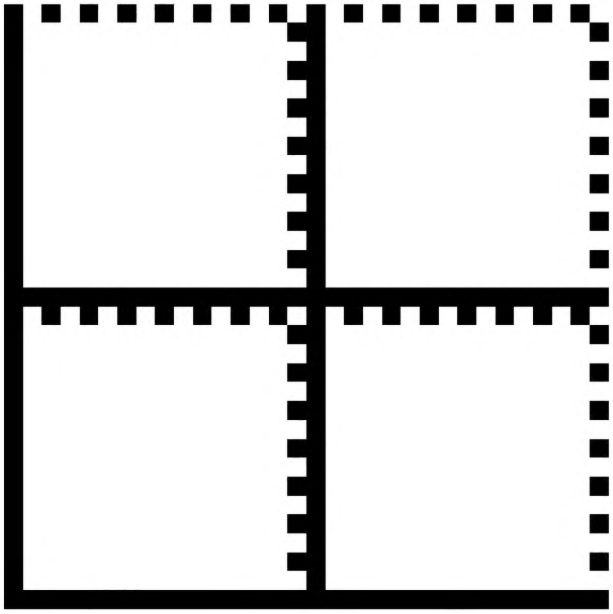


Рисунок D.1 — Конфигурация направляющего шаблона для символа квадратной формы 32 × 32 модуля

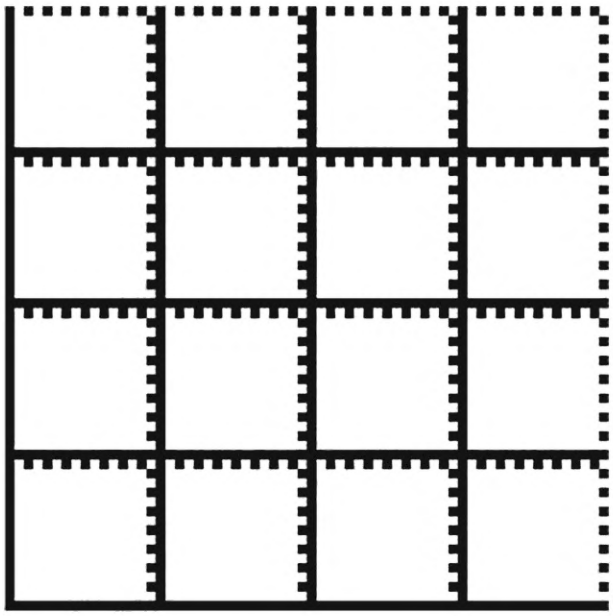


Рисунок D.2 — Конфигурация направляющего шаблона для символа квадратной формы 64 × 64 модуля

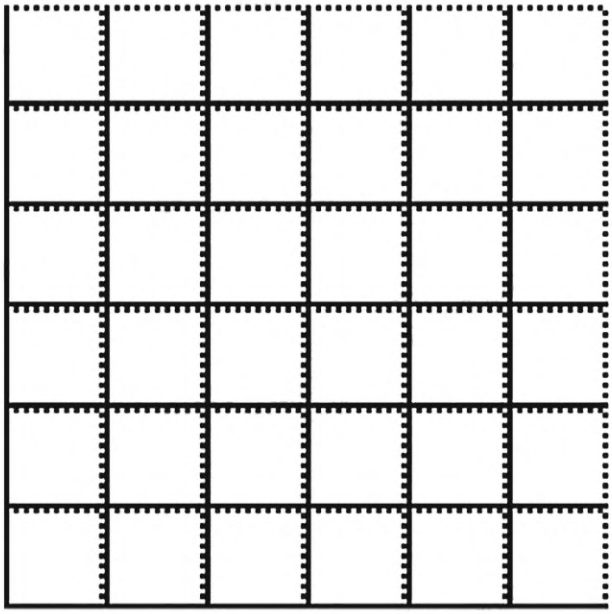


Рисунок D.3 — Конфигурация направляющего шаблона для символа квадратной формы 120 × 120 модулей

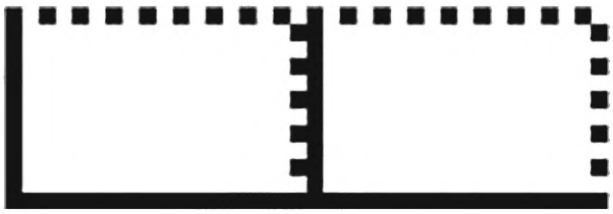


Рисунок D.4 — Конфигурация направляющего шаблона для символа прямоугольной формы 12 × 36 модулей

Приложение Е
(обязательное)

Алгоритм обнаружения и исправления ошибок Рида—Соломона для символики Data Matrix

Е.1 Порождающие полиномы для кодовых слов исправления ошибок

Кодовые слова исправления ошибок являются коэффициентами остатка от деления на порождающий полином $g(x)$ произведения полинома данных символа $d(x)$ на x^k. Каждый порождающий полином является произведением полиномов первой степени: $x - 2^1$, $x - 2^2$,..., $x - 2^n$, где n — показатель степени порождающего полинома.

Например, порождающий полином пятой степени представляет собой следующее:

$(x + 2)(x + 4)(x + 8)(x + 16)(x + 32)$

$= x^5 + (2 + 4 + 8 + 16 + 32)x^4 + [(2 * 4) + (2 * 8) + (2 * 16) + (2 * 32) + (4 * 8) + (4 * 16) + (4 * 32) + (8 * 16) + (8 * 32) + (16 * 32)]x^3 + [(2 * 4 * 8) + (2 * 4 * 16) + (2 * 4 * 32) + (2 * 8 * 16) + (2 * 8 * 32) + (2 * 16 * 32) + (4 * 8 * 16) + (4 * 8 * 32) + (4 * 16 * 32) + (8 * 16 *32)]x^2 + [(2 * 4 * 8 * 16) + (2 * 4 * 8 * 32) + (2 * 4 * 16 * 32) + (2 * 8 * 16 * 32) + (4 * 8 * 16 * 32)]x + (2 * 4 * 8 * 16 * 32)$

$= x^5 + 62x^4 + 111x^3 + 15x^2 + 48x + 228.$

Следует обратить внимание на то, что арифметика в этом поле Галуа не является обычной целочисленной арифметикой: операция «–» эквивалентна операции «+», которая представляет собой выполнение операции «исключающего ИЛИ» («exclusive-or») в этом поле, а операция умножения представляет собой побитовую операцию взятия по модулю 100101101 результата побитового умножения двух сомножителей.

Полином-делитель для порождения пяти проверочных знаков представляет собой следующее:

$g(x) = x^5 + 62x^4 + 111x^3 + 15x^2 + 48x + 228.$

Полином-делитель для порождения семи проверочных знаков представляет собой следующее:

$g(x) = x^7 + 254x^6 + 92x^5 + 240x^4 + 134x^3 + 144x^2 + 68x + 23.$

Полином-делитель для порождения 10 проверочных знаков представляет собой следующее:

$g(x) = x^{10} + 61x^9 + 110x^8 + 255x^7 + 116x^6 + 248x^5 + 223x^4 + 166x^3 + 185x^2 + 24x + 28.$

Полином-делитель для порождения 11 проверочных знаков представляет собой следующее:

$g(x) = x^{11} + 120x^{10} + 97x^9 + 60x^8 + 245x^7 + 39x^6 + 168x^5 + 194x^4 + 12x^3 + 205x^2 + 138x + 175.$

Полином-делитель для порождения 12 проверочных знаков представляет собой следующее:

$g(x) = x^{12} + 242x^{11} + 100x^{10} + 178x^9 + 97x^8 + 213x^7 + 142x^6 + 42x^5 + 61x^4 + 91x^3 + 158x^2 + 153x + 41.$

Полином-делитель для порождения 14 проверочных знаков представляет собой следующее:

$g(x) = x^{14} + 185x^{13} + 83x^{12} + 186x^{11} + 18x^{10} + 45x^9 + 138x^8 + 119x^7 + 157x^6 + 9x^5 + 95x^4 + 252x^3 + 192x^2 + 97x + 156.$

Полином-делитель для порождения 18 проверочных знаков представляет собой следующее:

$g(x) = x^{18} + 188x^{17} + 90x^{16} + 48x^{15} + 225x^{14} + 254x^{13} + 94x^{12} + 129x^{11} + 109x^{10} + 213x^9 + 241x^8 + 61x^7 + 66x^6 + 75x^5 + 188x^4 + 39x^3 + 100x^2 + 195x + 83.$

Полином-делитель для порождения 20 проверочных знаков представляет собой следующее:

$g(x) = x^{20} + 172x^{19} + 186x^{18} + 174x^{17} + 27x^{16} + 82x^{15} + 108x^{14} + 79x^{13} + 253x^{12} + 145x^{11} + 153x^{10} + 160x^9 + 188x^8 + 2x^7 + 168x^6 + 71x^5 + 233x^4 + 9x^3 + 244x^2 + 195x + 15.$

Полином-делитель для порождения 24 проверочных знаков представляет собой следующее:

$g(x) = x^{24} + 193x^{23} + 50x^{22} + 96x^{21} + 184x^{20} + 181x^{19} + 12x^{18} + 124x^{17} + 254x^{16} + 172x^{15} + 5x^{14} + 21x^{13} + 155x^{12} + 223x^{11} + 251x^{10} + 197x^9 + 155x^8 + 21x^7 + 176x^6 + 39x^5 + 109x^4 + 205x^3 + 88x^2 + 190x + 52.$

Полином-делитель для порождения 28 проверочных знаков представляет собой следующее:

$g(x) = x^{28} + 255x^{27} + 93x^{26} + 168x^{25} + 233x^{24} + 151x^{23} + 120x^{22} + 136x^{21} + 141x^{20} + 213x^{19} + 110x^{18} + 138x^{17} + 17x^{16} + 121x^{15} + 249x^{14} + 34x^{13} + 75x^{12} + 53x^{11} + 170x^{10} + 151x^9 + 37x^8 + 174x^7 + 103x^6 + 96x^5 + 71x^4 + 97x^3 + 43x^2 + 231x + 211.$

Полином-делитель для порождения 36 проверочных знаков представляет собой следующее:

$g(x) = x^{36} + 112x^{35} + 81x^{34} + 98x^{33} + 225x^{32} + 25x^{31} + 59x^{30} + 184x^{29} + 175x^{28} + 44x^{27} + 115x^{26} + 119x^{25} + 95x^{24} + 137x^{23} + 101x^{22} + 33x^{21} + 68x^{20} + 4x^{19} + 2x^{18} + 18x^{17} + 229x^{16} + 182x^{15} + 80x^{14} + 251x^{13} + 220x^{12} + 179x^{11} + 84x^{10} + 120x^9 + 102x^8 + 181x^7 + 162x^6 + 250x^5 + 130x^4 + 218x^3 + 242x^2 + 127x + 245.$

Полином-делитель для порождения 42 проверочных знаков представляет собой следующее:

$g(x) = x^{42} + 5x^{41} + 9x^{40} + 5x^{39} + 226x^{38} + 177x^{37} + 150x^{36} + 50x^{35} + 69x^{34} + 202x^{33} + 248x^{32} + 101x^{31} + 54x^{30} + 57x^{29} + 253x^{28} + x^{27} + 21x^{26} + 121x^{25} + 57x^{24} + 111x^{23} + 214x^{22} + 105x^{21} + 167x^{20} + 9x^{19} + 100x^{18} + 95x^{17} + 175x^{16} + 8x^{15} + 242x^{14} + 133x^{13} + 245x^{12} + 2x^{11} + 122x^{10} + 105x^9 + 247x^8 + 153x^7 + 22x^6 + 38x^5 + 19x^4 + 31x^3 + 137x^2 + 193x + 77.$

Полином-делитель для порождения 48 проверочных знаков представляет собой следующее:

$g(x) = x^{48} + 19x^{47} + 225x^{46} + 253x^{45} + 92x^{44} + 213x^{43} + 69x^{42} + 175x^{41} + 160x^{40} + 147x^{39} + 187x^{38} + 87x^{37} + 176x^{36} + 44x^{35} + 82x^{34} + 240x^{33} + 186x^{32} + 138x^{31} + 66x^{30} + 100x^{29} + 120x^{28} + 88x^{27} + 131x^{26} + 205x^{25} + 170x^{24} + 90x^{23} + 37x^{22} + 23x^{21} + 118x^{20} + 147x^{19} + 16x^{18} + 106x^{17} + 191x^{16} + 87x^{15} + 237x1^4 + 188x^{13} + 205x^{12} + 231x^{11} + 238x^{10} + 133x^9 + 238x^8 + 22x^7 + 117x^6 + 32x^5 + 96x^4 + 223x^3 + 172x^2 + 132x + 245.$

Полином-делитель для порождения 56 проверочных знаков представляет собой следующее:

$g(x) = x^{56} + 46x^{55} + 143x^{54} + 53x^{53} + 233x^{52} + 107x^{51} + 203x^{50} + 43x^{49} + 155x^{48} + 28x^{47} + 247x^{46} + 67x^{45} + 127x^{44} + 245x^{43} + 137x^{42} + 13x^{41} + 164x^{40} + 207x^{39} + 62x^{38} + 117x^{37} + 201x^{36} + 150x^{35} + 22x^{34} + 238x^{33} + 144x^{32} + 232x^{31} + 29x^{30} + 203x^{29} + 117x^{28} + 234x^{27} + 218x^{26} + 146x^{25} + 228x^{24} + 54x^{23} + 132x^{22} + 200x^{21} + 38x^{20} + 223x^{19} + 36x^{18} + 159x^{17} + 150x^{16} + 235x^{15} + 215x^{14} + 192x^{13} + 230x^{12} + 170x^{11} + 175x^{10} + 29x^{9} + 100x^{8} + 208x^{7} + 220x^{6} + 17x^{5} + 12x^{4} + 238x^{3} + 223x^{2} + 9x + 175.$

Полином-делитель для порождения 62 проверочных знаков представляет собой следующее:

$g(x) = x^{62} + 204x^{61} + 11x^{60} + 47x^{59} + 86x^{58} + 124x^{57} + 224x^{56} + 166x^{55} + 94x^{54} + 7x^{53} + 232x^{52} + 107x^{51} + 4x^{50} + 170x^{49} + 176x^{48} + 31x^{47} + 163x^{46} + 17x^{45} + 188x^{44} + 130x^{43} + 40x^{42} + 10x^{41} + 87x^{40} + 63x^{39} + 51x^{38} + 218x^{37} + 27x^{36} + 6x^{35} + 147x^{34} + 44x^{33} + 161x^{32} + 71x^{31} + 114x^{30} + 64x^{29} + 175x^{28} + 221x^{27} + 185x^{26} + 106x^{25} + 250x^{24} + 190x^{23} + 197x^{22} + 63x^{21} + 245x^{20} + 230x^{19} + 134x^{18} + 112x^{17} + 185x^{16} + 37x^{15} + 196x^{14} + 108x^{13} + 143x^{12} + 189x^{11} + 201x^{10} + 188x^{9} + 202x^{8} + 118x^{7} + 39x^{6} + 210x^{5} + 144x^{4} + 50x^{3} + 169x^{2} + 93x + 242.$

Полином-делитель для порождения 68 проверочных знаков представляет собой следующее:

$g(x) = x^{68} + 186x^{67} + 82x^{66} + 103x^{65} + 96x^{64} + 63x^{63} + 132x^{62} + 153x^{61} + 108x^{60} + 54x^{59} + 64x^{58} + 189x^{57} + 211x^{56} + 232x^{55} + 49x^{54} + 25x^{53} + 172x^{52} + 52x^{51} + 59x^{50} + 241x^{49} + 181x^{48} + 239x^{47} + 223x^{46} + 136x^{45} + 231x^{44} + 210x^{43} + 96x^{42} + 232x^{41} + 220x^{40} + 25x^{39} + 179x^{38} + 167x^{37} + 202x^{36} + 185x^{35} + 153x^{34} + 139x^{33} + 66x^{32} + 236x^{31} + 227x^{30} + 160x^{29} + 15x^{28} + 213x^{27} + 93x^{26} + 122x^{25} + 68x^{24} + 177x^{23} + 158x^{22} + 197x^{21} + 234x^{20} + 180x^{19} + 248x^{18} + 136x^{17} + 213x^{16} + 127x^{15} + 73x^{14} + 36x^{13} + 154x^{12} + 244x^{11} + 147x^{10} + 33x^{9} + 89x^{8} + 56x^{7} + 159x^{6} + 149x^{5} + 251x^{4} + 89x^{3} + 173x^{2} + 228x + 220.$

E.2 Алгоритм обнаружения и исправления ошибок

Алгоритм Петерсона—Горенштейна—Зирлера может быть использован для исправления ошибок в декодированных символах Data Matrix.

Нижеуказанные вычисления следуют этому алгоритму исправления ошибок, используя кодовые слова исправления ошибок Рида—Соломона.

Стирания должны быть исправлены как ошибки путем первоначального заполнения позиций всех стертых кодовых слов фиктивными значениями.

Все вычисления проводят с помощью арифметических операций GF(2^8). Сложение и вычитание в этом поле соответствует проведению двоичной операции «исключающего ИЛИ» («exclusive-or» — XOR). Умножение и деление могут быть выполнены с помощью таблиц логарифмов и антилогарифмов.

Составляют полином знаков символа $C(x) = C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + ... + C_1x^1 + C_0$, где n коэффициентов полинома являются считанными кодовыми словами, причем Cn-1 относится к первому знаку символа, а n — общее число знаков символа.

Вычисляют i величин синдромов от S_0 до S_{i-1} путем вычисления полинома C(x) при $x = 2^k$ для k = от 1 до i, где i — число кодовых слов исправления ошибок в символе.

Составляют и решают систему j уравнений с j неизвестными от L_0 до L_{j-1}, используя i синдромов:

$$S_0L_0 + S_1L_1 + ... + S_j - 1L_{j-1} = S_j$$
$$S_1L_0 + S_2L_1 + ... + S_jL_{j-1} = S_{j+1}$$
$$\vdots$$
$$\vdots$$
$$S_j - 1L_0 + S_jL_1 + ... + S_{2j} - 2L_j - 1 = S_{2j-1},$$

где $j = i/2$.

Составляют полином указания местонахождения ошибок:

$$L(x) = L_j - 1x^j + L_j - 2x^{j-1} + ... + L_0x + 1$$

из j величин L, вычисленных выше. Вычисляют $L(x)$ для $x = 2^k$, где k = от 0 до n – 1, где n — общее число кодовых слов в символе.

Как только $L(2^k) = 0$, позиция ошибки определяется как $n – 1 – k$. Если найдено больше местоположений ошибок, чем значение j, данный символ невозможно исправить.

Сохраняют местоположения ошибок в m переменных указателях местоположения ошибок от E_0 до E_{m-1}, где m — число найденных местоположений ошибок. Составляют и решают систему m уравнений с m неизвестными от X_0 до X_{m-1} (значения ошибок), используя переменные указатели местоположения ошибок и первые m синдромов S:

$$E_0X_0 + E_1X_1 + ... + E_{m-1}X_{m-1} = S_0$$
$$E_0^2X_0 + E_1^2X_1 + ... + E_{(m-1)}^2X_{m-1} = S_1$$
$$E_0^3X_0 + E_1^3X_1 + ... + E_{(m-1)}^3X_{m-1} = S_2$$
$$\vdots$$

:

$$E_0^m X_0 + E_1^m X_1 + ... + E_{(m-1)}^m X_{m-1} = S_{m-1}.$$

Добавляют значения ошибок от X_0 до X_{m-1} к значениям знаков символа в соответствующих местоположениях ошибок от E_0 до E_{m-1} для исправления ошибок.

Примечание — E_0, ..., E_{m-1} являются корнями полинома указателя местоположения ошибок.

E.3 Вычисление кодовых слов исправления ошибки

В следующем примере программы на языке программирования С вычисляются кодовые слова исправления ошибки для заданной входной строки данных длиной «nd», записанной в целочисленном массиве wd[ ]. Функция ReedSolomon( ) сначала генерирует таблицы логарифмов и антилогарифмов для поля Галуа размером «gf» (= 2^8) с примитивным полиномом «pp» (= 301), затем использует их в функции prod( ), сначала для вычисления коэффициентов порождающего полинома степени «nc», а затем для вычисления «nc» дополнительных проверочных кодовых слов, которые добавляют к кодовым словам данных в массиве wd, см. [5, 6].

```
/* "prod(x,y,log,alog,gf)" returns the product "x" times "y" */
int prod(int x, int y, int *log, int *alog, int gf) {
   if (!x || !y) return 0;
   ELSE return alog[(log[x] + log[y]) % (gf-1)];
}

/* "ReedSolomon(wd,nd,nc,gf.pp)" takes "nd" data codeword values in wd[] */
/* and adds on "nc" check codewords, all within GF(gf) where "gf" is a */
/* power of 2 and "pp" is the value of its prime modulus polynomial */
void ReedSolomon(int *wd, int nd, int nc, int gf, int pp) {
int i, j, k. *log,*alog,*c;

/* allocate, then generate the log & antilog arrays: */
   log = malloc(sizeof(int) * gf);
   alog = malloc(sizeof(int) * gf);
   log[0] = 1-gf; alog[0] = 1;
   for (i = 1; i < gf; i++) {
      alog[i] = alog[i-1] * 2;
      if (alog[i] >= gf) alog[i] ^= pp;
      log[alog[i]] = i;
}

/* allocate, then generate the generator polynomial coefficients: */
   c = malloc(sizeof(int) * (nc+1));
   for (i=1; i<=nc; i++) c[i] = 0; c[0] = 1;
   for (i=1; i<=nc; i++) {
      c[i] = c[i-1];
      for (j=i-1; j>=1; j--) {
      c[j] = c[j-1] ^ prod(c[j],alog[i],log,alog,gf);
   }
      c[0] = prod(c[0],alog[i],log,alog,gf);
}

/* clear, then generate "nc" checkwords in the array wd[] : */
   for (i=nd; i<=(nd+nc); i++) wd[i] = 0;
   for (i=0; i<nd; i++) {
      k = wd[nd] ^ wd[i] ;
      for (j=0; j<nc; j++) {
      wd[nd+j] = wd[nd+j+1] ^ prod(k,c[nc-j-1],log, alog,gf);
      }
   }
   free(c);
   free(alog);
   free(log);
}
```

Приложение F
(обязательное)

Размещение знаков символа

F.1 Пример программы размещения символов

Следующая программа на языке программирования С генерирует схемы размещения знаков символа

```
#include <stdio.h>
#include <alloc.h>

int nrow, ncol, *array;

/* "module" places "chr+bit" with appropriate wrapping within array[] */
void module(int row, int col, int chr, int bit)
{ if (row < 0) { row += nrow; col += 4 - ((nrow+4)%8); }
   if (col < 0) { col += ncol; row += 4 - ((ncol+4)%8); }
   array[row*ncol+col] = 10*chr + bit;
}
/* "utah" places the 8 bits of a utah-shaped symbol character */
void utah(int row, int col, int chr)
{  module(row-2,col-2,chr,1);
      module(row-2,col-1,chr,2);
      module(row-1,col-2,chr,3);
      module(row-1,col-1,chr,4);
      module(row-1,col,chr,5);
      module(row,col-2,chr,6);
      module(row,col-1,chr,7);
      module(row,col,chr,8);
}
/* "cornerN" places 8 bits of the four special corner cases */
void corner1(int chr)
{  module(nrow-1,0,chr,1);
      module(nrow-1,1,chr,2);
      module(nrow-1,2,chr,3);
      module(0,ncol-2,chr,4);
      module(0,ncol-1,chr,5);
      module(1,ncol-1,chr,6);
      module(2,ncol-1,chr,7);
      module(3,ncol-1,chr,8);
}
void corner2(int chr)
{  module(nrow-3,0,chr,1);
      module(nrow-2,0,chr,2);
      module(nrow-1,0,chr,3);
      module(0,ncol-4,chr,4);
      module(0,ncol-3,chr,5);
      module(0,ncol-2,chr,6);
      module(0,ncol-1,chr,7);
      module(1,ncol-1,chr,8);
}
void corner3(int chr)
{  module(nrow-3,0,chr,1);
      module(nrow-2,0,chr,2);
      module(nrow-1,0,chr,3);
      module(0,ncol-2,chr,4);
      module(0,ncol-1,chr,5);
      module(1,ncol-1,chr,6);
      module(2,ncol-1,chr,7);
      module(3,ncol-1,chr,8);
}
void corner4(int chr)
{  module(nrow-1,0,chr,1);
```

```
      module(nrow-1,ncol-1,chr,2); module(0,ncol-3,chr,3);
      module(0,ncol-2,chr,4);
      module(0,ncol-1,chr,5);
      module(1,ncol-3,chr,6);
      module(1,ncol-2,chr,7);
      module(1,ncol-1,chr,8);
   }
   /* "PlaceDM" fills an nrow x ncol array with appropriate values */
   void PlaceDM (void)
   { int row, col, chr;

   /* First, fill the array[] with invalid entries */
      for (row=0; row<nrow; row++) {
for (col=0; col<ncol; col++) {
      array[row*ncol+col] = 0;
}
   }
   /* Starting in the correct location for character #1, bit 8,... */
     chr = 1; row = 4; col = 0;

   do {
   /* repeatedly first check for one of the special corner cases, then... */
      if ((row == nrow) && (col == 0)) corner1(chr++);
      if ((row == nrow-2) && (col == 0) && (ncol%4)) corner2(chr++);
      if ((row == nrow-2) && (col == 0) && (ncol%8 == 4)) corner3(chr++);
      if ((row == nrow+4) && (col == 2) && (!(ncol%8))) corner4(chr++);
   /* sweep upward diagonally, inserting successive characters,... */
      do {
if ((row < nrow) && (col >= 0) && (!array[row*ncol+col]))
      utah(row,col,chr++);
      row -= 2; col += 2;
         } while ((row >= 0) && (col < ncol));
         row += 1; col += 3;

   /* & then sweep downward diagonally, inserting successive characters,... */
      +
      do {
           if ((row >= 0) && (col < ncol) && (!array[row*ncol+col]))
           utah(row,col,chr++);
   row += 2; col -= 2;
      } while ((row < nrow) && (col >= 0));
      row += 3; col += 1;

   /* ... until the entire array is scanned */
      } while ((row < nrow) || (col < ncol));

   /* Lastly, if the lower righthand corner is untouched, fill in fixed pattern */
      if (!array[nrow*ncol-1]) {
   array[nrow*ncol-1] = array[nrow*ncol-ncol-2] = 1;
   }
      }

   /* "main" checks for valid command line entries, then computes & displays array */
   void main(int argc, char *argv[])
   { int x, y, z;
      if (argc =< 3) {
      printf("Command line: #_of_Data_Rows #_of_Data_Columns\n");
         } ELSE {
            nrow = ncol = 0;
            nrow = atoi(argv[1]); ncol = atoi(argv[2]);
            if ((nrow >= 6) && (~nrow&0x01) && (ncol >= 6) && (~ncol&0x01)) {
            array = malloc(sizeof(int) * nrow * ncol);
```

```
        PlaceDM();

        for (x=0; x<nrow; x++) {
           for (y=0; y<ncol; y++) {
              z = array[x*ncol+y];
              if (z == 0) printf(" WHI");
                 ELSE if (z == 1) printf("BLK");
                 ELSE printf("%3d.%d",z/10,z%10);
              }
              printf("\n");
           }
           free(array);
        }
     }
}
```

F.2 Правила размещения знаков символа

F.2.1 Нетипичная форма знака символа

Ввиду того, что знаки символа типичной формы не всегда могут быть размещены в границах модулей данных в символе и в некоторых его углах, необходим небольшой набор знаков символа нетипичной формы. Установлено шесть условий размещения — два условия размещения на границах, которые действуют во всех форматах символа, и четыре различных условия размещения в углах, которые применимы для отдельных форматов символа:

a) одну часть знака символа нетипичной формы размещают на одной стороне символа и другую — на противоположной. Это условие применяют к двум базовым формам знака символа (см. рисунок F.1). Варианты таких компоновок зависят от взаимосвязи между строками на левой и правой границах (см. таблицу F.1);

b) одну часть знака символа размещают на верхней границе символа, а другую — на нижней. Это условие применяют к двум базовым формам знака символа (см. рисунок F.2). Варианты такой компоновки зависят от взаимосвязи между столбцами на верхней и нижней границах (см. таблицу F.1);

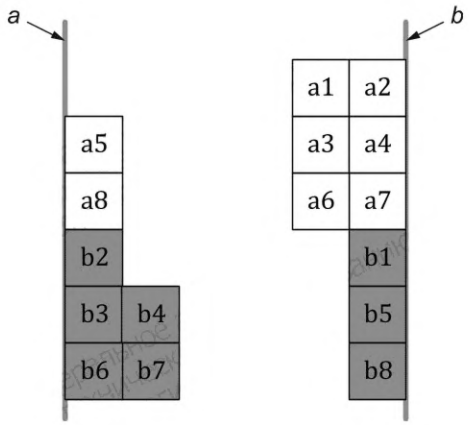
c) четыре знака символа нетипичной формы распределяют по двум или трем углам (см. рисунки F.3 — F.6). Знаки символа нетипичной формы размещают на противоположных границах. Число таких пар возрастает прямо пропорционально периметру координатной матрицы. Базовые комбинации представлены на рисунках F.1 и F.2. На рисунке F.1 модули a8 и a7 находятся в одной и той же строке, так же как и модули b7 и b6. На рисунке F.2 модули c6 и c3 находятся в одном и том же столбце, так же как и модули d3 и d1. Существует семь вариантов размещения знаков символа на границах, которые определяют относительное положение знаков символа по вертикали, приведенное на рисунке F.1, положение по горизонтали, приведенное на рисунке F.2, и условия размещения в углах.

Таблица F.1 — Факторы, определяющие вариант размещения на границах

Вариант размещения на границах	Взаимосвязь модулей a8 и a7 в строках	Взаимосвязь модулей c6 и c3 в столбцах	Номер рисунка, поясняющего условия в углах	Используемые отображающие матрицы	Номер рисунка приложения F с примером
1	Строка с модулем a7 = строке с модулем a8	Столбец с модулем c3 = столбцу с модулем c6	Нет	Квадратные: 8^2, 16^2, 24^2, 32^2, 40^2, 48^2, 56^2, 64^2, 72^2, 80^2, 88^2, 96^2 и 120^2	F.9 и F.16
2	Строка с модулем a7 = строке с модулем a8 – 2	Столбец с модулем c3 = столбцу с модулем c6 – 2	Нет	Квадратные: 10^2 и 18^2	F.10 и F.17
3	Строка с модулем a7 = строке с модулем a8 + 4	Столбец с модулем c3 = столбцу с модулем c6 + 4	F.3	Квадратные: 12^2, 20^2, 28^2, 36^2, 44^2, 108^2 и 132^2	F.11 и F.18
4	Строка с модулем a7 = строке с модулем a8 + 2	Столбец с модулем c3 = столбцу с модулем c6 + 2	F.4	Квадратные: 14^2 и 22^2	F.12 и F.19

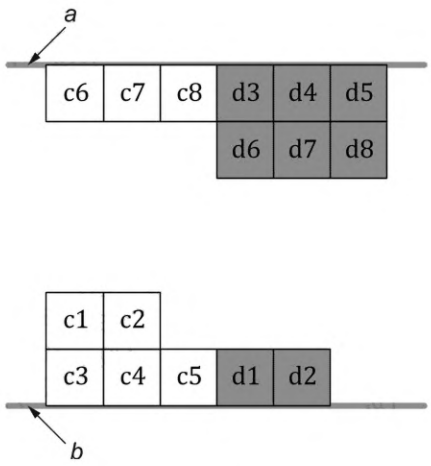
Окончание таблицы F.1

Вариант размещения на границах	Взаимосвязь модулей a8 и a7 в строках	Взаимосвязь модулей c6 и c3 в столбцах	Номер рисунка, поясняющего условия в углах	Используемые отображающие матрицы	Номер рисунка приложения F с примером
5	Строка с модулем a7 = строке с модулем a8	Столбец c3 = столбцу с модулем c6 + 2	F.5	Прямоугольные: 6 × 16 и 14 × 32	F.13
6	Строка с модулем a7 = строке с модулем a8	Столбец с модулем c3 = столбцу с модулем c6 – 2	Нет	Прямоугольные: 10 × 24 и 10 × 32	F.14
7	Строка с модулем a7 = строке с модулем a8 + 4	Столбец с модулем c3 = столбцу с модулем c6 + 2	F.6	Прямоугольные: 6 × 28 и 14 × 44	F.15



a — левая граница; *b* — правая граница

Рисунок F.1 — Правый и левый знаки символа



a — верхняя граница; *b* — нижняя граница

Рисунок F.2 — Верхний и нижний знаки символа

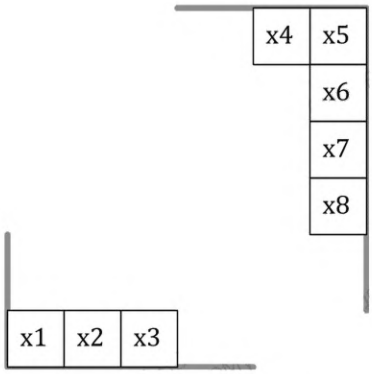


Рисунок F.3 — Первое условие размещения знака символа в углах

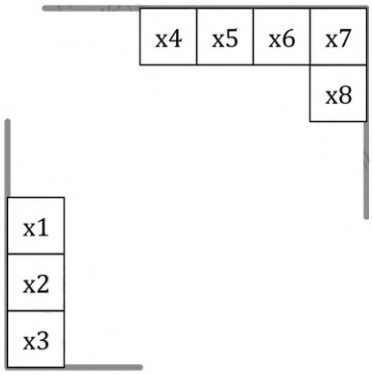


Рисунок F.4 — Второе условие размещения знака символа в углах

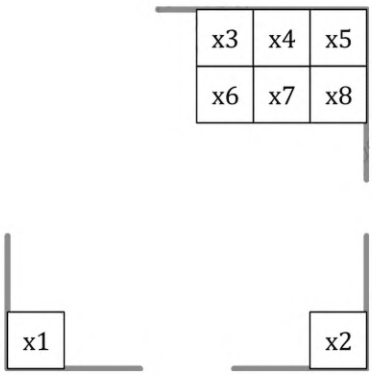


Рисунок F.5 — Третье условие размещения знака символа в углах

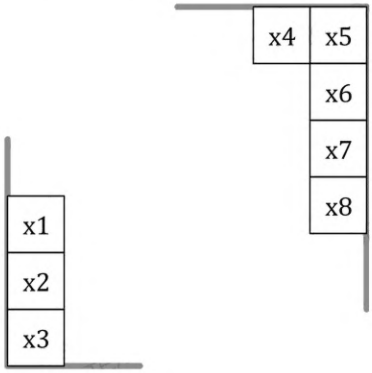


Рисунок F.6 — Четвертое условие размещения знака символа в углах

Примечание 1 — Для идентификации знака символа используют алгебраические обозначения, поскольку они варьируются в зависимости от формата символа.

Примечание 2 — Угловые знаки идентифицируют по модулю в левом нижнем и правом верхнем углах.

F.2.2 Размещение знаков символа

Знаки символа помещают в матрицу следующим образом:

a) создают координатную матрицу:

1) в небольших символах с единственной областью данных эта область совпадает с координатной матрицей;

2) в больших символах, имеющих более одной области данных, координатная матрица соответствует совокупному размеру смежных областей данных. Таким образом, координатная матрица не содержит разделительных направляющих шаблонов. Например, символ формата 36 × 36 имеет четыре области данных 16 × 16 которые, примыкая друг к другу, образуют координатную матрицу 32 × 32. Размер координатной матрицы для каждого формата символа приведен в таблице 10. Варианты размещения на границах приведены в таблице F.1;

b) второй знак символа размещают в верхней левой позиции, где его модули образуют последовательность битов (модулей), указанную на рисунке 3. Используют обозначение 2.1 для идентификации первого модуля второго знака символа. Этот модуль находится в верхней строке и в крайнем левом столбце каждой координатной матрицы. Последовательность массивов модулей, изображенная на рисунке F.7, является постоянной для всех координатных матриц;

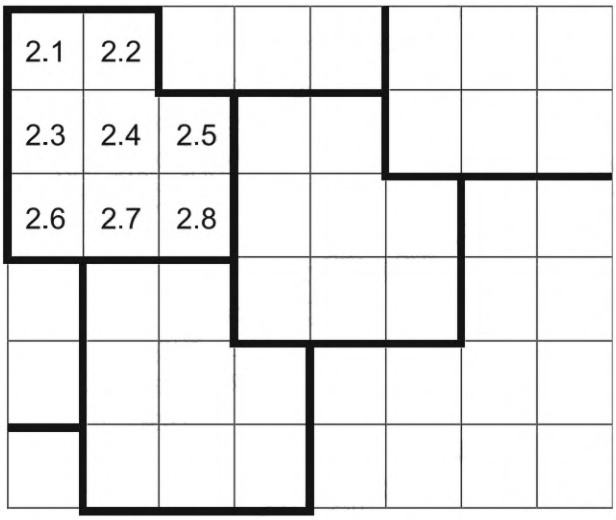


Рисунок F.7 — Начальная последовательность размещения модулей

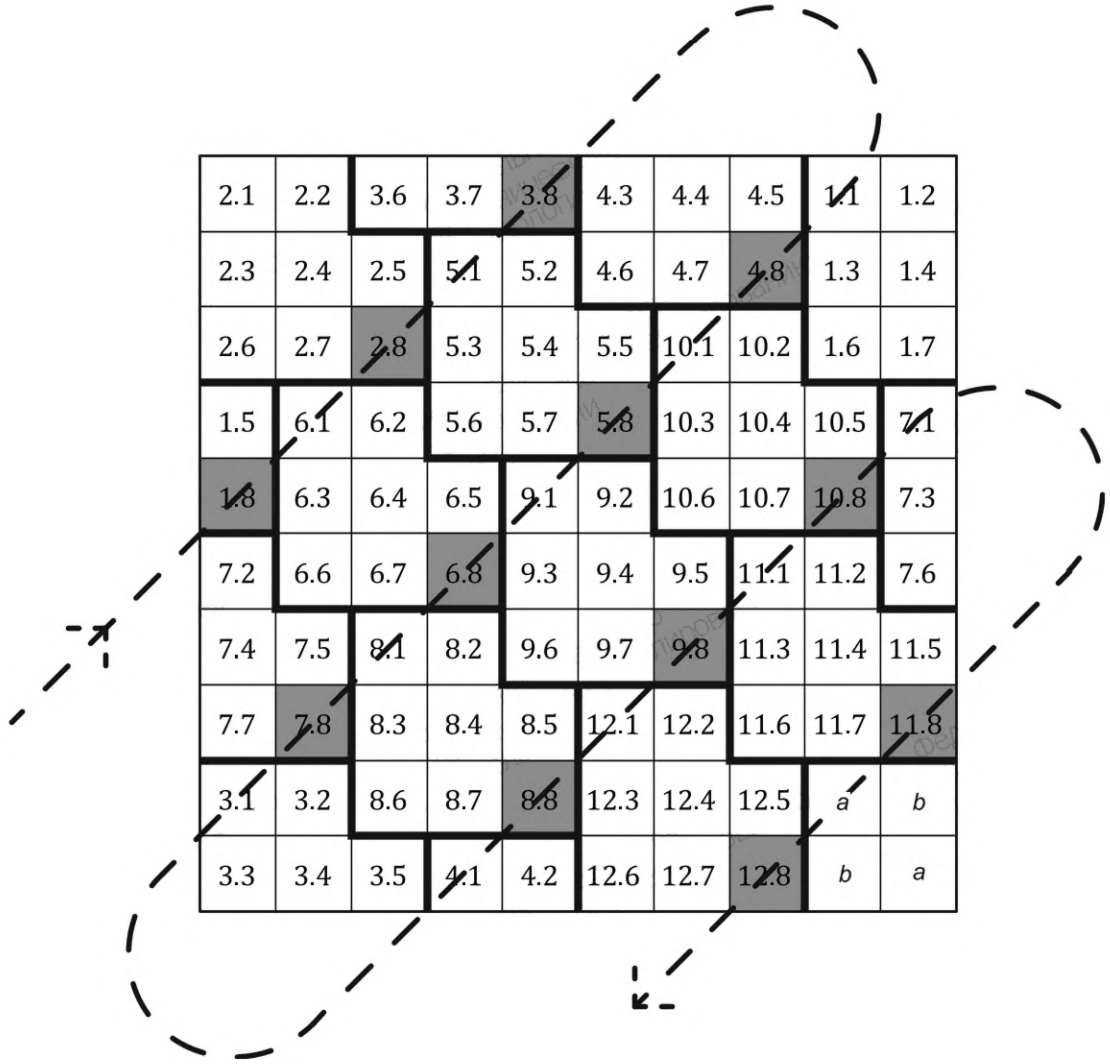
c) размещение модулей в углах должно соответствовать таблице F.1 и соответствующим рисункам (см. рисунки F.3 — F.6). Построение знаков символа типичной формы продолжают, сопрягая формы, как это показано выше для знаков символа 2, 5 и 6. Нетипичные знаки символа располагают в соответствии с таблицей F.1. Этот процесс дает в результате полное покрытие координатной матрицы знаками символа, большинство из которых не пронумерованы;

d) порядок следования знаков символа определен следующим образом. Знаки символа размещают вдоль параллельных диагональных линий с крайними точками левой нижней и правой верхней, наклоненных под углом 45° к границам символа, которые проходят через центры восьмых модулей знаков;

e) первую диагональную линию начинают как линию, проходящую через восьмой модуль первого знака символа, за исключением использования координатной матрицы размером 6 × 28, когда условие размещения в углах, приведенное на рисунке F.6, определяет значения модулей в первом знаке символа (т. е. модуль, обозначенный на рисунке F.7 как 1.b, представляет модуль 1.2). Диагональную линию продолжают через модули 2.8 и 3.8;

f) в этой точке диагональная линия пересекает границу верхней строки. Следующую диагональную линию начинают с точки четвертого модуля справа от точки пересечения с верхней границей в верхней строке или, при использовании координатной матрицы размером 8 × 8, в точке третьего модуля справа и одного модуля вниз, т. е. начало диагональной линии смещено вправо на четыре модуля. Знаки символа нумеруют по очередности их следования вдоль пути размещения, пересекающего восьмые модули. Таким образом, следующие знаки определены нисходящей диагональной линией, пересекающей модули 4.8, 5.8, 6.8 и так далее;

g) путь размещения знаков символа, как показано на рисунке F.8, продолжают по диагональным линиям, смещенным на четыре модуля вправо (или на четыре модуля вниз, или на комбинацию этих вариантов) от предыдущей диагональной линии. Первая и все нечетные диагональные линии отображают порядок следования знаков символа по направлению снизу вверх и слева направо. Вторая и все четные диагональные линии отображают порядок следования знаков символа по направлению сверху вниз и справа налево;



a — BLK (темный модуль); *b* — WHT (светлый модуль)

Рисунок F.8 — Последовательность размещения знаков символа

h) когда на пути размещения встречают условие для размещения знака символа нетипичной формы, который целиком не может быть размещен в границах координатной матрицы, часть этого знака символа продолжают на противоположной стороне матрицы, что приводит к необходимости нумерации противоположных частей таких знаков символа до того, как путь размещения пересечет отделенную часть. Например, в координатной матрице (см. рисунок F.8) отдельные части знаков символа три и семь пронумерованы до того, как путь размещения пересек их. Таким образом, вдоль пути размещения нумеруют только непронумерованные знаки символа. Данные условия размещения на границах и в углах определены в таблице F.1, что подтверждается рисунком F.8 для знаков символа 1, 3, 4 и 7. Условия размещения в углах также влияют на порядок нумерации. Нумерацию нижнего левого угла проводят:

- непосредственно перед знаком символа (см. рисунок F.3), расположенным над ним (примеры приведены на рисунках F.11 и F.18);
- непосредственно перед знаком символа (см. рисунок F.4), расположенным над ним (примеры приведены на рисунках F.12 и F.19);
- непосредственно после знака символа (см. рисунок F.5), расположенного справа от него (пример приведен на рисунке F.13);
- непосредственно перед знаком символа (см. рисунок F.6), расположенным над ним (пример приведен на рисунке F.15).

Остающиеся модули угла нумеруют перед тем, как путь размещения пересечет их;

i) процедуру размещения продолжают до тех пор, пока не будут размещены все знаки символа, и заканчивают в нижнем правом углу координатной матрицы. В координатных матрицах, имеющих четыре размера (10 × 10,

14 × 14, 18 × 18 и 22 × 22), остается область 2 × 2 в правом нижнем углу. Верхний левый и нижний правый модули этой области — темные (номинально кодирующие двоичную единицу), как показано на рисунке F.8.

Типовые координатные матрицы, сформированные согласно этой процедуре, приведены в F.3. На рисунках F.9—F.15 представлены варианты размещения на границах с 1 по 7 соответственно. На рисунках F.16—F.19 представлены другие примеры для вариантов размещения с 1 по 4. Программа на языке программирования С, способная отображать все кодируемые биты в соответствующей координатной матрице, приведена в F.1.

F.3 Примеры размещения знаков символа

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6

Рисунок F.9 — Размещение кодовых слов в квадратной координатной матрице размером 8 × 8

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	1.1	1.2
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	1.3	1.4
2.6	2.7	2.8	5.3	5.4	5.5	10.1	10.2	1.6	1.7
1.5	6.1	6.2	5.6	5.7	5.8	10.3	10.4	10.5	7.1
1.8	6.3	6.4	6.5	9.1	9.2	10.6	10.7	10.8	7.3
7.2	6.6	6.7	6.8	9.3	9.4	9.5	11.1	11.2	7.6
7.4	7.5	8.1	8.2	9.6	9.7	9.8	11.3	11.4	11.5
7.7	7.8	8.3	8.4	8.5	12.1	12.2	11.6	11.7	11.8
3.1	3.2	8.6	8.7	8.8	12.3	12.4	12.5	*a*	*b*
3.3	3.4	3.5	4.1	4.2	12.6	12.7	12.8	*b*	*a*

a — BLK (темный модуль); *b* — WHT (светлый модуль)

Рисунок F.10 — Размещение кодовых слов в квадратной координатной матрице размером 10 × 10[1)]

[1)] *Размещение кодовых слов на рисунке F.10 соответствует рисунку F.10 ISO/IEC 16022:2006, поскольку рисунок F.10 ISO/IEC 16022:2024 содержит ошибку в отношении кодового слова 9 и границ кодового слова 3.*

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	8.4	8.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	8.6
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	8.7
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	14.1	14.2	8.8
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	14.3	14.4	14.5
7.2	6.6	6.7	6.8	11.3	11.4	11.5	15.1	15.2	14.6	14.7	14.8
7.4	7.5	10.1	10.2	11.6	11.7	11.8	15.3	15.4	15.5	1.1	1.2
7.7	7.8	10.3	10.4	10.5	16.1	16.2	15.6	15.7	15.8	1.3	1.4
9.1	9.2	10.6	10.7	10.8	16.3	16.4	16.5	18.1	18.2	1.6	1.7
9.3	9.4	9.5	17.1	17.2	16.6	16.7	16.8	18.3	18.4	18.5	7.1
9.6	9.7	9.8	17.3	17.4	17.5	3.1	3.2	18.6	18.7	18.8	7.3
8.1	8.2	8.3	17.6	17.7	17.8	3.3	3.4	3.5	4.1	4.2	7.6

Рисунок F.11 — Размещение кодовых слов в квадратной координатной матрице размером 12 × 12

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	8.4	8.5	8.6	8.7
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	14.1	14.2	8.8
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	14.3	14.4	14.5
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	15.1	15.2	14.6	14.7	14.8
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	15.3	15.4	15.5	1.1	1.2
7.2	6.6	6.7	6.8	11.3	11.4	11.5	16.1	16.2	15.6	15.7	15.8	1.3	1.4
7.4	7.5	10.1	10.2	11.6	11.7	11.8	16.3	16.4	16.5	22.1	22.2	1.6	1.7
7.7	7.8	10.3	10.4	10.5	17.1	17.2	16.6	16.7	16.8	22.3	22.4	22.5	7.1
9.1	9.2	10.6	10.7	10.8	17.3	17.4	17.5	21.1	21.2	22.6	22.7	22.8	7.3
9.3	9.4	9.5	18.1	18.2	17.6	17.7	17.8	21.3	21.4	21.5	23.1	23.2	7.6
9.6	9.7	9.8	18.3	18.4	18.5	20.1	20.2	21.6	21.7	21.8	23.3	23.4	23.5
8.1	19.1	19.2	18.6	18.7	18.8	20.3	20.4	20.5	24.1	24.2	23.6	23.7	23.8
8.2	19.3	19.4	19.5	3.1	3.2	20.6	20.7	20.8	24.3	24.4	24.5	*a*	*b*
8.3	19.6	19.7	19.8	3.3	3.4	3.5	4.1	4.2	24.6	24.7	24.8	*b*	*a*

a — BLK (темный модуль); *b* — WHT (светлый модуль)

Рисунок F.12 — Размещение кодовых слов в квадратной координатной матрице размером 14 × 14

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	9.1	9.2	10.6	10.7	10.8	7.3	7.4	7.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	9.3	9.4	9.5	11.1	11.2	7.6	7.7	7.8
2.6	2.7	2.8	5.3	5.4	5.5	8.1	8.2	9.6	9.7	9.8	11.3	11.4	11.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	8.3	8.4	8.5	12.1	12.2	11.6	11.7	11.8	1.3	1.4
1.8	6.3	6.4	6.5	3.1	3.2	8.6	8.7	8.8	12.3	12.4	12.5	10.1	10.2	1.6	1.7
7.1	6.6	6.7	6.8	3.3	3.4	3.5	4.1	4.2	12.6	12.7	12.8	10.3	10.4	10.5	7.2

Рисунок F.13 — Размещение кодовых слов в прямоугольной координатной матрице размером 6 × 16

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	11.1	11.2	12.6	12.7	12.8	13.3	13.4	13.5	21.1	21.2	22.6	22.7	22.8	23.3	23.4	23.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	11.3	11.4	11.5	14.1	14.2	13.6	13.7	13.8	21.3	21.4	21.5	24.1	24.2	23.6	23.7	23.8
2.6	2.7	2.8	5.3	5.4	5.5	10.1	10.2	11.6	11.7	11.8	14.3	14.4	14.5	20.1	20.2	21.6	21.7	21.8	24.3	24.4	24.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	10.3	10.4	10.5	15.1	15.2	14.6	14.7	14.8	20.3	20.4	20.5	25.1	25.2	24.6	24.7	24.8	1.3	1.4
1.8	6.3	6.4	6.5	9.1	9.2	10.6	10.7	10.8	15.3	15.4	15.5	19.1	19.2	20.6	20.7	20.8	25.3	25.4	25.5	29.1	29.2	1.6	1.7
7.2	6.6	6.7	6.8	9.3	9.4	9.5	16.1	16.2	15.6	15.7	15.8	19.3	19.4	19.5	26.1	26.2	25.6	25.7	25.8	29.3	29.4	29.5	7.1
7.4	7.5	8.1	8.2	9.6	9.7	9.8	16.3	16.4	16.5	18.1	18.2	19.6	19.7	19.8	26.3	26.4	26.5	28.1	28.2	29.6	29.7	29.8	7.2
7.7	7.8	8.3	8.4	8.5	17.1	17.2	16.6	16.7	16.8	18.3	18.4	18.5	27.1	27.2	26.6	26.7	26.8	28.3	28.4	28.5	30.1	30.2	7.3
3.1	3.2	8.6	8.7	8.8	17.3	17.4	17.5	12.1	12.2	18.6	18.7	18.8	27.3	27.4	27.5	22.1	22.2	28.6	28.7	28.8	30.3	30.4	30.5
3.3	3.4	3.5	4.1	4.2	17.6	17.7	17.8	12.3	12.4	12.5	13.1	13.2	27.6	27.7	27.8	22.3	22.4	22.5	23.1	23.2	30.6	30.7	30.8

Рисунок F.14 — Размещение кодовых слов в прямоугольной координатной матрице размером 10 × 24

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	8.1	8.2	9.6	9.7	9.8	10.3	10.4	10.5	14.1	14.2	15.6	15.7	15.8	16.3	16.4	16.5	20.1	20.2	1.4	1.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	8.3	8.4	8.5	11.1	11.2	10.6	10.7	10.8	14.3	14.4	14.5	17.1	17.2	16.6	16.7	16.8	20.3	20.4	20.5	1.6
2.6	2.7	2.8	5.3	5.4	5.5	7.1	7.2	8.6	8.7	8.8	11.3	11.4	11.5	13.1	13.2	14.6	14.7	14.8	17.3	17.4	17.5	19.1	19.2	20.6	20.7	20.8	1.7
1.1	6.1	6.2	5.6	5.7	5.8	7.3	7.4	7.5	12.1	12.2	11.6	11.7	11.8	13.3	13.4	13.5	18.1	18.2	17.6	17.7	17.8	19.3	19.4	19.5	21.1	21.2	1.8
1.2	6.3	6.4	6.5	3.1	3.2	7.6	7.7	7.8	12.3	12.4	12.5	9.1	9.2	13.6	13.7	13.8	18.3	18.4	18.5	25.1	25.2	19.6	19.7	19.8	21.3	21.4	21.5
1.3	6.6	6.7	6.8	3.3	3.4	3.5	4.1	4.2	12.6	12.7	12.8	9.3	9.4	9.5	26.1	26.2	18.6	18.7	18.8	25.3	25.4	25.5	16.1	16.2	21.6	21.7	21.8

Рисунок F.15 — Размещение кодовых слов в прямоугольной координатной матрице размером 6 × 28[1)]

[1)] *Размещение кодовых слов на рисунке F.15 соответствует рисунку F.15 ISO/IEC 16022:2006, поскольку рисунок F.15 ISO/IEC 16022:2024 содержит ошибку в отношении кодового слова 17.*

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5	13.1	13.2	14.6	14.7	14.8	15.3	15.4	15.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	16.1	16.2	15.6	15.7	15.8
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	16.3	16.4	16.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	17.1	17.2	16.6	16.7	16.8	1.3	1.4
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	17.3	17.4	17.5	27.1	27.2	1.6	1.7
7.2	6.6	6.7	6.8	11.3	11.4	11.5	18.1	18.2	17.6	17.7	17.8	27.3	27.4	27.5	7.1
7.4	7.5	10.1	10.2	11.6	11.7	11.8	18.3	18.4	18.5	26.1	26.2	27.6	27.7	27.8	7.3
7.7	7.8	10.3	10.4	10.5	19.1	19.2	18.6	18.7	18.8	26.3	26.4	26.5	28.1	28.2	7.6
9.1	9.2	10.6	10.7	10.8	19.3	19.4	19.5	25.1	25.2	26.6	26.7	26.8	28.3	28.4	28.5
9.3	9.4	9.5	20.1	20.2	19.6	19.7	19.8	25.3	25.4	25.5	29.1	29.2	28.6	28.7	28.8
9.6	9.7	9.8	20.3	20.4	20.5	24.1	24.2	25.6	25.7	25.8	29.3	29.4	29.5	8.1	8.2
8.5	21.1	21.2	20.6	20.7	20.8	24.3	24.4	24.5	30.1	30.2	29.6	29.7	29.8	8.3	8.4
8.8	21.3	21.4	21.5	23.1	23.2	24.6	24.7	24.8	30.3	30.4	30.5	32.1	32.2	8.6	8.7
22.2	21.6	21.7	21.8	23.3	23.4	23.5	31.1	31.2	30.6	30.7	30.8	32.3	32.4	32.5	22.1
22.4	22.5	3.1	3.2	23.6	23.7	23.8	31.3	31.4	31.5	14.1	14.2	32.6	32.7	32.8	22.3
22.7	22.8	3.3	3.4	3.5	4.1	4.2	31.6	31.7	31.8	14.3	14.4	14.5	15.1	15.2	22.6

Рисунок F.16 — Размещение кодовых слов в квадратной координатной матрице размером 16 × 16[1)]

[1)] *Размещение бита 8.8 приведено в соответствие с рисунком F.16 ISO/IEC 16022:2006, поскольку рисунок F.16 ISO/IEC 16022:2024 содержит ошибку.*

2.1	2.2	3.1	3.7	3.8	4.3	4.4	4.5	13.1	13.2	14.6	14.7	14.8	15.3	15.4	15.5	1.1	1.2
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	16.1	16.2	15.6	15.7	15.8	1.3	1.4
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	16.3	16.4	16.5	29.1	29.2	1.6	1.7
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	17.1	17.2	16.6	16.7	16.8	29.3	29.4	29.5	7.1
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	17.3	17.4	17.5	28.1	28.2	29.6	29.7	29.8	7.3
7.2	6.6	6.7	6.8	11.3	11.4	11.5	18.1	18.2	17.6	17.7	17.8	28.3	28.4	28.5	30.1	30.2	7.6
7.4	7.5	10.1	10.2	11.6	11.7	11.8	18.3	18.4	18.5	27.1	27.2	28.6	28.7	28.8	30.3	30.4	30.5
7.7	7.8	10.3	10.4	10.5	19.1	19.2	18.6	18.7	18.8	27.3	27.4	27.5	31.1	31.2	30.6	30.7	30.8
9.1	9.2	10.6	10.7	10.8	19.3	19.4	19.5	26.1	26.2	27.6	27.7	27.8	31.3	31.4	31.5	8.1	8.2
9.3	9.4	9.5	20.1	20.2	19.6	19.7	19.8	26.3	26.4	26.5	32.1	32.2	31.6	31.7	31.8	8.3	8.4
9.6	9.7	9.8	20.3	20.4	20.5	25.1	25.2	26.6	26.7	26.8	32.3	32.4	32.5	38.1	38.2	8.6	8.7
8.5	21.1	21.2	20.6	20.7	20.8	25.3	25.4	25.5	33.1	33.2	32.6	32.7	32.8	38.3	38.4	38.5	22.1
8.8	21.3	21.4	21.5	24.1	24.2	25.6	25.7	25.8	33.3	33.4	33.5	37.1	37.2	38.6	38.7	38.8	22.3
22.2	21.6	21.7	21.8	24.3	24.4	24.5	34.1	34.2	33.6	33.7	33.8	37.3	37.4	37.5	39.1	39.2	22.6
22.4	22.5	23.1	23.2	24.6	24.7	24.8	34.3	34.4	34.5	36.1	36.2	37.6	37.7	37.8	39.3	39.4	39.5
22.7	22.8	23.3	23.4	23.5	35.1	35.2	34.6	34.7	34.8	36.3	36.4	36.5	40.1	40.2	39.6	39.7	39.8
3.1	3.2	23.6	23.7	23.8	35.3	35.4	35.5	14.1	14.2	36.6	36.7	36.8	40.3	40.4	40.5	*a*	*b*
3.3	3.4	3.5	4.1	4.2	35.6	35.7	35.8	14.3	14.4	14.5	15.1	15.2	40.6	40.7	40.8	*b*	*a*

a — BLK (темный модуль); *b* — WHT (светлый модуль)

Рисунок F.17 — Размещение кодовых слов в квадратной координатной матрице размером 18 × 18[1)]

[1)] *Размещение бита 8.8 приведено в соответствие с рисунком F.17 ISO/IEC 16022:2006, поскольку рисунок F.17 ISO/IEC 16022:2024 содержит ошибку.*

2.1	2.2	3.1	3.7	3.8	4.3	4.4	4.5	13.1	13.2	14.6	14.7	14.8	15.3	15.4	15.5	32.1	32.2	23.4	23.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	16.1	16.2	15.6	15.7	15.8	32.3	32.4	32.5	23.6
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	16.3	16.4	16.5	31.1	31.2	32.6	32.7	32.8	23.7
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	17.1	17.2	16.6	16.7	16.8	31.3	31.4	31.5	33.1	33.2	23.8
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	17.3	17.4	17.5	30.1	30.2	31.6	31.7	31.8	33.3	33.4	33.5
7.2	6.6	6.7	6.8	11.3	11.4	11.5	18.1	18.2	17.6	17.7	17.8	30.3	30.4	30.5	34.1	34.2	33.6	33.7	33.8
7.4	7.5	10.1	10.2	11.6	11.7	11.8	18.3	18.4	18.5	29.1	29.2	30.6	30.7	30.8	34.3	34.4	34.5	1.1	1.2
7.7	7.8	10.3	10.4	10.5	19.1	19.2	18.6	18.7	18.8	29.3	29.4	29.5	35.1	35.2	34.6	34.7	34.8	1.3	1.4
9.1	9.2	10.6	10.7	10.8	19.3	19.4	19.5	28.1	28.2	29.6	29.7	29.8	35.3	35.4	35.5	45.1	45.2	1.6	1.7
9.3	9.4	9.5	20.1	20.2	19.6	19.7	19.8	28.3	28.4	28.5	36.1	36.2	35.6	35.7	35.8	45.3	45.4	45.5	7.1
9.6	9.7	9.8	20.3	20.4	20.5	27.1	27.2	28.6	28.7	28.8	36.3	36.4	36.5	44.1	44.2	45.6	45.7	45.8	7.3
8.5	21.1	21.2	20.6	20.7	20.8	27.3	27.4	27.5	37.1	37.2	36.6	36.7	36.8	44.3	44.4	44.5	46.1	46.2	7.6
8.8	21.3	21.4	21.5	26.1	26.2	27.6	27.7	27.8	37.3	37.4	37.5	43.1	43.2	44.6	44.7	44.8	46.3	46.4	46.5
22.2	21.6	21.7	21.8	26.3	26.4	26.5	38.1	38.2	37.6	37.7	37.8	43.3	43.4	43.5	47.1	47.2	46.6	46.7	46.8
22.4	22.5	25.1	25.2	26.6	26.7	26.8	38.3	38.4	38.5	42.1	42.2	43.6	43.7	43.8	47.3	47.4	47.5	8.1	8.2
22.7	22.8	25.3	25.4	25.5	39.1	39.2	38.6	38.7	38.8	42.3	42.4	42.5	48.1	48.2	47.6	47.7	47.8	8.3	8.4
24.1	24.2	25.6	25.7	25.8	39.3	39.4	39.5	41.1	41.2	42.6	42.7	42.8	48.3	48.4	48.5	50.1	50.2	8.6	8.7
24.3	24.4	24.5	40.1	40.2	39.6	39.7	39.8	41.3	41.4	41.5	49.1	49.2	48.6	48.7	48.8	50.3	50.4	50.5	22.1
24.6	24.7	24.8	40.3	40.4	40.5	3.1	3.2	41.6	41.7	41.8	49.3	49.4	49.5	14.1	14.2	50.6	50.7	50.8	22.3
23.1	23.2	23.3	40.6	40.7	40.8	3.3	3.4	3.5	4.1	4.2	49.6	49.7	49.8	14.3	14.4	14.5	15.1	15.2	22.6

Рисунок F.18 — Размещение кодовых слов в квадратной координатной матрице размером 20 × 20[1)]

[1)] *Размещение бита 8.8 приведено в соответствие с рисунком F.18 ISO/IEC 16022:2006, поскольку рисунок F.18 ISO/IEC 16022:2024 содержит ошибку.*

2.1	2.2	3.1	3.7	3.8	4.3	4.4	4.5	13.1	13.2	14.6	14.7	14.8	15.3	15.4	15.5	32.1	32.2	23.4	23.5	23.6	23.7
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8	13.3	13.4	13.5	16.1	16.2	15.6	15.7	15.8	32.3	32.4	32.5	33.1	33.2	23.8
2.6	2.7	2.8	5.3	5.4	5.5	12.1	12.2	13.6	13.7	13.8	16.3	16.4	16.5	31.1	31.2	32.6	32.7	32.8	33.3	33.4	33.5
1.5	6.1	6.2	5.6	5.7	5.8	12.3	12.4	12.5	17.1	17.2	16.6	16.7	16.8	31.3	31.4	31.5	34.1	34.2	33.6	33.7	33.8
1.8	6.3	6.4	6.5	11.1	11.2	12.6	12.7	12.8	17.3	17.4	17.5	30.1	30.2	31.6	31.7	31.8	34.3	34.4	34.5	1.1	1.2
7.2	6.6	6.7	6.8	11.3	11.4	11.5	18.1	18.2	17.6	17.7	17.8	30.3	30.4	30.5	35.1	35.2	34.6	34.7	34.8	1.3	1.4
7.4	7.5	10.1	10.2	11.6	11.7	11.8	18.3	18.4	18.5	29.1	29.2	30.6	30.7	30.8	35.3	35.4	35.5	49.1	49.2	1.6	1.7
7.7	7.8	10.3	10.4	10.5	19.1	19.2	18.6	18.7	18.8	29.3	29.4	29.5	36.1	36.2	35.6	35.7	35.8	49.3	49.4	49.5	7.1
9.1	9.2	10.6	10.7	10.8	19.3	19.4	19.5	28.1	28.2	29.6	29.7	29.8	36.3	36.4	36.5	48.1	48.2	49.6	49.7	49.8	7.3
9.3	9.4	9.5	20.1	20.2	19.6	19.7	19.8	28.3	28.4	28.5	37.1	37.2	36.6	36.7	36.8	48.3	48.4	48.5	50.1	50.2	7.6
9.6	9.7	9.8	20.3	20.4	20.5	27.1	27.2	28.6	28.7	28.8	37.3	37.4	37.5	47.1	47.2	48.6	48.7	48.8	50.3	50.4	50.5
8.5	21.1	21.2	20.6	20.7	20.8	27.3	27.4	27.5	38.1	38.2	37.6	37.7	37.8	47.3	47.4	47.5	51.1	51.2	50.6	50.7	50.8
8.8	21.3	21.4	21.5	26.1	26.2	27.6	27.7	27.8	38.3	38.4	38.5	46.1	46.2	47.6	47.7	47.8	51.3	51.4	51.5	8.1	8.2
22.2	21.6	21.7	21.8	26.3	26.4	26.5	39.1	39.2	38.6	38.7	38.8	46.3	46.4	46.5	52.1	52.2	51.6	51.7	51.8	8.3	8.4
22.4	22.5	25.1	25.2	26.6	26.7	26.8	39.3	39.4	39.5	45.1	45.2	46.6	46.7	46.8	52.3	52.4	52.5	58.1	58.2	8.6	8.7
22.7	22.8	25.3	25.4	25.5	40.1	40.2	39.6	39.7	39.8	45.3	45.4	45.5	53.1	53.2	52.6	52.7	52.8	58.3	58.4	58.5	22.1
24.1	24.2	25.6	25.7	25.8	40.3	40.4	40.5	44.1	44.2	45.6	45.7	45.8	53.3	53.4	53.5	57.1	57.2	58.6	58.7	58.8	22.3
24.3	24.4	24.5	41.1	41.2	40.6	40.7	40.8	44.3	44.4	44.5	54.1	54.2	53.6	53.7	53.8	57.3	57.4	57.5	59.1	59.2	22.6
24.6	24.7	24.8	41.3	41.4	41.5	43.1	43.2	44.6	44.7	44.8	54.3	54.4	54.5	56.1	56.2	57.6	57.7	57.8	59.3	59.4	59.5
23.1	42.1	42.2	41.6	41.7	41.8	43.3	43.4	43.5	55.1	55.2	54.6	54.7	54.8	56.3	56.4	56.5	60.1	60.2	59.6	59.7	59.8
23.2	42.3	42.4	42.5	3.1	3.2	43.6	43.7	43.8	55.3	55.4	55.5	14.1	14.2	56.6	56.7	56.8	60.3	60.4	60.5	*a*	*b*
23.3	42.6	42.7	42.8	3.3	3.4	3.5	4.1	4.2	55.6	55.7	55.8	14.3	14.4	14.5	15.1	15.2	60.6	60.7	60.8	*b*	*a*

a — BLK (темный модуль); *b* — WHT (светлый модуль)

Рисунок F.19 — Размещение кодовых слов в квадратной координатной матрице размером 22 × 22[1)]

[1)] *Размещение бита 8.8 приведено в соответствие с рисунком F.19 ISO/IEC 16022:2006, поскольку рисунок F.19 ISO/IEC 16022:2024 содержит ошибку.*

Приложение G
(обязательное)

Качество печати символов Data Matrix — аспекты, связанные с особенностями символики

G.1 Общие положения

Из-за различий в структурах, установленных символикой, и в рекомендуемых алгоритмах декодирования, влияние некоторых параметров на эффективность считывания символов может варьироваться для различных символик. ISO/IEC 15415 и ISO/IEC 29158 предусматривают для спецификаций символик установление градации определенных признаков, связанных с особенностями символики. В настоящем приложении установлена методика оценивания повреждения фиксированного шаблона (Fixed Pattern Damage), которую следует использовать при применении ISO/IEC 15415 или ISO/IEC 29158 к символике Data Matrix.

При испытаниях качества нанесения символа с использованием ISO/IEC 29158 все ссылки на ISO/IEC 15415 следует заменить на ISO/IEC 29158. Термин «модуляция» должен быть заменен на термин «модуляция ячейки», а термин «MOD» — на «CMOD».

G.2 Интерполяция между значениями классов

Существенным изменением, которое вводится в настоящем издании стандарта, является использование «непрерывной градации», при которой классы представляются в десятых долях значения класса вместо целого числа, представляющего все значение класса, как в предыдущем издании. Это уменьшает тревожные колебания значений классов, когда небольшие изменения в измерениях приводят к переходу значения класса между целыми значениями классов. Значения классов должны округляться до ближайшей десятой части, чтобы граница между целыми числами оставалась неизменной с предыдущего издания настоящего стандарта. Соответственно, во всех случаях можно преобразовать значение класса из данного издания в значение класса из предыдущего издания, просто отбросив десятую долю значения класса.

G.3 Повреждение фиксированного шаблона символики Data Matrix

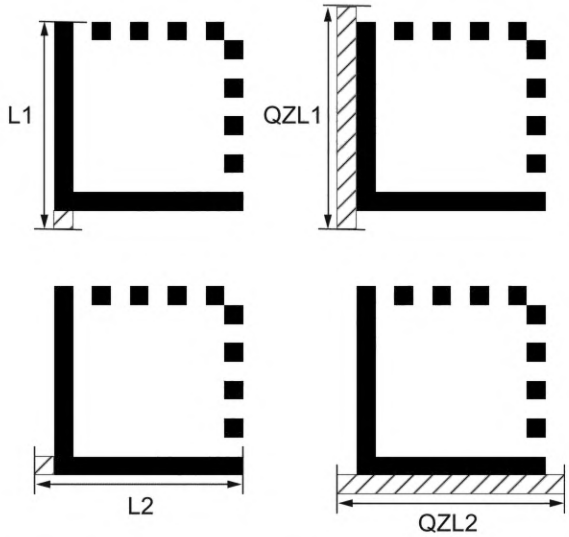
G.3.1 Параметры, подлежащие оценке

Параметры фиксированного шаблона, подлежащего оценке, находятся внутри периметра символа шириной в один модуль и окружающей символ свободной зоны шириной не менее одного модуля (или более, в соответствии с установленным значением в стандарте по применению). В символах большой емкости, содержащих направляющий шаблон (символы квадратной формы размером 32 × 32 модуля и более или символы прямоугольной формы размером 8 × 32, 12 × 36 и более), направляющий шаблон также является частью фиксированного шаблона. Левая и нижняя стороны символа должны образовывать однородную темную фигуру в виде буквы «L» шириной один модуль, а правая и верхняя стороны должны состоять из чередующихся одиночных темных и светлых модулей (называемых дорожкой синхронизации). Направляющие штрихи и внутренние дорожки синхронизации направляющего шаблона должны представлять собой по всему символу сплошные темные полосы шириной один модуль и последовательности из чередующихся одиночных темных и светлых модулей соответственно. При оценивании по параметру повреждение фиксированного шаблона (Fixed Pattern Damage) следует учитывать кроме общего числа поврежденных модулей также концентрацию повреждений.

G.3.2 Проведение оценки внешней L-образной структуры фиксированного шаблона

Повреждение каждой стороны L-образной структуры следует оценивать, основываясь на модуляции отдельных составляющих ее модулей. Соответствующие измерения проводят по всей длине каждой из сторон L-образной структуры и соседних свободных зон.

На рисунке G.1 приведены четыре сегмента L1, L2, QZL1 и QZL2. Сегмент L1 является вертикальной частью L-образной структуры, продленной на один модуль в свободную зону, смежную с углом L-образной структуры. Сегмент L2 является горизонтальной частью L-образной структуры, продленной на один модуль в свободную зону, смежную с углом L-образной структуры. Сегменты QZL1 и QZL2 являются частями свободной зоны, смежными с сегментами L1 и L2 соответственно и продленные на один модуль за внешние (удаленные от угла) концы L1 и L2 согласно тому, как это показано на затемненных участках рисунка G.1. Модуль, находящийся на пересечении сторон угла L-образной структуры, входит в оба сегмента L1 и в L2, так же, как и модуль на пересечении сегментов QZL1 и QZL2.



L1 — вертикальная часть L-образной структуры; L2 — горизонтальная часть L-образной структуры; QZL1 — часть свободной зоны, примыкающая к L1; QZL2 — часть свободной зоны, примыкающая к L2

Рисунок G.1 — Сегменты внешней L-образной структуры и свободной зоны фиксированного шаблона

Все сегменты поочередно должны быть подвергнуты следующей процедуре:

a) для каждого модуля определяют класс модуляции с использованием методики по ISO/IEC 15415. Так как заранее известно, какой модуль следует считать темным или светлым, любой модуль, представляющий собой модуль противоположного цвета должен приводить к классу модуляции ноль;

b) к каждому классу модуляции применяют метод оценки параметров, приведенный в ISO/IEC 15415:

1) для каждой стороны L-образной структуры (сегменты L1 и L2 на рисунке G.1) и каждой части свободной зоны (сегменты QZL1 и QZL2, прилегающие соответственно к сегментам L1 и L2 на рисунке G.1) считают, что все модули, не достигшие этого или более высокого класса, являются ошибками модулей, и выводят условный класс повреждения на основе пороговых значений классов, приведенных в таблице G.1. Принимают наименьшее из значений класса модуляции и условного класса повреждения;

2) классом любого сегмента следует считать наивысший полученный класс для всех значений класса модуляции;

c) в дополнение к этому для квадратных и прямоугольных символов с более чем одной областью данных повторяют этапы, указанные выше в перечислениях a) и b), где сегменты L1 и L2 начинаются с модуля в свободной зоне и заканчиваются модулем в области дорожки синхронизации той же области данных, а сегменты QZL1 и QZL2 состоят из свободой зоны, смежной с указанными сегментами L1 и L2, как указано на рисунке G.1. Другими словами, обрабатывают нижнюю левую область данных, как если бы это был символ с одной областью данных. Если значение этого класса ниже, чем полученное для сегментов L1, L2, QZL1 и QZL2 на этапах, указанных в перечислениях a) и b), то осуществляют замену значения класса, полученное в перечислениях a) и b), на данное значение класса;

d) кроме того, для сегментов L1 и L2 проводят проверку, чтобы установить, что все поврежденные участки разделены по крайней мере четырьмя правильными модулями и что ни один из поврежденных участков не превышает трех модулей; если этот проверка не пройдена, значение класса, полученное на вышеуказанных этапах, должно быть снижено до 0 для этого значения класса модуляции. Берут наименьшее из значений класса модуляции и условного класса повреждения;

Таблица G.1 — Пороговые значения условных классов повреждения

Повреждение модулей, %	Класс
0 %	4
> 0 % и ≤ 9 %	От 3,0 до 3,9
> 9 % и ≤ 13 %	От 2,0 до 2,9
> 13 % и ≤ 17 %	От 1,0 до 1,9
> 17 % и ≤ 21 %	От 0,0 до 0,9
> 21 %	0

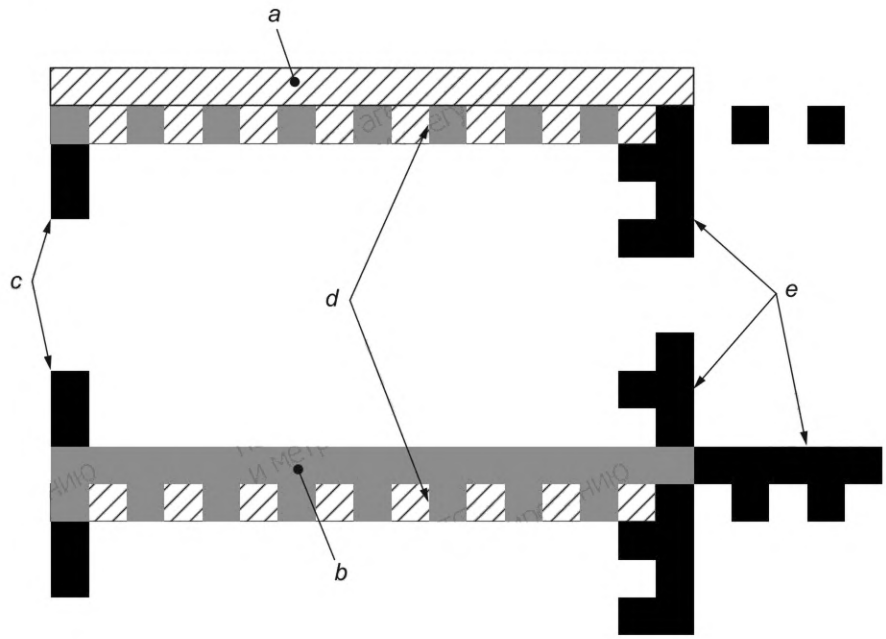
е) класс параметра повреждения фиксированного шаблона для сегмента должен соответствовать наибольшему из полученных значений модуляции.

G.3.3 Проведение оценки сегментов дорожки синхронизации и смежных однородных областей

В настоящем подразделе установлен метод измерения повреждений внутренних направляющих шаблонов (при их наличии), а также внешних дорожек синхронизации и примыкающих свободных зон. Данные испытания проводят отдельно для каждого сегмента внутренних направляющих шаблонов, дорожек синхронизации и соответствующих свободных зон, которые служат границами области данных символа, или отдельных областей данных для больших символов. Каждый сегмент состоит из частей дорожек синхронизации и однородных областей (которые являются либо частью свободной зоны, либо частью темной полосы внутреннего направляющего шаблона).

Сегмент дорожки синхронизации начинается с модуля в стороне L-образной структуры или внутреннего направляющего шаблона. Он продолжается и включает модуль, предшествующий либо свободной зоне, либо следующему внутреннему направляющему шаблону. Соседний модуль в свободной зоне или направляющий шаблон не включается в дорожку синхронизации (см. рисунок G.2).

Сегмент однородной области начинается рядом с первым модулем связанного сегмента дорожки синхронизации. Если другой конец этого сегмента соседствует со свободной зоной, то указанный сегмент однородной области завершается модулем, соседним с последним модулем связанного сегмента дорожки синхронизации (см. рисунок G.6). Если это не так, он заканчивается первым модулем соседнего однородного сегмента, имеющего тот же цвет (см. рисунок G.7).



a — однородная область (светлая); *b* — однородная область (темная); *c* — сторона L-образной структуры; *d* — дорожка синхронизации; *e* — направляющие шаблоны

Примечание — На рисунке G.2 приведен сегмент внутреннего направляющего шаблона, завершающийся у другого сегмента внутреннего направляющего шаблона того же цвета.

Рисунок G.2 — Структура сегмента внешней дорожки синхронизации и сегмента внутреннего направляющего шаблона

Для каждого сегмента внешней дорожки синхронизации или сегмента внутреннего направляющего шаблона (для символов с несколькими сегментами) повреждения вычисляют с помощью следующей процедуры:

а) для всех сегментов дорожки синхронизации в черно-белом (двоичном) изображении как внешних (смежных со свободной зоной), так и внутренних (смежных с однородной линией внутреннего направляющего шаблона) подсчитывают число переходов *T*s на стороне дорожки синхронизации и число переходов *T*s на стороне однородной линии и вычисляют класс коэффициента переходов (transition ratio) по формулам (G.1) и (G.2):

$$Ts' = \text{Max}\,(0,\ Ts - 1), \tag{G.1}$$

$$R_T = Ts' / Tc, \tag{G.2}$$

где R_T — коффициент переходов.

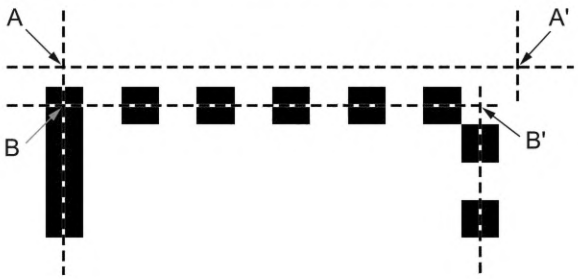
Определяют класс коэффициента переходов в соответствии с таблицей G.2.

Таблица G.2 — Классы коэффициента переходов

R_T (коэффициент переходов)	Класс
≤ 0,06	4
> 0,06 и ≤ 0,095	От 3,0 до 3,9
> 0,095 и ≤ 0,13	От 2,0 до 2,9
> 0,13 и ≤ 0,165	От 1,0 до 1,9
> 0,165 и ≤ 0,2	От 0,0 до 0,9
> 0,2	0

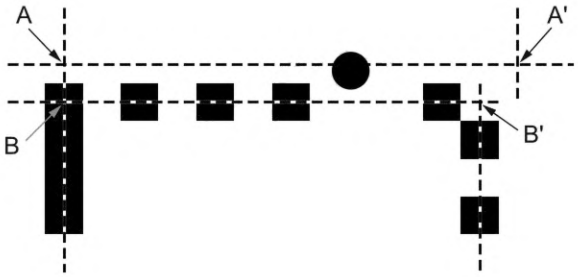
Конечными точками линий, по которым проводят подсчет числа переходов, являются пересечения линий сетки, нанесенные согласно рекомендуемому алгоритму декодирования, в первом и последнем модуле дорожки синхронизации или однородной области.

В качестве примера идеальный символ с нулевым числом переходов в свободной зоне приведен на рисунке G.3, поврежденный символ с двумя переходами в свободной зоне приведен на рисунке G.4;



--- — линия сетки центров модулей из рекомендованного алгоритма декодирования; A — A′ — пересечения, определяющие начало и конец однородной области; B — B′ — пересечения, определяющие начало и конец дорожки синхронизации

Рисунок G.3 — Переходы в идеальном символе



--- — линия сетки центров модулей из рекомендованного алгоритма декодирования; A — A′ — пересечения, определяющие начало и конец однородной области; B — B′ — пересечения, определяющие начало и конец дорожки синхронизации

Рисунок G.4 — Переходы в поврежденном символе

b) условный класс повреждения: вычисляют класс модуляции для каждого модуля на основе значений, приведенных в ISO/IEC 15415. Так как заранее известно, является данный модуль темным или светлым, любой модуль, предполагаемый темным, коэффициент отражения которого выше глобального порога, а также любой модуль, предполагаемый светлым, коэффициент отражения которого ниже глобального порога, получает класс модуляции, равный нулю;

c) для каждого значения класса модуляции: предполагают, что все модули, не достигшие значения этого класса или более высокого класса, являются поврежденными модулями, и вычисляют условный класс повреждения на основе следующих трех оценок;

d) оценка регулярности дорожки синхронизации: для каждого сегмента дорожки синхронизации берут группы, состоящие из пяти смежных модулей, и, двигаясь вдоль сегмента с шагом в один модуль, проверяют, чтобы в

любой группе из пяти смежных модулей было не более двух поврежденных модулей. Если это условие выполняется, значение класса регулярности дорожки синхронизации должно быть равно четырем, в противном случае оно должно быть равно нулю;

e) оценка наличия повреждений дорожки синхронизации: для каждого сегмента подсчитывают число некорректных модулей дорожки синхронизации. Определение доли повреждений *P* — отношения числа некорректных модулей к общему числу модулей в сегменте (в процентах) — в результате даст классы повреждения дорожки синхронизации, приведенные в таблице G.3;

f) оценка фиксированного шаблона однородной области: для каждого сегмента подсчитывают число некорректных модулей в однородных областях (темные линии внутреннего направляющего шаблона или внешней свободной зоны), смежных с дорожкой синхронизации. Определение доли повреждений *P* — отношения числа некорректных модулей к общему числу модулей в сегменте (в процентах) — в результате даст классы повреждений (таблица G.3);

Таблица G.3 — Оценка повреждений сегментов дорожек синхронизации и сегментов однородных областей

Доля повреждений *P*, %	Класс
≤ 10 %	4
> 10 % и ≤ 15 %	От 3,0 до 3,9
> 15 % и ≤ 20 %	От 2,0 до 2,9
> 20 % и ≤ 25 %	От 1,0 до 1,9
> 25 % и ≤ 30 %	От 0,0 до 0,9
> 30 %	0

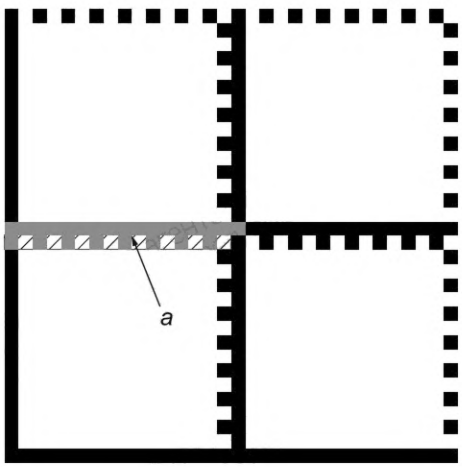
g) для каждого значения класса выбирают наименьшее значение класса модуляции, класса регулярности дорожки синхронизации, класса доли повреждений дорожки синхронизации и класса доли повреждений фиксированного шаблона однородной области в процентах;

h) условный класс повреждения для сегмента должен соответствовать наибольшему значению класса модуляции из всех значений класса модуляции;

i) класс повреждения фиксированного шаблона для сегмента должен соответствовать наименьшему из значений класса коэффициента переходов и условного класса повреждений;

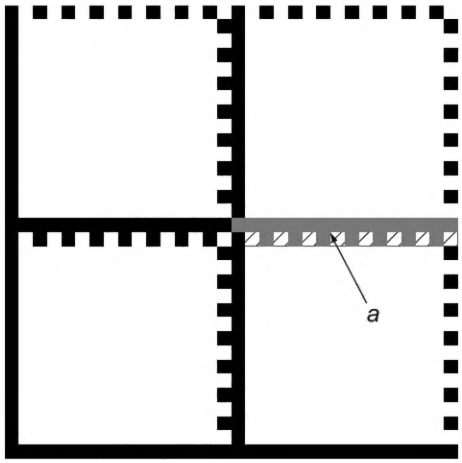
j) полный класс повреждения фиксированных шаблонов для сегментов дорожки синхронизации и смежной однородной области соответствует наименьшему значению из классов, полученных для каждого отдельного сегмента.

Сегменты внутреннего направляющего шаблона, включающие в себя часть дорожки синхронизации и часть однородной области, для которых проводят оценки коэффициента переходов на регулярность и наличие повреждений фиксированных шаблонов однородных областей, представлены в качестве примеров на рисунках G.5 и G.6 и выделены затенением для каждого типа внутренних границ сегмента.



a — сегмент внутреннего направляющего шаблона

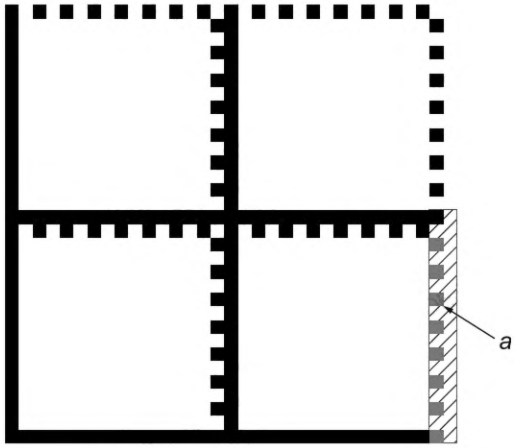
Рисунок G.5 — Сегменты внутреннего направляющего шаблона, ограниченные внешней однородной областью и внутренней однородной областью



a — сегмент внутреннего направляющего шаблона

Рисунок G.6 — Сегменты внутреннего направляющего шаблона, ограниченные внутренней однородной областью и внутренней однородной областью и внешней дорожкой синхронизации

Сегмент внешней дорожки синхронизации и смежной свободной зоны, для которых производят проверки коэффициента переходов на регулярность и наличие повреждений фиксированных шаблонов однородных областей, представлен в качестве примера на рисунке G.7 и выделен затенением.



a — сегмент дорожки синхронизации

Рисунок G.7 — Сегмент внешней дорожки синхронизации



Рисунок G.8 — Пример 37 модулей, оцененных для стороны L-образной структуры символа размером 36 × 36 модулей

Пример оценивания сегмента L1 символа размером 36 × 36 модулей, в котором SC = 89 % и GT = 51 % приведен на рисунке G.8. Значения коэффициентов отражения и модуляции, а также классы модуляции приведены в таблице G.4 для всех 37 модулей (от 0 до 36) этого сегмента. Приставленный модуль свободной зоны, примыкающий к углу L-образной структуры, обозначен как модуль 0.

Таблица G.4 — Пример оценивания параметра модуляции для сегмента из 36 модулей

Номер модуля	0	1	2	3	4	5	6	7	8	9
Коэффициент отражения, %	84	15	13	13	13	9	11	84	11	10
MOD (значение модуляции)	74	80	86	86	86	94	90	(74)	90	92
Класс модуляции	4,0	4.0	4.0	4,0	4,0	4,0	4,0	0,0	4,0	4,0
Номер модуля		10	11	12	13	14	15	16	17	18
Коэффициент отражения, %		9	11	70	13	12	15	11	11	11
MOD (значение модуляции)		94	90	(42)	86	88	80	90	90	90
Класс модуляции		4,0	4,0	0,0	4,0	4,0	4,0	4,0	4,0	4,0
Номер модуля		19	20	21	22	23	24	25	26	27
Коэффициент отражения, %		27	11	14	10	12	50	12	11	14
MOD (значение модуляции)		54	90	83	92	88	2	88	90	83
Класс модуляции		4,0	4,0	4,0	4,0	4,0	0,0	4,0	4,0	4,0
Номер модуля		28	29	30	31	32	33	34	35	36
Коэффициент отражения, %		13	12	37	13	12	13	11	13	12
MOD (значение модуляции)		86	88	31	86	88	86	90	86	88
Класс модуляции		4,0	4,0	2,1	4,0	4,0	4,0	4,0	4,0	4,0

Следует обратить внимание на то, что модули 7 и 12 — светлые, а модуль 24 и, в меньшей степени, модуль 30 имеют низкую модуляцию.

На основе этих значений оценивают сегмент (см. таблицу G.5).

Таблица G.5 — Пример оценивания сегмента

Класс MOD	Число модулей	Общее число модулей	Оставшиеся поврежденные модули	Доля поврежденных модулей, %	Условный класс повреждений	Наименьшее значение класса
4	33	33	4	10,8	2,5	2,5
3,9	0	33	4	10,8	2,5	2,5
3,8	0	33	4	10,8	2,5	2,5
3,7	0	33	4	10,8	2,5	2,5
3,6	0	33	4	10,8	2,5	2,5
3,5	0	33	4	10,8	2,5	2,5
3,4	0	33	4	10,8	2,5	2,5
3,3	0	33	4	10,8	2,5	2,5
3,2	0	33	4	10,8	2,5	2,5
3,1	0	33	4	10,8	2,5	2,5
3,0	0	33	4	10,8	2,5	2,5
2,9	0	33	4	10,8	2,5	2,5
2,8	0	33	4	10,8	2,5	2,5
2,7	0	33	4	10,8	2,5	2,5
2,6	0	33	4	10,8	2,5	2,5

Окончание таблицы G.5

Класс MOD	Число модулей	Общее число модулей	Оставшиеся поврежденные модули	Доля поврежденных модулей, %	Условный класс повреждений	Наименьшее значение класса
2,5	0	33	4	10,8	2,5	2,5
2,4	0	33	4	10,8	2,5	2,4
2,3	0	33	4	10,8	2,5	2,3
2,2	0	33	4	10,8	2,5	2,2
2,1	1	34	3	8,1	3,1	2,1
2	0	34	3	8,1	3,1	2
1,9	0	34	3	8,1	3,1	1,9
1,8	0	34	3	8,1	3,1	1,8
1,7	0	34	3	8,1	3,1	1,7
1,6	0	34	3	8,1	3,1	1,6
1,5	0	34	3	8,1	3,1	1,5
1,4	0	34	3	8,1	3,1	1,4
1,3	0	34	3	8,1	3,1	1,3
1,2	0	34	3	8,1	3,1	1,2
1,1	0	34	3	8,1	3,1	1,1
1	0	34	3	8,1	3,1	1
0,9	0	34	3	8,1	3,1	0,9
0,8	0	34	3	8,1	3,1	0,8
0,7	0	34	3	8,1	3,1	0,7
0,6	0	34	3	8,1	3,1	0,6
0,5	0	34	3	8,1	3,1	0,5
0,4	0	34	3	8,1	3,1	0,4
0,3	0	34	3	8,1	3,1	0,3
0,2	0	34	3	8,1	3,1	0,2
0,1	0	34	3	8,1	3,1	0,1
0	3	37	0	0	4	0
Окончательный класс сегмента соответствует наибольшему значению, указанному в графе «Наименьшее значение класса»						2,5

G.3.4 Вычисление и оценивание по классам рассредоточенных повреждений фиксированных шаблонов

Дополнительно к оценке отдельных сегментов следует также вычислить *DFPD* (distributed fixed pattern damage grade, класс рассредоточенных повреждений фиксированных шаблонов), чтобы учесть совокупный эффект повреждений, которые имеют относительно незначительное влияние в отдельных сегментах, но затрагивают несколько сегментов. В основу данного вычисления положено определение среднего значения классов сегментов L1, L2, QZL1, QZL2 и полного класса сегмента дорожки синхронизации и смежной однородной области.

Если определены классы всех сегментов, вычисляют класс рассредоточенных повреждений фиксированных шаблонов G_{DFPD} по формуле

$$G_{DFPD} = \sum G_S / 5, \quad \text{(G.3)}$$

где G_{DFPD} — класс рассредоточенных повреждений фиксированных шаблонов;

G_S — классы пяти сегментов.

Назначают класс *DFPD* в соответствии с таблицей G.6.

Класс повреждений фиксированного шаблона для символа должен быть меньшим из пяти классов сегментов и класса *DFPD*.

Таблица G.6 — Оценивание *DFPD*

Среднее значение классов пяти сегментов	Класс
= 4	4
≥ 3,5 и < 4	От 3,0 до 3,9
≥ 3,0 и < 3,5	От 2,0 до 2,9
≥ 2,5 и < 3,0	От 1,0 до 1,9
≥ 2,0 и < 2,5	От 0,0 до 0,9
< 2,0	0

Пример 1 — Если четыре из пяти сегментов имеют класс 4 и один сегмент имеет класс 1, в этом случае:

(4 × 4) + (1 × 1) = 17.

Следовательно, G_{DFPD} = 17 / 5 = 3,4.

По таблице G.6 среднему значению 3,4 соответствует класс 2,8. Наименьшим из этих шести классов является класс 1, и класс повреждений фиксированного шаблона соответственно принимает значение 1.

Пример 2 — Если три из пяти сегментов имеют класс 4, один сегмент — класс 3 и один сегмент имеет класс 1, в этом случае:

(3 × 4) + (1 × 3) + (1 × 1) = 16.

Следовательно, G_{DFPD} = 16 / 5 = 3,2.

По таблице G.6 среднему значению 3,2 соответствует класс 2,4. Наименьшим из этих шести классов является класс 1, и класс повреждений фиксированного шаблона соответственно принимает значение 1.

Пример 3 — Если все пять сегментов имеют класс 3, в этом случае:

5 × 3 = 15.

Следовательно, G_{DFPD} = 15 / 5 = 3,0.

По таблице G.6 среднему значению 3,0 соответствует класс 2. Наименьшим из этих шести классов является 2, и класс повреждений фиксированного шаблона соответственно принимает значение 2.

G.4 Приращение/сокращение при печати

Приращение/сокращение при печати следует измерять и оценивать в соответствии с ISO/IEC 15415, используя все горизонтальные дорожки синхронизации для определения приращения/сокращения при печати по горизонтальной оси и все вертикальные дорожки синхронизации для определения приращения/сокращения при печати по вертикальной оси.

Приложение Н
(обязательное)

Идентификатор символики

ISO/IEC 15424[1)] предоставляет единую методологию для сообщения о считанной символике, наборе вариантов обработки устройством считывания и других особенностях, встречающихся в символике.

Идентификатором символики Data Matrix является:

]dm

где] — знак флага идентификатора символики (знак набора знаков ASCII с десятичным значением 93);

d — знак кода для символики Data Matrix;

m — знак-модификатор с одним из значений, установленных в таблице Н.1.

Таблица Н.1 — Значения вариантов идентификатора символики для символа Data Matrix

Значение m	Вариант обработки
0	Символ Data Matrix ECC000-ECC140
1	Символ Data Matrix
2	Символ Data Matrix со знаком FNC1 в 1-й или 5-й позиции
3	Символ Data Matrix со знаком FNC1 во 2-й или 6-й позиции
4	Символ Data Matrix с поддержкой протокола интерпретаций ECI
5	Символ Data Matrix со знаком FNC1 в 1-й или 5-й позиции и поддержкой протокола интерпретаций ECI
6	Символ Data Matrix со знаком FNC1 во 2-й или 6-й позиции и поддержкой протокола интерпретаций ECI

Примечание 1 — Допустимые значения m: 0, 1, 2, 3, 4, 5 и 6.

Примечание 2 — Значение 0 характеризует символы от версии Data Matrix ECC000 до версии Data Matrix ECC140. Указанные ранее действовавшие версии приведены в [5]. Версии этих символов могут быть визуально определены по факту наличия в правом верхнем углу символа элемента с тем же коэффициентом отражения, что и у L-образной структуры шаблона поиска.

[1)] *См.* [7].

Приложение I
(справочное)

Пример кодирования

I.1 Общие положения

В этом примере данными пользователя, подлежащими кодированию, является строка «123456» длиной 6 знаков.

I.2 Этап 1: кодирование данных

Представление в 7-битовых знаках набора знаков ASCII:

Знаки данных:	'1'	'2'	'3'	'4'	'5'	'6'
Десятичные значения знаков:	49	50	51	52	53	54

В схеме кодирования ASCII осуществляется преобразование шести вышеуказанных знаков в три байта с использованием следующей формулы для пар цифр:

значение кодового слова = (численное значение пары цифр) + 130.

Подробные сведения о вычислении:

"12" = 12 + 130 = 142

"34" = 34 + 130 = 164

"56" = 56 + 130 = 186

Поток данных после кодирования:

десятичные значения: 142 164 186

В соответствии с таблицей 10 размещают три кодовых слова данных в символе размером 10 × 10, который необходимо дополнить пятью кодовыми словами исправления ошибок. Если кодируемые данные не полностью заполняют область данных, то должны быть закодированы дополнительные знаки-заполнители.

I.3 Этап 2: проверка и исправление ошибок

Кодовые слова исправления ошибок генерируют с использованием алгоритма Рида—Соломона и добавляют к потоку кодированных данных:

Номер кодового слова:	1	2	3	4	5	6	7	8
Десятичное значение:	142	164	186	114	25	5	88	102
Шестнадцатеричное значение:	8E	A4	BA	72	19	05	58	66
		Данные			Проверка			

Процесс генерации кодовых слов исправления ошибок описан в приложении E. Пример процедуры расчета кодовых слов исправления ошибок приведен в E.3.

I.4 Этап 3: размещение модулей в матрице

Сформированные кодовые слова после этапа 2 размещают в двоичной матрице как знаки символа согласно алгоритму, указанному в 7.7.2 (также см. рисунок I.1):

1	0	0	1	0	1	1	0
1	0	0	0	0	0	1	0
1	0	0	0	1	1	1	0
1	0	0	0	0	1	0	0
0	0	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
0	0	1	1	1	0	1	0

Рисунок I.1 — Позиционирование модулей в матрице

I.5 Этап 4: действительный символ

Окончательный вид символа Data Matrix формируют путем добавления модулей шаблона поиска и преобразования двоичных единиц в темные модули и двоичных нулей — в светлые, как показано на рисунке I.2.

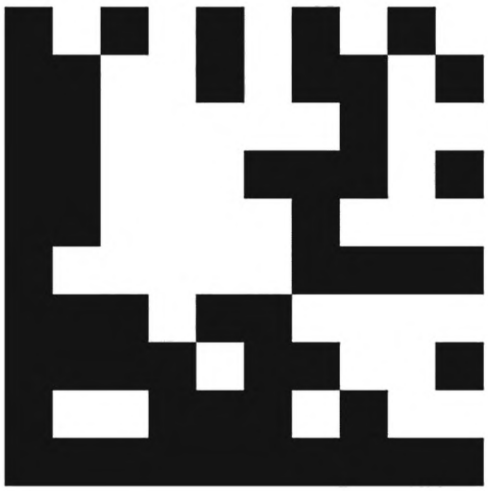


Рисунок I.2 — Окончательный вид символа Data Matrix, кодирующего строку «123456»

Приложение J
(рекомендуемое)

Кодирование данных с использованием минимального числа знаков символа

Одни и те же данные могут быть представлены различными символами Data Matrix путем использования различных схем кодирования.

Следующий алгоритм обычно порождает самый короткий поток кодовых слов:

a) кодирование начинают по схеме кодирования ASCII;

b) при кодировании по схеме кодирования ASCII:

1) если следующая последовательность данных состоит по крайней мере из двух последовательных цифр, то кодируют следующие две цифры как сдвоенный разряд в схеме кодирования ASCII;

2) если проверка с предварительным просмотром [начинающимся на этапе, указанном в перечислении j)], указывает иную схему кодирования, то следует перейти на эту схему;

3) если указана схема кодирования Base 256, кодируют знак фиксации этой схемы кодирования со следующим за ним байтом длины, значение которого пока неизвестно. На этапе, указанном в перечислении g) или i), поле длины будет заполнено (может потребоваться второй дополнительный байт длины);

4) если следующий знак данных относится к расширенному набору знаков ASCII (со значением более 127), то его кодируют в схеме кодирования ASCII с использованием в качестве первого знака знака верхнего регистра (Upper Shift) (со значением 235);

5) в противном случае следующий знак кодируют по схеме кодирования ASCII;

c) при кодировании по схеме кодирования C40:

1) если схема кодирования C40 начинает обработку нового двойного знака символа и если проверка с предварительным просмотром [начинающаяся на этапе, приведенном в перечислении j)], указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак обрабатывают по схеме кодирования C40;

d) при кодировании по схеме кодирования Text:

1) если схема кодирования Text начинает обработку нового двойного знака символа и если проверка с предварительным просмотром [начинающаяся на этапе, приведенном в перечислении j)], указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак обрабатывают по схеме кодирования Text;

e) при кодировании по схеме кодирования X12:

1) если следующий знак не может быть закодирован по схеме кодирования X12, следует переключиться на схему кодирования ASCII и перейти к выполнению этапа, указанного в перечислении b);

2) если схема кодирования ANSI X12 начинает обработку нового двойного знака символа и если проверка с предварительным просмотром [начинающаяся на этапе, приведенном в перечислении j)] указывает иную схему кодирования, следует перейти на использование этой схемы;

3) в противном случае следующий знак обрабатывают по схеме кодирования X12;

f) при кодировании по схеме кодирования EDIFACT (EDF):

1) если следующий знак не может быть закодирован по схеме кодирования EDIFACT, следует переключиться на схему кодирования ASCII и перейти к выполнению этапа, указанного в перечислении b);

2) если схема кодирования EDIFACT начинает обработку нового тройного знака символа и если проверка с предварительным просмотром [начинающаяся на этапе, приведенном в перечислении j)] указывает иную схему кодирования, следует перейти на использование этой схемы;

3) в противном случае следующий знак обрабатывают по схеме кодирования EDIFACT;

g) при кодировании по схеме кодирования Base 256 (B256):

1) если проверка с предварительным просмотром [начинающаяся на этапе, приведенном в перечислении j)] указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак обрабатывают по схеме кодирования Base 256;

h) повторяют этап, указанный в перечислении b), до конца данных;

i) после просмотра данных, если действует схема кодирования Base 256, устанавливают поле длины в ноль (ноль указывает, что символ завершается схемой кодирования Base 256).

При проверке с предварительным просмотром [этапы, указанные в перечислениях j) — s)] изучают кодируемые данные для определения наилучшей схемы кодирования;

j) задают начальное значение счетчика знаков символа для каждой схемы:

1) если текущей является схема кодирования ASCII, устанавливают следующее:

значение ASCII равно 0;

значение C40 равно 1;

значение Text равно 1;

значение X12 равно 1;

значение EDF равно 1;

значение B256 равно 1,25;
в противном случае устанавливают:
значение ASCII равно 1;
значение C40 равно 2;
значение Text равно 2;
значение X12 равно 2;
значение EDF равно 2;
значение B256 равно 2,25.

2) если текущей является схема кодирования C40, то значение C40 равно 0;
3) если текущей является схема кодирования Text, то значение Text равно 0;
4) если текущей является схема кодирования X12, то значение X12 равно 0;
5) если текущей является схема кодирования EDIFACT, то значение EDF равно 0;
6) если текущей является схема кодирования Base 256, то значение B256 равно 0;

k) после просмотра данных:
1) округляют в большую сторону все значения счетчиков до целых чисел;
2) если значение счетчика ASCII менее или равно значениям остальных счетчиков, завершают проверку с указанием схемы кодирования ASCII;
3) если значение счетчика B256 менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования Base 256;
4) если значение счетчика EDF менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования EDIFACT;
5) если значение Text менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования Text;
6) если значение счетчика X12 менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования X12;
7) заканчивают проверку с указанием схемы кодирования C40;

l) обработка значений счетчика ASCII:
1) если знак данных — цифра, добавляют 1/2 к значению ASCII;
2) если знак данных — знак расширенного набора ASCII (знак со значением более 127), округляют значение ASCII в большую сторону до целого значения и добавляют к этому значению 2;
3) в противном случае округляют значение ASCII в большую сторону и добавляют к этому значению 1;

m) обработка значений счетчика C40:
1) если знак данных входит в основной набор знаков C40, добавляют 2/3 к значению C40;
2) если знак данных — знак расширенного набора ASCII (знак со значением более 127), добавляют 8/3 к значению C40;
3) в противном случае добавляют 4/3 к значению C40;

n) обработка значений счетчика Text:
1) если знак данных входит в основной набор знаков схемы кодирования Text, добавляют 2/3 к значению Text;
2) если знак данных — знак расширенного набора ASCII (знак со значением более 127), добавляют 8/3 к значению Text;
3) в противном случае добавляют 4/3 к значению Text;

o) обработка значений счетчика X12:
1) если знак данных — знак установленного набора X12, добавляют 2/3 к значению X12;
2) если знак данных — знак расширенного набора ASCII (знак со значением более 127), добавляют 13/3 к значению X12;
3) в противном случае добавляют 10/3 к значению X12;

p) обработка значений счетчика EDF:
1) если знак данных — знак установленного набора EDF, добавляют 3/4 к значению EDF;
2) если знак данных — знак расширенного набора ASCII (знак со значением более 127), добавляют 17/4 к значению EDF;
3) в противном случае добавляют 13/4 к значению EDF;

q) обработка значений счетчика B256:
1) если знак является управляющим знаком [знаком FNC 1, знаком структурированного соединения (Structured Append), знаком программирования устройства считывания (Reader Program) или знаком кодовой страницы (Code Page)], добавляют 4 к значению B256;
2) в противном случае добавляют 1 к значению B256;

r) если по крайней мере четыре знака данных B256 были обработаны в цикле проверки:
1) если значение ASCII, к которому добавляют 1, не более значений остальных счетчиков, завершают проверку с указанием схемы кодирования ASCII;

2) если значение B256, к которому добавляют 1, не более значения ASCII или менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования Base 256;
3) если значение EDF, к которому добавляют 1, менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования EDIFACT;
4) если значение Text, к которому добавляют 1, менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования Text;
5) если значение X12, к которому добавляют 1, менее значений остальных счетчиков, завершают проверку с указанием схемы кодирования X12;
6) если значение C40, к которому добавляют 1, менее значений ASCII, B256, EDF и Text, то:
   i) если значение C40 менее значения счетчика X12, завершают проверку с указанием значения C40;
   ii) если значение C40 равно значению X12, то:
      I) если один из трех знаков: знак-ограничитель и знаки-разделители X12 впервые встречаются в еще необработанных данных перед знаком, не представленным в X12, завершают проверку с указанием схемы кодирования X12;
      II) в противном случае проводят возврат к схеме кодирования C40;

s) повторяют этап, указанный в перечислении k), до встречи с условием возврата.

Приложение К
(рекомендуемое)

Возможность автоматического распознавания

Символы Data Matrix могут быть считаны с помощью надлежащим образом запрограммированных декодеров штрихового кода, которые предназначены для их автоматического распознавания среди символов иных символик. Набор символик, на распознавание которого запрограммирован декодер, должен быть ограничен той потребностью, которая определяется данным прикладным применением, что позволяет повысить надежность считывания.

Приложение L
(рекомендуемое)

Системный подход

Любое практическое применение символики Data Matrix следует рассматривать как законченное системное решение. Всеми компонентами, связанными с кодированием и декодированием символики (устройством нанесения символа на подложку или принтером, этикетками, устройствами считывания), образующими конкретную систему применения, необходимо управлять как целостной системой. Отказ в любом звене цепочки или несогласованность между ними могут поставить под угрозу эффективность функционирования всей системы.

Хотя соответствие спецификациям является единственным ключом к построению и нормальной работе системы, рекомендуется принимать во внимание иные соображения, оказывающие влияние на эффективность функционирования системы. Следующие рекомендации учитывают ряд факторов, которые необходимо иметь в виду при разработке или внедрении систем штрихового кодирования:

- следует выбирать плотность печати с допусками, которые могут быть обеспечены используемой технологией маркирования или печати;
- следует подбирать устройство считывания с разрешающей способностью, наиболее подходящей для плотности и качества печати символа, которая обеспечивает данная технология печати;
- должна быть обеспечена совместимость оптических свойств напечатанного символа с длиной волны источника излучения и измерительным элементом сканера;
- следует проверять соответствие символа на законченной этикетке или окончательной конфигурации упаковки. Покрытия, прозрачная упаковка, а также изогнутые и неровные поверхности могут оказывать влияние на считывание символа.

Технологии маркирования, которые не способны последовательно формировать сплошные линии непрерывных модулей, например, матричные или струйные принтеры, требуют проведения специальных мер, гарантирующих, что промежутки между номинально различимыми модулями не препятствуют декодированию символа при использовании размера апертуры, установленного в документе по применению. Кроме того, относительное позиционирование модулей, горизонтальных и вертикальных осей должно соответствовать требованиям к осевой неоднородности, приведенным в ISO/IEC 15415. Спецификации по применению должны также учитывать требования ISO/IEC 15415 относительно определения размера апертуры, освещения и прочих параметров.

Сканирующие системы должны учитывать колебания в диффузном отражении между темными и светлыми областями. При использовании ряда углов сканирования зеркальные составляющие отраженного излучения могут превышать требуемые составляющие диффузного компонента отражения, затрудняя эффективное считывание. В случаях, когда поверхности детали или материала могут быть изменены путем получения матовой поверхности или окончательной обработки, устраняющей влияние глянца, это может помочь минимизировать эффекты зеркального отражения. Если подобные мероприятия осуществить невозможно, необходимо принять меры по обеспечению специального освещения для считывания маркировки, чтобы оптимизировать контраст символа до приемлемого уровня.

Приложение М
(рекомендуемое)

Рекомендации пользователям

М.1 Общие положения

Символика Data Matrix предлагает гибкие способы кодирования данных. К альтернативным наборам знаков следует обращаться с использованием протокола интерпретаций ECI. Данные могут быть закодированы в символ квадратной или прямоугольной формы. Если длина сообщения превышает емкость символа, то оно может быть закодировано с использованием последовательности структурированного соединения нескольких (до 16) отдельных, но логически связанных символов штрихового кода Data Matrix (см. 7.2.4.9).

М.2 Выбор пользователем интерпретации в расширенном канале

Использование альтернативной интерпретации в расширенном канале (ECI) для задания определенной кодовой страницы (набора) или более специфичной интерпретации данных требует вызова дополнительных кодовых слов для активизации этой возможности. Использование протокола интерпретаций в расширенном канале (см. 7.3) обеспечивает возможность кодирования в данных знаков алфавитов, отличающихся от латинского (по ISO/IEC 8859-1), поддерживаемого интерпретацией по умолчанию (последовательность ECI 000003).

М.3 Выбор пользователем формы и размера символа

Символика штрихового кода Data Matrix предоставляет двадцать четыре квадратных и шесть прямоугольных конфигураций символа. Можно выбрать подходящий размер и форму символа, в зависимости от требований к его практическому применению. Технические требования к данным конфигурациям приведены в 7.4.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов межгосударственным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего межгосударственного стандарта
ISO/IEC 19762	MOD	ГОСТ 30721—2020 (ISO/IEC 19762:2016) «Информационные технологии. Технологии автоматической идентификации и сбора данных (АИСД). Гармонизированный словарь»
ISO/IEC 15415	—	*, 1)
ISO/IEC 646	NEQ	ГОСТ 27463—87 «Система обработки информации. 7-битные кодированные наборы символов»
ISO/IEC 8859-1	—	*
ISO/IEC 29158	—	*, 2)

* Соответствующий межгосударственный стандарт отсутствует. До его принятия рекомендуется использовать перевод на русский язык данного международного стандарта.

Примечание — В настоящей таблице использованы следующие условные обозначения степени соответствия стандарта:

- MOD — модифицированный стандарт;
- NEQ — неэквивалентный стандарт.

1) В Российской Федерации действует ГОСТ Р ИСО/МЭК 15415—2012 «Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация испытаний символов штрихового кода для оценки качества печати. Двумерные символы».

2) В Российской Федерации действует ГОСТ Р ИСО/МЭК 29158—2022 «Информационные технологии. Технологии автоматической идентификации и сбора данных. Рекомендации по контролю качества маркировки при прямом маркировании изделий (ПМИ)».

Библиография

[1] AIM Inc., ITS/04-001, International Technical Standard: Extended Channel Interpretations — Part 1: Identification Schemes and Protocol, aimglobal.org (Междунароный технический стандарт: Интепретации в расширенном канале. Часть 1. Идентификационные схемы и протокол)

[2] AIM Inc., Matrix Code Developer Software, aimglobal.org (Программа разработчика матричных кодов)

[3] GS1, «GS1 General Specifications» (Общие спецификации GS1)

[4] ISO/IEC 21471 Information technology — Automatic identification and data capture techniques — Extended rectangular data matrix (DMRE) bar code symbology specification [Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация символики штрихового кода с прямоугольным расширением матрицы данных (DMRE)]

[5] AIM Inc., International Symbology Standard — Data Matrix — AIM BC11 OBSOLETE, aimglobal.org

[6] ISO/IEC 8859-5 Information technology — 8-bit single-byte coded graphic character sets — Part 5: Latin/Cyrillic alphabet (Информационные технологии. 8-битовые однобайтовые наборы кодированных графических знаков. Часть 5. Латинский/кириллический алфавит)

[7] ISO/IEC 15424 Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers) [Информационные технологии. Технологии автоматической идентификации и сбора данных. Идентификаторы носителей данных (включая идентификаторы символик)][1)]

[1)] *Действует ГОСТ ISO/IEC 15424—2018 «Информационные технологии. Технологии автоматической идентификации и сбора данных. Идентификаторы носителей данных (включая идентификаторы символик)».*

УДК 003.295.8:004.223:006.354 МКС 35.040.50 01.080.50 IDT

Ключевые слова: технологии автоматической идентификации и сбора данных, кодирование, штриховой код, спецификация символики, Data Matrix, матричная символика

Редактор *Н.В. Таланова*
Технический редактор *В.Н. Прусакова*
Корректор *И.А. Королева*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 18.12.2025. Подписано в печать 03.02.2026. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 10,23. Уч.-изд. л. 9,26.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва. Нахимовский пр-т. д. 31. к. 2.
gost.e-ecolog.ru gost.e-ecolog.ru