Graficzne kody dwuwymiarowe po inżyniersku (3)

Kody kreskowe zrewolucjonizowały życie, a często też pracę ludzi na całym świecie. Łatwość naniesienia i odczytu sprawiły, że już w latach 80. ubiegłego wieku pojawiły się pierwsze elektroniczne systemy „inteligentne", a więc samodzielnie rozpoznające przedmioty. Wraz z rosnącą ilością danych, jakie użytkownicy chcieli zapisywać w kodach oraz z rozwojem technologii obrazowania komputerowego, pojawiły się kody dwuwymiarowe, znacznie zwiększające możliwości zautomatyzowanej identyfikacji. W niniejszym artykule opisujemy technikalia najpopularniejszych z nich, a w kolejnym numerze EP pokażemy mniej powszechne, specyficzne kody 2D.

Kody dwuwymiarowe są znacznie bardziej różnorodne niż jednowymiarowe. Powstały zarówno standardy bardzo proste, bazujące na bezpośrednim – bitowym zapisie liczb za pomocą kropek lub kwadratowych modułów, jak i skomplikowane, w których poszczególne bloki są poprzeplatane tak, by zminimalizować ryzyko utraty danych. Dwuwymiarowy zapis, w którym najczęściej minimalna szerokość pojedynczego modułu jest równa minimalnej wysokości sprawia, że ilość informacji, które można zmieścić w kodzie 2D jest znacznie większa niż w przypadku kodów 1D. W klasycznych kodach kreskowych, poza pewnymi wyjątkami, wysokość pasków nie niesie ze sobą żadnych informacji, a często wymagane jest, by nie była ona mniejsza niż kilkadziesiąt szerokości minimalnego modułu. Oznacza to, że w ogólności, na podobnej przestrzeni kod 2D jest w stanie zmieścić kilkadziesiąt razy więcej informacji niż kod jednowymiarowy.

Kody dwuwymiarowe, tak jak i jednowymiarowe, są w przeważającej mierze monochromatyczne, choć powstało nieco więcej kodów kolorowych. Wynika to z faktu, że skoro potrzebna jest duża gęstość upakowania informacji oraz i tak konieczne jest użycie czujnika obrazu i przetworzenie bardziej skomplikowanych informacji, można skorzystać z dużej wydajności nowoczesnych procesorów i przetwarzać także różnice w kolorach. Oprócz tego kody 2D znalazły zastosowanie również w przemyśle, w którym służą do oznaczania miniaturowych elementów, bez stosowania jakichkolwiek technik drukarskich. Zamiast tego piktogram z kodem jest wypalany lub wybijany na powierzchni obiektu, a nie nanoszony tuszem ani farbą. W praktyce dotyczy to np. właśnie elektroniki, gdzie małe kody DataMatrix lub Aztec Code używane są do znakowania PCB czy nawet podzespołów.

Tabela 15. Układ pasków numerujących rzędy w Code 16K. Liczby dodatnie reprezentują szerokości czarnych pasków, a ujemne to szerokości białych przerw.

Rząd	START		STOP	
	Wzór	Bitowo	Wzór	Bitowo
1	3 -2 1 -1 1	11100101	-3 2 -1 1	0001101
2	2 -2 2 -1 1	11001101	-2 2 -2 1	0011001
3	2 -1 2 -2 1	11011001	-2 1 -2 2	0010011
4	1 -4 1 -1 1	10000101	-1 4 -1 1	0111101
5	1 -1 3 -2 1	10111001	-1 1 -3 2	0100011
6	1 -2 3 -1 1	10011101	-1 2 -3 1	0110001
7	1 -1 1 -4 1	10100001	-1 1 -1 4	0101111
8	3 -1 1 -2 1	11101001	-3 1 -1 2	0001011
9	3 -2 1 -1 1	11100101	-1 1 -3 2	0100011
10	2 -2 2 -1 1	11001101	-1 2 -3 1	0110001
11	2 -1 2 -2 1	11011001	-1 1 -1 4	0101111
12	1 -4 1 -1 1	10000101	-3 1 -1 2	0001011
13	1 -1 3 -2 1	10111001	-3 2 -1 1	0001101
14	1 -2 3 -1 1	10011101	-2 2 -2 1	0011001
15	1 -1 1 -4 1	10100001	-2 1 -2 2	0010011
16	3 -1 1 -2 1	11101001	-1 4 -1 1	0111101

Rysunek 51. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Core 49

Fakt, że kody dwuwymiarowe mogą zmieścić dużo informacji na małej powierzchni został wykorzystany w wielu gałęziach przemysłu m.in.: w logistyce do kontroli przesyłu paczek, w transporcie lotniczym do przenoszenia informacji o pasażerach, na poczcie do potwierdzania wniesienia opłaty za przesyłkę i w produkcji, do rozpoznawania przedmiotów. Coraz powszechniej stosowane są też w aplikacjach internetowych, w których służą do wskazywania hyperlinków, przechowywania treści wizytówek, czy nawet jako unikalne identyfikatory użytkowników – np. na potrzeby logowania.

Znaczny wpływ na rozpowszechnienie kodów 2D ma rozwój smartfonów, z których praktycznie każdy jest wyposażony w kamerę. Pozwalają one na skanowanie różnorodnych kodów graficznych, a do tego ułatwiają korzystanie z aplikacji. Sama budowa oprogramowania i maszyn, wyposażonych w czytniki kodów 2D też jest znacznie prostsza niż kiedyś, choćby ze względu na powszechną dostępność sensorów obrazu, czy to w postaci pojedynczych komponentów, czy gotowych modułów. Znaczenie ma również powszechność narzędzi do kodów, a także (często darmowe) biblioteki do analizowania obrazów pod kątem zawartości piktogramów. Ręczne czytniki kodów, przeznaczone do pracy w punktach sprzedaży detalicznej, również coraz częściej są w stanie skanować obrazy dwuwymiarowe.

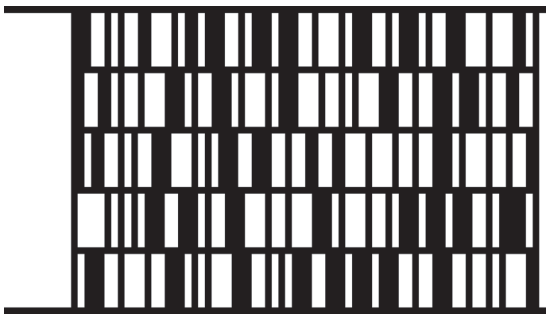
Poniżej opisane zostały najbardziej popularne kody dwuwymiarowe oraz kody powstałe jako rozwinięcie techniki jednowymiarowej. Łatwo zdobyć narzędzia do ich obsługi i w praktyce mogą stanowić pierwszy wybór podczas tworzenia nowej aplikacji z użyciem dwuwymiarowych kodów graficznych.

Code 49

Fakt, że odczytanie dwuwymiarowego kodu mozaikowego jest znacznie trudniejsze, niż odczyt kodu jednowymiarowego, to jedna z przyczyn, dla których pierwszym popularnym kodem dwuwymiarowym stał się Code 49, opracowany w 1987 roku przez firmę Intermec.

Tabela 16. Tryby Code 16K oraz ich znaczenie

Tryb (deklarowany na początku)	Zestaw znaków	Pierwszy, automatycznie dodawany znak
0	auto	
1	A	
2	B	
3	C	
4	B	Fnc1
5	C	Fnc1
6	C	Shift B
7	C	Podwójny Shift B



Rysunek 52. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Code 16K Mode 0

Z wyglądu prezentuje się jakby składał się z wielu rzędów zwykłych kodów kreskowych i w gruncie rzeczy jest to prawda. Jego konstrukcja opiera się o kod paskowy Code 39. Code 49 pozwala na zapisywanie dowolnych znaków ASCII, cechując się przy tym gęstszym upakowaniem informacji niż jednowymiarowe kody kreskowe, a więc lepiej nadaje się do oznaczania małych przedmiotów. Przy określonych minimalnych szerokościach najcieńszego paska (tzw. pojedynczego modułu), Code 49 mieści 93,3 znaki alfanumeryczne lub 154,3 cyfry w przeliczeniu na cal. Wynika to z faktu, że poszczególnym symbolom odpowiadają inne znaki, w zależności od trybu kodowania, a pomiędzy trybami kodowania można się przełączać korzystając z znaków „shift” i znaków funkcyjnych (podobnie jak w Code 128). Warto przy tym wspomnieć, że nazwa Code 49 pochodzi od 49-elementowej części tablicy ASCII, która służy do standardowego zapisywania większości ciągów alfanumerycznych. Sumy kontrolne również obliczane są modulo 49 i zapisywane na końcu każdego z rzędów, przed znakami stopu. Dodatkowo obliczane są też sumy kontrolne dla wszystkich rzędów – dwie lub trzy, zależnie od liczby wierszy.

Dopuszczalna liczba wierszy wynosi od 2 do 8. Wiersze są stałej długości 81 modułów, licząc razem z marginesami. Na każdy wiersz składa się 18 pasków czarnych i 17 białych. Pozwala to na zapisanie czterech dwuznakowych słów kodowych w rzędzie. Każde słowo kodowe składa się z 4 czarnych i 4 białych pasków, o łącznej szerokości 16 modułów. rzędy są też numerowane i oddzielone od siebie cienkimi czarnymi liniami. Wysokość kresek w rzędach nie jest z góry określona. Cechy te sprawiają, że w trybie numerycznym możliwe jest łącznie zapisanie 81 cyfr w jednym kodzie, a w zwykłym trybie alfanumerycznym dopuszczalne jest 49 znaków.

Code 16K

Bardzo podobny z wyglądu do Code 49 jest Code 16K, opracowany przez twórców kreskowego Code 128. Opiera się na Code 128, przy czym cechuje się większym upakowaniem informacji. Na powierzchni 2,4 cm^2 pozwala zmieścić 77 znaki ASCII lub 154 cyfry. Dopuszczalna liczba rzędów wynosi od 2 do 16. Kontrola poprawności odczytu opiera się o sprawdzanie parzystości dla każdego ze znaków, porównywanie sposobu prezentacji znaków startu i stopu dla poszczególnych rzędów (a więc w oparciu o numery rzędów) oraz sprawdzanie dwóch sum kontrolnych (modulo 107), umieszczanych na końcu kodu. W każdym rzędzie znajduje się 5

Tabela 17. Dodatkowe funkcje i znaki specjalne w Code 16K

Numer kodu	znak ASCII	Znaczenie w zestawie A	Znaczenie w zestawie B	Znaczenie w zestawie C
197	Å	DEL		
198	Æ	funkcja 3	funkcja 3	
199	Ç	funkcja 2	funkcja 2	
200	È	shift	shift	
201	É	zestaw C	zestaw C	
202	Ê	zestaw B	funkcja 4	zestaw B
203	Ë	funkcja 4	zestaw A	zestaw A
204	Ì	funkcja 1	funkcja 1	funkcja 1
205	Í	start A	start A	start A
206	Î	start B	start B	start B
207	Ï	start C	start C	start C
208	Ð	NUL		

Rysunek 53. Treść „0222578463” zakodowana w postaci kodu Code 16K Mode 2

znaków danych, które mogą być kodowane za pomocą trzech różnych trybów:

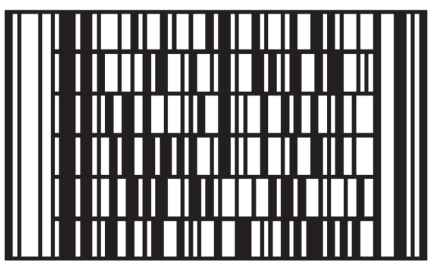
- A – w którym reprezentowane są znaki ASCII od numeru 0 do 95. Zawiera on znaki kontrolne ASCII, interpunkcyjne, cyfry, duże litery oraz 11 znaków specjalnych;
- B – w którym reprezentowane są znaki ASCII od numeru 32 do 127. Zawiera on duże i małe litery, cyfry, znaki interpunkcyjne oraz 11 znaków specjalnych;
- C – w którym dostępne są tylko cyfry i 7 znaków specjalnych.

Domyślnym trybem jest B, przy czym za pomocą odpowiedniego kodu startowego możliwe jest zmienienie trybu oraz ewentualnie automatyczne wstawienie znaku specjalnego/funkcyjnego. Oznaczenia numerów wierszy pokazano w tabeli 15, dopuszczalne tryby startowe i ich znaczenie zostały zebrane w tabeli 16, a wybrane znaki specjalne w tabeli 17.

Code 16K zdobył popularność w farmaceutyce, do oznaczania leków oraz w telekomunikacji, do znakowania urządzeń.

Codablock

Kolejnym kodem dwuwymiarowym, przypominającym zestawione ze sobą rzędy kodów paskowych jest Codablock. W istocie jest to kod dosłownie złożony z rzędów Code 39 (w przypadku wersji Codablock A) lub Code 128 (w wersjach Codablock F i Codablock 256). Wszystkie rzędy mają wspólne znaki startu i stopu, przy czym dodane są dodatkowe znaki informujące o numerach wierszy (tzw. wskaźniki wierszy) i znaki kontrolne wierszy. W przypadku powstałej na początku Codablock A, możliwe jest zestawienie od dwóch do 22 rzędów oddzielonych od siebie cienkimi czarnymi liniami. W każdym wierszu może się znaleźć od dwóch do 61 zakodowanych znaków, przy czym cały piktogram może pomieścić od 4 do 1340 znaków. Suma kontrolna liczona jest modulo 43.



Rysunek 54. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Codablock F

W znacznie bardziej rozpowszechnionej odmianie Codablock F możliwe jest użycie od 2 do 44 wierszy i od 4 do 62 zakodowanych znaków w każdym wierszu. W praktyce oznacza to możliwość zapisania od 16 do 2726 znaków alfanumerycznych w całym kodzie lub do 5452 cyfr, jeśli skorzysta się z zestawu znaków Code 128C, w którym zapisywane są po dwie cyfry jednocześnie.

Codablock 256 jest zmodyfikowaną wersją Codablock F, w którym dla każdego wiersza dodawane są dodatkowe znaki kontrolne.

Codablock został opracowany w taki sposób, by w razie potrzeby mógł być skanowany przez czytniki kodów jednowymiarowych. Wystarczy by każdy wiersz był zeskanowany oddzielnie czytnikiem Code 39 lub Code 128, w zależności od wersji Codablock. Dzięki temu, szczególnie wersja Codablock F, rozpowszechniła się w zastosowaniach medycznych, w logistyce i w magazynach, czyli wszędzie tam, gdzie stosowny był Code 128, ale istniała potrzeba oznaczania mniejszych elementów, na których długi Code 128 się nie mieścił.

PDF417

Bardzo ważnym kodem dwuwymiarowym jest PDF417, czyli Portable Document Format 417. Nazwa pochodzi od tego, że został on opracowany na potrzeby przenoszenia danych oraz, że dane zapisywane są w postaci słów, z których każde składa się z czterech czarnych i czterech białych pasków, które w sumie zajmują długość 17 modułów (a więc 17 szerokości najcieńszego paska). Pomimo zdecydowanie dwuwymiarowej budowy, kod ten jest niekiedy klasyfikowany jako zestaw kodów 1-wymiarowych; i rzeczywiście – da się go czytać odpowiednio zaawansowanymi skanerami liniowymi.

PDF417 składa się z minimum 3, a maksymalnie 90 rzędów. Rzędy mają wysokość co najmniej trzech modułów i mieszczą od 1 do 30 słów kodowych każdy. Dostępne jest 929 różnych słów kodowych, przy czym każde może być zapisane na trzy sposoby. Co trzeci rząd korzysta z tego samego sposobu zapisu słów kodowych, co pozwoliło zrezygnować z linii oddzielających rzędy – nie ma obaw, że kody w poszczególnych rzędach się ze sobą zleją i zostaną potraktowane jako jeden rząd. Cały kod ma wspólne znaki startu i stopu. Wewnątrz nich, każdy rząd zawiera swój lewy znak informacyjny, następnie umieszczane są słowa kodowe, po czym następuje prawy znak informacyjny. Wszystkie znaki (startu, stopu, informacyjne i słowa kodowe) zaczynają się od czarnego paska.

Rysunek 55. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci podłużnego kodu PDF417

Tabela 18. Liczba słów kodowych, które można odzyskać w kodzie PDF417, w zależności od zastosowanego poziomu bezpieczeństwa

Poziom	Liczba
0	0
1	2
2	6
3	14
4	30
5	62
6	126
7	154
8	510

Rysunek 56. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci wysokiego kodu PDF417

Rysunek 57. Treść „222578463” zakodowana w postaci kodu PDF417

Rysunek 58. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Micro PDF417

Rysunek 59. Treść „222578463” zakodowana w postaci kodu Micro PDF417

Rysunek 60. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu PDF417 Truncated

PDF417 może zapisywać informacje w trzech trybach: alfanumerycznym, binarnym i numerycznym, przy czym pierwszy z nich ma cztery odmiany:

- EXC Lower: domyślnie małe litery,
- EXC Alpha: domyślnie duże litery,
- EXC Mixed: domyślnie cyfry i znaki interpunkcyjne,
- EXC Punctuation: domyślnie znaki interpunkcyjne i nawiasy.

Spośród 929 dostępnych słów, 900 jest używanych do przechowywania danych, a 29 na potrzeby funkcji specjalnych. W trybie alfanumerycznym, w każdym słowie mieści się średnio niecałe 2 znaki. W trybie binarnym, każde 5 kolejnych słów pozwala na zapisanie dokładnie 6 bajtów danych. W trybie numerycznym tworzone są grupy o wielkości do 15 słów kodowych, mieszczących w sobie łącznie do 44 cyfr dziesiętnych. Oznacza to, że jeden cały kod pozwala na zapisanie do 2710 cyfr lub do 1850 znaków alfanumerycznych albo też do 1018 bajtów danych.

Tak wypełniony kod może jednak zajmować zbyt dużą powierzchnię, dlatego jego twórcy dopuścili możliwość decydowania o proporcjach kodu: liczbie słów kodowych w rzędzie (w każdym musi być tyle samo) i wynikającej z tego liczbie rzędów. W praktyce większą niezawodnością odczytu cechują się kody wyższe niż szersze. Tymczasem biorąc pod uwagę stosowaną minimalną wielkość modułu, PDF417 pozwala na zapisanie mniej więcej 680 bajtów lub 1150 znaków alfanumerycznych na cal kwadratowy.

PDF417 cechuje się bardzo dużymi możliwościami zabezpieczenia zapisanych danych przed błędnym odczytem. Standardowo stosowany jest mechanizm detekcji błędów, oparty o dwa słowa kodowe umieszczane na samym końcu. W efekcie cały kod PDF417 składa się z kolejno: symbolu opisu długości, słów kodowych z właściwymi danymi, uzupełniających słów kodowych, by cały kod miał formę prostokąta, opcjonalnych słów kodowych dla korekcji błędów i obowiązkowych dwóch słów kodowych dla detekcji błędów.

Liczba słów kodowych do korekcji błędów zależy od zastosowanego poziomu bezpieczeństwa. Istnieje 9 takich poziomów, od których zależy liczba uszkodzonych słów kodowych, które można odzyskać z kodu (tabela 18). Odzyskiwane są zarówno słowa kodowe nieczytelne, jak i błędnie zapisane. Stosowanie wysokiego poziomu zabezpieczeń ma jednak znaczenie tylko, jeśli w kodzie mieści się dużo danych – wraz z poziomem bezpieczeństwa znacząco rośnie rozmiar całego kodu.

PDF417, odkąd został opracowany w pierwszej połowie lat 90. znalazł zastosowanie w ogromnej liczbie aplikacji na całym świecie oraz został zestandaryzowany jako ISO15438. Jego stosowanie nie wiąże się z żadnymi ograniczeniami licencyjnymi i jest używany m.in. w logistyce, na lotniskach do znakowania kart pokładowych, do zapisu danych biometrycznych w dokumentach, a nawet do informowania o uiszczeniu opłaty za przesyłkę na listach w USA.

Popularność PDF417 sprawiła, że starano się go wykorzystać również tam, gdzie rozmiary klasycznej odmiany tego kodu były zbyt duże. Łatwo zauważyć, że dużą część powierzchni w kodzie zajmują znaki startu i stopu, dlatego zaproponowano format Truncated PDF417 (nazywany też Compact PDF), w którym z normalnego PDF417 ucięty jest znak stopu wraz z prawym słowem kodowym z każdego rzędu.

Innym, bardziej eleganckim sposobem na zmniejszenie rozmiarów PDF417 było wprowadzenie Micro PDF417, w którym ograniczono maksymalną długość kodowanych ciągów oraz zmniejszono minimalną wysokość rzędów z 3 do 2 modułów. W Micro PDF417 można ułożyć od 1 do 4 słów kodowych w rzędzie i od 4 do 44 rzędów, przy czym nie wszystkie dostępne kombinacje tych wartości są dopuszczalne. Łącznie można zapisać do 250 znaków alfanumerycznych, do 150 bajtów danych binarnych lub do 366 cyfr dziesiętnych. Mała wysokość rzędów może stanowić problem dla niektórych czytników liniowych, a brak możliwości zmiany stopnia korekcji błędów nie pozwala na zniwelowanie tego problemu.

W przypadku, gdy objętość standardowego PDF417 jest zbyt mała, można zastosować Macro PDF417, który pozwala na łączenie ze sobą bloków klasycznego PDF417 z użyciem odpowiednich komend funkcyjnych.

Ciekawą, ale mało popularną odmianą PDF417 jest również pochodzący z lat 90. SuperCode. Cechuje się on mniejszymi znakami startu i stopu, swobodnym rozmieszczeniem poszczególnych słów kodowych pogrupowanych w ściśle określone pakiety, pozwala na stosowanie 32 poziomów korekcji błędów oraz kodowanie znaków z alfabetów niełacińskich. Maksymalna pojemność SuperCode, przy najmniejszym poziomie korekcji błędów to 4083 znaki alfanumeryczne, 5102 cyfry lub 2546 bajtów.

DataMatrix

Prawdziwie dwuwymiarowym kodem, którego nie da się odczytać bez zastosowania czujnika obrazu, jest DataMatrix. Został opracowany w celu oznaczania bardzo różnych obiektów i aby spełnić wymagania zupełnie odmiennych aplikacji, przygotowano wiele odmian kodu. Podstawowy podział obejmuje sposób korekcji błędów i wielkość kodu, a co za tym idzie – pojemność. Sposób korekcji

błędów oznaczany jest z użyciem skrótu ECC XXX, gdzie XXX reprezentuje liczbę (000, 050, 050, 100, 140 lub 200). Im wyższa liczba, tym większa ilość danych służących redundantnych. Kody o oznaczeniach od ECC 000 do ECC 140 mają odmienny algorytm korekcji błędów niż ECC 200 i w związku z tym inaczej wyglądają. Aktualnie w praktyce stosuje się niemal wyłącznie odmianę ECC 200.

Tabela 19. Dostępne rozmiary kodów DataMatrix ECC 200 oraz ich pojemność a także zdolność korekcji błędów i obszar kodu zajmowany przez dane redundantne

Rozmiar	Pojemność			Maksymalna liczba błędów, które można poprawić		Procent kodu zajmowany przez korekcję błędów
	Cyfr	Znaków alfanumerycznych	Bajtów	Standardowe kodowanie korekcji	Kodowanie korekcji Erasure	
10×10	6	3	1	2	2	62,5%
12×12	10	6	3	3	3	58,3%
14×14	16	10	6	5	7	55,6%
16×16	24	16	10	6	9	50,0%
18×18	36	25	16	7	11	43,8%
20×20	44	31	20	9	15	45,0%
22×22	60	43	28	10	17	40,0%
24×24	72	52	34	12	21	40,0%
26×26	88	64	42	14	25	38,9%
32×32	124	91	60	18	33	36,7%
36×36	172	127	84	21	39	32,8%
40×40	228	169	112	24	45	29,6%
44×44	288	214	142	28	53	28,0%
48×48	348	259	172	34	65	28,1%
52×52	408	304	202	42	78	29,2%
64×64	560	418	278	56	106	28,6%
72×72	736	550	366	72	132	28,1%
80×80	912	682	454	96	180	29,6%
88×88	1152	862	574	112	212	28,0%
96×96	1392	1042	694	136	260	28,1%
104×104	1632	1222	814	168	318	29,2%
120×120	2100	1573	1048	204	390	28,0%
132×132	2608	1954	1302	248	472	27,6%
144×144	3116	2335	1556	310	590	28,5%
8×18	10	6	3	3	3	58,3%
8×32	20	13	8	5	5	52,4%
12×26	32	22	14	7	11	46,7%
12×36	44	31	20	9	15	45,0%
16×36	64	46	30	12	21	42,9%
16×48	98	72	47	14	25	36,4%

Rysunek 61. Treść „ELEKTRONIKA PRAKTYCZNA" zakodowana w postaci kodu DataMatrix

Kod DataMatrix ma zawsze kształt prostokąta, a w wielu przypadkach nawet kwadratu. Jego lewa i dolna linia mają grubość tzw. jednego modułu, czyli najmniejszego elementu składowego kodu i są całe czarne. Służą do wykrywania pozycji kodu, podczas gdy górna i prawa linia są przerywane i pozwalają czytnikowi określić wielkość pojedynczego modułu. Rozmiary kodów w wersji ECC 200 przyjmują zawsze wartości parzyste, co oznacza że prawy górny róg tych piktogramów jest biały. W przypadku ECC 100 – 140, rozmiary są nieparzyste (od 9×9 do 49×49 modułów), a więc prawy górny róg jest czarny. Dostępne rozmiary kodów DataMatrix ECC 200 zostały zebrane w tabeli 19. Kwadratowe kody o rozmiarach większych lub równych 32×32 i niektóre kody niekwadratowe, powstają z połączenia mniejszych kodów.

Treść kodu DataMatrix może być zapisana w różnych trybach. W przypadku wersji starszych niż ECC 200 dostępne tryby obejmują:

- zestaw cyfr i spację,
- zestaw dużych liter i spację,
- zestaw dużych liter, cyfr i spację,
- zestaw dużych liter, cyfr, spację oraz kropkę, przecinek, minus i slash,
- zestaw podstawowej tablicy ASCII,
- tryb zapisu binarnego.

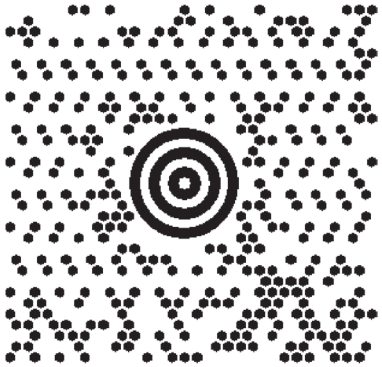
W ECC 200 dostępne są natomiast tryby:

- ASCII (128 znaków),
- C40 – cyfry i duże litery (trzy znaki zapisywane są na dwóch bajtach),
- tekstowy – cyfry i małe litery (trzy znaki zapisywane są na dwóch bajtach),
- X12 – standardowe znaki ANSI X12 (duże litery, cyfry, spacja i trzy znaki specjalne),
- EDIFACT – 63 znaki ASCII (od 32 do 94),
- binarny.

W zależności od wybranego trybu zmienia się pojemność kodu, zgodnie z tabelą 19, a więc możliwe jest zapisanie do 3116 cyfr dziesiętnych, do 2335 znaków alfanumerycznych, czy do 1556 bajtów. W poszczególnych trybach dostępne są różne znaki funkcyjne, które przełączają aktualny tryb lub pozwalają wybrać podzestaw znaków danego trybu.

Sposób tworzenia kodów DataMatrix jest bardzo skomplikowany, dzięki czemu udało się uzyskać jego dużą odporność na uszkodzenia. Kolejne znaki zapisywane są w postaci nachodzących na siebie, kwadratowych bloków, ułożonych skośnie. Fragmenty pierwszych znaków są ponadto umieszczone w okolicach prawego dolnego rogu kodu, w okolicach ewentualnej pustej przestrzeni i za danymi korekcji błędów.

Kody DataMatrix są wykorzystywane we wszelkiego rodzaju zastosowaniach. Bywają nadrukowywane na etykiety w logistyce, w lotnictwie, medycynie, a także – podobnie jak kody PDF417 – służą jako potwierdzenia wniesienia opłaty pocztowej w USA oraz w zmodyfikowanym formacie PostMatrix, w niemieckiej poczcie i Mailmark w poczcie w Wielkiej Brytanii. Świetnie sprawdzają się do znakowania drobnych przedmiotów i części. Nie muszą być drukowane – zamiast tego mogą być nanoszone laserem, wytrawiane chemicznie lub wybijane mechanicznie. Wielkość modułu zależy od parametrów czytnika i urządzenia tworzącego kod, dzięki czemu może on mieć rozmiary zarówno mikroskopijne, jak i wielocentymetrowe. DataMatrix używany jest w elektronice do znakowania układów elektronicznych i płytek drukowanych. Został też doceniony przez organizację GS-1, która przystosowała go do przechowywania danych z kodów EAN, tworząc kod GS1 Data Matrix. Data Matrix jest chętnie wykorzystywany w marketingu, do przenoszenia adresów internetowych – w tej formie nazywa się go kodem Semacode. DataMatrix jest wykorzystywany w wielu standardach ISO, a sam został opisany w ISO



Rysunek 62. Treść „ELEKTRONIKA PRAKTYCZNA" zakodowana w postaci kodu MaxiCode

16022. Ciekawostką jest fakt, że jeszcze niedawno pojawiały się roszczenia co do naruszenia patentów firm, których rozwiązania składają się na opis omawianego kodu.

MaxiCode

Bardzo ciekawe podejście do tematu kodów graficznych zaprezentowali inżynierowie z amerykańskiej firmy transportowej UPS, którzy opracowali na początku lat 90. kod MaxiCode. Jak można się domyślić, został on wymyślony na potrzeby logistyki, a podstawowe wymagania, jakie stawiano przed MaxiCode obejmowały możliwość sprawnego czytania szybko poruszających się kodów, naklejonych na paczki ułożone w dowolnej orientacji w przestrzeni. Kod musiał więc być łatwy do zlokalizowania, mieć mechanizm korekcji błędów, umożliwiać zapis znaków alfanumerycznych i w końcu być też nieduży, dzięki czemu mógłby zastąpić długie kody kreskowe.

Inżynierowie firmy UPS bardzo wzięli sobie do serca ten ostatni czynnik i określili dokładne wymiary kodu. Musi on mieć mniej więcej kwadratowy rozmiar o powierzchni 1 cala kwadratowego (dokładniej rzecz ujmując 25,5 mm×24,4 mm z dokładnością do 1,5 mm wzdłuż i wszerz). Aby ułatwić wykrywanie kodu, na jego środku znajdują się trzy koncentryczne okręgi określonej grubości, oddzielone od siebie białymi przerwami. Naokoło okręgów znajdują się 884 moduły reprezentujące poszczególne zapisane bity. Moduły mają kształt heksagonalny o wymiarach 1,02 mm w pionie i 0,88 mm w poziomie. Ze względu na 4 osie symetrii (przyjmując idealnie kwadratową budowę kodu), konieczne było też oznaczenie w jakiś sposób orientacji kodu w przestrzeni. W tym celu zastosowano sześć trójelementowych, równo rozmieszczonych od środka grup modułów, których wygląd jednoznacznie wskazuje, gdzie należy zaczynać czytać kod. Ponieważ żadne moduły nie mogą się ze sobą zlewać, ani zlewać z okręgami, w praktyce najczęściej drukuje się je w postaci kropek niemalże wpisanych w przestrzeń rzeczonych sześciokątnych modułów.

Dostępne są dwa stopnie korekcji błędów:

- standardowy (SEC), w którym na każde 42 bity właściwych danych dodawane jest 20 bitów korekcji błędów, co pozwala na odczyt kodów uszkodzonych maksymalnie w 16 %,
- rozszerzony (EEC), w którym na każdy bit danych dodawany jest jeden bit korekcji błędów, co w praktyce pozwala na poprawny odczyt kodów uszkodzonych maksymalnie w 25%.

W jednym kodzie można zapisać 93 znaki alfanumeryczne (za pomocą 1 słowa kodowego na każdy znak) lub 138 cyfr, przy czym ta druga wartość wynika z kodowania 9 cyfr w 6 słowach kodowych. Kod pozwala też na zapis dowolnych znaków ASCII, znaków rozszerzonych alfabetów łacińskich, a nawet liter arabskich, greckich, hebrajskich i cyrylicy, które jednakże zmniejszają dostępną użyteczną przestrzeń. W razie potrzeby przeniesienia większych ilości informacji, standard dopuszcza wydrukowanie obok siebie 8 kodów.

Wymienione liczby wynikają m.in. z faktu, że dwa moduły nie są w ogóle wykorzystywane, a 3 przeznaczone na informację o trybie kodu. Dostępne jest 7 trybów, przy czym dwa są nieużywane, dwa to standardowe formaty wykorzystywane przez UPS do znakowania przesyłek, dwa kolejne pozwalają na zapis dowolnych danych z wybranym stopniem korekcji błędów, a ostatni służy do programowania urządzeń odczytujących te kody.

W przypadku standardowych kodów UPS (tryby 2 i 3), kluczowe dane są zapisywane z wyższym stopniem korekcji błędów, a informacje dodatkowe (np. określające dopuszczalny sposób podnoszenia paczki) korzystają z korekcji EEC. Co więcej, kluczowe dane wraz z bitami korekcji umieszczane są bliżej centralnych okręgów, co ułatwia ich lokalizację. Mniej ważne dane znajdują się dalej od środka.

Właściwe słowa kodowe są reprezentowane przez grupy 6 modułów, ułożonych w trzech rzędach po dwa moduły. Moduł zaczerniony oznacza jedynkę, a biały – zero. Najbardziej znaczący bit słowa kodowego znajduje się w prawym górnym rogu grupy, a najmniej znaczący, w lewym dolnym. Inny układ mają tylko początkowe i końcowe słowa kodu.

MaxiCode jest obecnie dostępny do użytku publicznego i został opisany w standardzie ISO16023. Znany jest też pod nazwami: „UPS code", „Bird's Eye", „Target" i „Dense Code". Wadą MaxiCode jest wrażliwość na niską jakość druku, a więc na sytuacje gdy odstępy pomiędzy modułami nie są zachowane, lub gdy moduły zlewają się z okręgami. Dlatego – w tym. ze względu na określone fizyczne rozmiary kodu, zaleca się go drukować na drukarkach o rozdzielczości 300 dpi. Ponadto sam kod również powinien być generowany tak, by jego wymiary wyrażone w pikselach jak najbardziej odpowiadały liczbie punktów drukowanych przez daną drukarkę na docelowej powierzchni kodu (czyli aby wartość DPI drukarki odpowiadała wyliczonej wartości DPI kodu).

Aztec Code

Bardziej uniwersalnym, choć mocno podobnym do MaxiCode jest opracowany w połowie lat 90. Aztec Code. Tak jak w MaxiCode, tak i tu, elementem informującym o położeniu kodu jest umieszczony w środku obiekt, przy czym, zamiast koncentrycznych okręgów zastosowano koncentryczne kwadraty. To od ich wyglądu, przypominającego południowoamerykańskie piramidy widziane z góry, pochodzi nazwa kodu.

Poszczególne bity również zapisywane są czarnymi (jedynki) i białymi (zera) plamkami (modułami), ale kwadratowymi. Ich wymiar minimalny nie jest określony, dzięki

Rysunek 63. Treść „ELEKTRONIKA PRAKTYCZNA" zakodowana w postaci kodu Aztec Code

Tabela 20. Dopuszczalne rozmiary Aztec Code. Pierwsze cztery mogą być z realizowane w postaci formatu Compact (small). Pozo

Wymiary
15×15
19×19
23×23
27×27
31×31
35×35
39×39
43×43
47×47
51×51
55×55
59×59
63×63
67×67
71×71
75×75
79×79
83×83
87×87
91×91
95×95
99×99
103×103
107×107
111×111
115×115
119×119
123×123
127×127
131×131
135×135
139×139
143×143
147×147
151×151

czemu Aztec Code może mieć dowolną wielkość i być stosowany zarówno w postaci nadruków, jak i wytrawiany czy wypalany laserem w mikroskali.

Dostępne są dwie wersje kodu Aztec, różniące się liczbą kwadratów w środku symbolu oraz pojemnością.

- Kody kompaktowe mają jeden czarny kwadrat w środku, wielkości jednego modułu oraz dwa czarne kwadraty koncentryczne, których krawędzie mają grubość jednego modułu. Wszystkie te kwadraty są od siebie oddzielone przerwami (naturalnie też kwadratowymi) o szerokości jednego modułu. Łączne pole zajmowane przez te kwadraty to 9×9 modułów.
- Kody pełne mają o jeden czarny (a wiec i jeden biały w postaci przerwy) kwadrat więcej. Łączne pole zajmowane przez te kwadraty wynosi więc 13×13 modułów.

Na zewnątrz rogów zewnętrznego z kwadratów umieszczone są czarne moduły określające orientację. Lewy górny róg jest symbolizowany przez trzy takie moduły, prawy górny przez dwa, lewy dolny przez jeden, a prawy dolny przez ich brak, co zwiększa rozmiar środkowej części kodów do 11×11 modułów i 15×15 modułów, w zależności od formatu. Dodatkowo, w przypadku kodów w formacie pełnym, dodawane są jeszcze pionowe i poziome linie centrujące, składające się z naprzemiennych zer i jedynek. Dla kodów złożonych z maksymalnie 8 warstw linie centrujące biegną przez środek kodu. Dla większych wprowadzane są dodatkowe linie centrujące, w określonej odległości od środka, biegnące poziomo i pionowo przez cały kod.

Kody Aztec nie mają z góry określonego całkowitego rozmiaru – są tak jak większość kodów kreskowych, tym większe im więcej danych zawierają. Dane zapisane są pierścieniami umieszczanymi kolejno wokół kwadratów centrujących. Liczba pierścieni z danymi może być tylko parzysta, co oznacza, że dostępne rozmiary całkowite kodów to 15×15 modułów, 19×19 modułów, 23×23 moduły itd. Dodatkowo, aby nie marnować cennej przestrzeni, bezpośrednio naokoło największego z kwadratów centrujących umieszczane są informacje o całkowitej wielkości kodu. W przypadku kodów kompaktowych dostępnych w tym celu jest 28 bitów, a w przypadku formatu pełnego – 40 bitów. fakt, że wielkość kodu jest zapisana niemal w jego centrum sprawia, że skaner automatycznie wie, gdzie poszukiwać początku kodu i jest w stanie zignorować wszystkie elementy znajdujące się zaraz obok niego. Dzięki temu Aztec Code teoretycznie nie wymaga stosowania jakiegokolwiek marginesu!

Tabela 21. Wersje kodów QR i odpowiadające im rozmiary kodów, wyrażone w liczbie modułów wzdłuż i wszerz

Wersja	Rozmiar
1	21×21
2	25×25
3	29×29
4	33×33
5	37×37
6	41×41
7	45×45
8	49×49
9	53×53
10	57×57
11	61×61
12	65×65
13	69×69
14	73×73
15	77×77
16	81×81
17	85×85
18	89×89
19	93×93
20	97×97
21	101×101
22	105×105
23	109×109
24	113×113
25	117×117
26	121×121
27	125×125
28	129×129
29	133×133
30	137×137
31	141×141
32	145×145
33	149×149
34	153×153
35	157×157
36	161×161
37	165×165
38	169×169
39	173×173
40	177×177

Liczba pierścieni z danymi może wynosić od 2 do 8 dla formatu kompaktowego i od 2 do 64 dla pełnego. Oznacza to, że maksymalny rozmiar kodu kompaktowego będzie miał wymiary 27×27 modułów, a pełnego – 151×151 modułów (tabela 20). Naturalnie mniejsze porcje danych warto kodować w kodach typu compact, przy czym istnieje też możliwość zakodowania pojedynczego bajtu za pomocą małego kodu (tzw. runy – Aztec Rune) o rozmiarze 11×11 modułów. W przypadku największego rozmiaru da się zapisać 3832 cyfry lub 3067 znaków alfanumerycznych lub po prostu 1914 bajty danych, przy zalecanym poziomie korekcji błędów, wynoszącym 23%. Standard mówi, że poziom korekcji może wynosić od 5% do 95%, przy czym liczba ta wskazuje zawartość bitów korekcji błędów w całym symbolu, ale w praktyce są generatory przyjmujące poziom korekcji z zakresu od 1% do 99%. Można również dokonywać łączenia ze sobą symboli – do 26 kodów obok siebie.

Informacja o wielkości kodu, w przypadku formatu kompaktowego, jest zapisana na dwóch bitach, reprezentujących liczbę podwójnych warstw wokół zewnętrznego kwadratu centrującego. Liczba słów kodowych na 6 kolejnych bitach. W przypadku pełnych Aztec Code, liczba warstw jest określana przez 5 bitów, a liczba słów przez 11. Informacje te są zapisywane z korekcją błędów tak, by zajmować całe 28 lub 40 bitów dostępnych naokoło kwadratów centrujących.

Wielkość kodu jest podczas generowania obliczania automatycznie, w oparciu o dane zamienione na ciąg bitów oraz wskazanie odnośnie poziomu korekcji błędów. Do ciągu wyliczane są bity kontroli błędów, po czym całość zamieniana jest na odpowiednie słowa, z tym że w taki sposób, by nie zaczynały się od długich ciągów samych zer albo samych jedynek, a pomiędzy nimi umieszczane są pola odstępu celem zajęcia całej powierzchni kodu określonego rozmiaru. Następnie treść jest zapisywana spiralnie od zewnątrz, od lewego górnego rogu, odwrotnie do kierunku ruchu wskazówek zegara.

Słowa (bajty) zapisywane w Aztec Code są domyślnie traktowane jako znaki ASCII oraz znaki rozszerzonego alfabetu łacińskiego ISO 8859-1. Dodatkowo znaki funkcyjne pozwalają na wskazanie kodu aplikacji (tak jak w innych kodach zgodnych z GS1-128) lub są traktowane jako znaki ucieczki, umożliwiające wybór innego zestawu znaków – np. alfabetu arabskiego.

W związku z powyższym kody Aztec są używane w bardzo wielu aplikacjach. Bywają stosowane w logistyce, w liniach lotniczych, zamiast kodów PDF417, w kupowanych przez Internet biletach na różne wydarzenia i przejazdy, a także w przemyśle do znakowania elementów oraz w dokumentach. W Polsce można je znaleźć przykładowo w dowodach rejestracyjnych samochodów. Choć początkowo były chronione za pomocą patentu, szybko dozwolono ich swobodny użytek. Zostały też opisane w normie ISO 24778:2008.

QR Code

Zdecydowanie najbardziej popularnymi i rozpoznawanymi przez użytkowników dwuwymiarowymi kodami graficznymi są kody QR. Ich popularność nie wynika bynajmniej z niezrównanych parametrów, ale z uwarunkowań rynkowych. QR Code opracowała japońska firma Denso w 1994 roku, chcąc stworzyć standard szybkich w odczycie kodów graficznych – stąd też nazwa: Quick Response Code – w skrócie QR Code. Rodowód QR Code przyczynił się do jego sukcesu. Inżynierom z Denso zależało, by kod pozwalał standardowo przechowywać nie tylko bajty, cyfry i łacińskie znaki

Rysunek 64. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu QR Code Level L

Tabela 22. Liczba bitów użytych do określenia liczby kodowanych znaków w danym trybie, w zależności od trybu i wersji kodu QR

Tryb \ wersja	1...9	10...26	27...40
Cyfry	10	12	14
Znaki alfanumeryczne	9	11	13
Bajty (tryb binarny)	8	16	16
Kanji	8	10	12

Rysunek 65. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu QR Code Level H

Rysunek 66. Treść „ELEKTRONIKA” zakodowana w postaci kodu micro QR Code

alfanumeryczne, ale też japońskie kanji. QR Code miał wstępnie służyć do znakowania pojazdów podczas ich produkcji. Natomiast Japończycy, znani z miłości do nowoczesnych technologii, szybko podchwycili pomysł i zaczęli korzystać z kodów QR do przechowywania wszelkiego rodzaju informacji – począwszy od liczb, słów i zdań, a kończąc na wizytówkach i hyperlinkach. Z czasem popularność kodu rozszerzyła się na cały świat, przy czym w zbiorowej świadomości funkcjonują one jako kody z adresami internetowymi, podczas gdy w rzeczywistości mogą przechowywać znacznie więcej różnorodnych informacji.

Specyfikacja kodu QR jest zawarta w normie ISO 18004. Obejmuje ona dwa modele kodów – tzw. „model 1” i „model 2” oraz 40 wersji, które definiują wymiary całego symbolu. Dostępne są też warianty, które zostaną opisane nieco dalej. Standardowo używany jest obecnie QR Code model 2, w wersji dostosowanej do ilości danych, jakie mają być zapisane. Istotny jest też poziom korekcji błędów, przy czym dostępne są 4 opcje:

- L (poziom niski), w którym 7% słów kodowych można odtworzyć, as słowa korekcji zajmują około 20% kodu,
- M (poziom średni), w którym można odzyskać 15% słów kodowych, a słowa korekcji zajmują około 37% kodu,
- Q (poziom ćwiartki), w którym 25% słów można odzyskać, a słowa korekcji zajmują około 55% kodu,
- H (poziom wysoki), w którym da się odzyskać 30% słów kodowych, a słowa korekcji zajmują około 65% kodu.

Naturalnie im wyższy poziom korekcji błędów, tym mniej właściwych danych da się zmieścić w kodzie danej wersji (czyli danego rozmiaru). Do korekcji, tak jak w MaxiCode i Aztec, stosowane jest kodowanie Reeda-Solomona. Celem zapewnienia równomiernych możliwości odzysku nieczytelnych słów kodowych oraz redukcji złożoności algorytmu dekodującego, w przypadku dużych kodów QR są one dzielone na bloki i słowa kontrolne są obliczane dla nich niezależnie.

Charakterystycznymi elementami kodów QR są trzy duże (3×3 moduły) czarne kwadraty otoczone kolejno białą i czarną obwódką, umieszczone w rogach symbolu. Prawy dolny róg pozostaje pusty, co pozwala określić orientację kodu w przestrzeni. Fakt, że pozycjonowanie kodu opiera się o jego rogi sprawia, że konieczne jest zastosowanie marginesu. Ponadto naokoło czarnej obwódki kwadratów pozycjonujących również konieczne jest zachowanie wolnej przestrzeni o szerokości 1 modułu.

Dodatkowo pomiędzy dolnymi brzegami czarnych obwódek górnych kwadratów pozycjonujących biegnie przerywana, pozioma linia o grubości jednego modułu. Taka sama, tyle że pionowa linia biegnie pomiędzy prawymi brzegami czarnych obwódek lewych kwadratów

Wydanie specjalne „Raspberry Pi”

to polski przekład światowego bestsellera na temat słynnego minikomputera

To kompendium wiedzy o konfiguracji i sposobach programowania tego uniwersalnego urządzenia oraz prawie dwieście pomysłów i sztuczek aplikacyjnych

Nie będziesz rozczarowany!

Nie musisz być elektronikiem, aby zaprzęgnąć Raspberry Pi do wykonywania niezliczonych rodzajów funkcji i aplikacji

Z tym poradnikiem możesz to osiągnąć!

Szukaj na www.UlubionyKiosk.pl

(przesyłka GRATIS)

REKLAMA

Tabela 23. Wzory maskowania w kodach QR. Jeśli podana formuła, obliczana na podstawie koordynatów danego modułu daje w wyniku 0, kolor danego modułu musi zostać odwrócony; „a mod b" reprezentuje operację obliczania reszty z dzielenia a przez b; „floor(a) oznacza zaokrąglanie w dół do całości.

Numer wzoru maskowania	Formuła
0	(rząd + kolumna) mod 2 == 0
1	(rząd) mod 2 == 0
2	(kolumna) mod 3 == 0
3	(rząd + kolumna) mod 3 == 0
4	(floor(rząd / 2) + floor(kolumna / 3)) mod 2 == 0
5	((rząd * kolumna) mod 2) + ((rząd * kolumna) mod 3) == 0
6	(((rząd * kolumna) mod 2) + ((rząd * kolumna) mod 3)) mod 2 == 0
7	(((rząd + kolumna) mod 2) + ((rząd * kolumna) mod 3)) mod 2 == 0

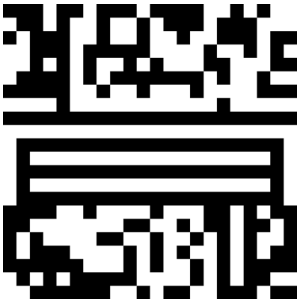
pozycjonujących. Służy ona ułatwieniu pozycjonowania oraz określenia wymiarów modułu. W kodach od wersji 2 wzwyż umieszczane są dodatkowe mniejsze kwadraty pozycjonujące o wielkości jednego modułu, otoczone białą i czarną obwódką, również o szerokościach 1 modułu.

Numer wersji kodu można łatwo odczytać poprzez pomiar odległości pomiędzy kwadratami pozycjonującymi (dostępne wersje zostały zebrane w tabeli 21), w czym pomagają wspomniane już przerywane linie pozycjonujące. Dodatkowo, dla wersji od 7 wzwyż konieczne jest zapisanie dwóch 18-bitowych (6 bitów danych właściwych i 12 bitów korekcji błędów) bloków, na lewo od prawego dużego kwadratu pozycjonującego i nad dolnym kwadratem pozycjonującym, w których przechowywane są informacje o wersji. Informacja o poziomie korekcji błędów jest zapisana na dwóch bitach i wraz z 3-bitową informacją o wzorze maskowania oraz z dodatkowymi danymi pozwalającymi sprawdzić poprawność tych informacji, umieszczane są w dwukrotnie w symbolu graficznym – obok lewego górnego kwadratu pozycjonującego oraz po części przy pozostałych dużych kwadratach pozycjonujących. Wzór maskowania służy takiej zmianie barw modułów, by uniknąć dużych jednolitych przestrzeni. Dostępne jest 8 wzorów maskowania.

Dzięki takim parametrom kod QR jest w stanie przechować największą ilość informacji spośród dotąd opisanych kodów 2-wymiarowych, a ponadto możliwe jest połączenie ze sobą do szesnastu kodów QR obok siebie. W pojedynczym kodzie modelu 2, wersji 40, przy poziomie korekcji błędów L można zapisać 2953 bajty w trybie binarnym, 4296 łacińskich znaków alfanumerycznych (gdzie każde dwa znaki zapisywane są na 11 bitach), 7089 cyfr (gdzie każde trzy cyfry zapisywane są na 10 bitach) lub 1817 znaków kanji/kana (1 znak wymaga 13 bitów).

Właściwe dane w postaci 8-bitowych słów kodowych są zapisywane od prawego dolnego rogu i układane w kolumny o szerokości 2 modułów. Pierwsza kolumna z prawej jest czytana od dołu do góry, druga od góry do dołu, po czym trzecia znów od dołu do góry itd. W zależności od wersji kodu, moduły mogą przyjmować inne kształty, by omijać kwadraty pozycjonujące, ale odczyt „wężykiem" jest zachowany. Ponadto pierwsze 4 bity kodu (a więc prawy dolny róg) zawiera informacje o trybie kodowania, a kolejny blok określa liczbę znaków zakodowanych w danym trybie, przy czym blok ten będzie miał różną długość, w zależności od trybu i wersji kodu (tabela 22).

Dostępne tryby określają, czy w kodzie przechowywane są dane binarne, znaki alfanumeryczne, cyfry czy kanji (to właśnie wyjątkowa sytuacja, w której alfabet niełaciński jest standardową opcją), czy też zawiera kontynuację danych z innego kodu QR, albo czy wśród kolejnych słów nie ma czasem znaków funkcyjnych, znanych z GS1-128. Warto dodać, że w trybie alfanumerycznym dostępne są tylko cyfry, duże litery i kilka dodatkowych znaków interpunkcyjnych (łącznie 45 znaków – tabela 24), co pozwala zwiększyć gęstość upakowania tych znaków oraz umożliwia zapisywanie adresów internetowych.



Rysunek 67. Treść „ELEKTRONIKA PRAKTYCZNA" zakodowana w postaci kodu Code One (Code 1)

Code One

Spośród mniej popularnych kodów warto w tej części artykułu wspomnieć o jednym z pierwszych, jakie pojawiły się na rynku, a zarazem pierwszym, który został udostępniony do użytku bez opłat licencyjnych. Code One, czasem oznaczany też jako „Code 1" został opracowany w 1991 roku. Składa się z czarnych i białych modułów oraz ze znaków pozycjonujących, przede wszystkim poziomych, na środku całego piktogramu. Code One ma z określone z góry osiem rozmiarów i jeden stały poziom korekcji błędów. Pozwala zapisywać zarówno znaki ASCII, dane binarne, jak i obsługuje znaki funkcji, znane już ze starszych, 1-wymiarowych kodów. Rozmiary kodu oznaczane są jako 1A, 1B, 1C… aż do 1H. Najmniejszy, 1A mieści do 13 znaków alfanumerycznych lub do 22 cyfr. Największy – 1H ma wymiary 134×148 kwadratowych modułów i pozwala zmieścić do 2218 znaków alfanumerycznych lub do 3550 cyfr. Code One może przyjmować różny kształt – przypominać wyglądem litery L, T lub U. Bywa stosowany w medycynie.

Podsumowanie

Omówione w tej części artykułu kody obejmują większość potencjalnych obszarów zastosowań i śmiało można stwierdzić, że poszukując standardu pozwalającego na umieszczanie informacji w postaci dwuwymiarowej grafiki, należałoby zacząć właśnie od zapoznania się z ich specyfiką. W kolejnym wydaniu EP opisane zostaną rzadziej spotykane kody dwuwymiarowe, ale które ze względu na swoje specyficzne cechy mogą okazać się wygodniejsze w użyciu w niektórych aplikacjach.

Marcin Karbowniczek, EP

Tabela 24. Kody odpowiadające poszczególnym znakom w QR Code w trybie alfanumerycznym. Kodowanie odbywa się poprzez zapisanie po dwa znaki na 11 bitach, przy czym zapisywana 11-bitowa wartość jest obliczana poprzez przemnożenie kodu pierwszego znaku przez 45 i dodanie do niego kodu drugiego zapisywanego znaku

Kod	Znak
00	0
01	1
02	2
03	3
04	4
05	5
06	6
07	7
08	8
09	9
10	A
11	B
12	C
13	D
14	E
15	F
16	G
17	H
18	I
19	J
20	K
21	L
22	M
23	N
24	O
25	P
26	Q
27	R
28	S
29	T
30	U
31	V
32	W
33	X
34	Y
35	Z
36	Spacja
37	$
38	%
39	*
40	+
41	–
42	.
43	/
44	: