

Kody kreskowe po inżyniersku (1)

W poprzednim numerze Elektroniki Praktycznej pokazaliśmy mnogość zastosowań kodów graficznych i w skrócie omówiliśmy ich zalety. Zaprezentowaliśmy typowe aplikacje i powszechnie stosowane rodzaje kodów 1D i 2D. Wskazaliśmy też, że decydując się na używanie kodów kreskowych, warto sięgnąć po te już opracowane, zamiast tworzyć własne rozwiązania. W niniejszym artykule dokonujemy przeglądu niemal wszystkich kodów kreskowych. Lekturę tę polecamy nie tylko projektantom planującym aplikację kodów, ale też wszystkim czytelnikom, jako zbiór innowacyjnych pomysłów inżynierskich, jakie zastosowano podczas projektowania kodów kreskowych.

Na przestrzeni lat opracowano kilkadziesiąt rodzajów monochromatycznych kodów kreskowych. Niektóre z nich służą do zapisywania tylko cyfr, niektóre do znaków alfanumerycznych, a jeszcze inne pozwalają na zapis dowolnej treści. Kody różnią się między sobą sposobem oznaczania zer i jedynek, dopuszczalną długością zapisanego ciągu znaków, gęstością upakowania znaków, odpornością na błędy odczytu oraz sformalizowanymi wymaganiami, dostosowanymi do specyfiki branż, w których są stosowane. I choć niektóre z kodów zostały zaprojektowane z myślą o bardzo konkretnych aplikacjach, niejednokrotnie zdarza się, że bywają też używane w zupełnie innych przypadkach, niż planowali autorzy. To czy tak się dzieje,

zależy przede wszystkim od dostępności narzędzi do kodowania i skanowania danych formatów, ale czasem też zapewne z nieświadomości inżynierów odnośnie dostępności innych standardów, bardziej adekwatnych do danego zastosowania. Niniejszy artykuł ma na celu zwiększyć świadomość polskich inżynierów w zakresie możliwości używania różnorodnych kodów kreskowych.

Najpopularniejsze kody

Chyba nie ma wątpliwości, że kodami kreskowymi, z jakimi większość ludzi spotyka się na co dzień są oznaczenia produktów sprzedawanych w sklepach. Są one drukowane w kilku formatach, których użycie zależy od regionu w którym towar został

wyprodukowany oraz od wielkości produktu. W USA, Kanadzie, Wielkiej Brytanii, Australii i Nowej Zelandii powszechnie stosowany jest kod standard UPC, a np. w Europie: EAN. Oba z tych kodów występują w wersjach krótszych i dłuższych, przy czym krótsze nadrukowuje się tam, gdzie dłuższy kod by się zwyczajnie nie zmieścił: na opakowaniach gum do żucia i innych małych przedmiotach.

Kody UPC

Kody w tym formacie mogą zawierać tylko cyfry w systemie dziesiętnym. Istnieją dwie powszechnie stosowane odmiany kodów UPC (Universal Product Code): 12-cyfrowy UPC-A i 6-cyfrowy UPC-E oraz kilka mających tylko specjalistyczne zastosowanie (UPC-B, UPC-C, UPC-D, UPC-2, UPC-5, które różnią się długością i obecnością cyfry kontrolnej). Kod UPC-E z założenia ma dać się przekształcić do kodu UPC-A, poprzez dodanie w jego środkowej części określonego ciągu zer i dodatkowych cyfr, zależnych od ostatniej z cyfr w kodzie UPC-E (zgodnie z tabelą 1). Kod UPC-A zawiera na swojej ostatniej pozycji cyfrę kontrolną, co oznacza, że może pomieścić 10^{11} (sto miliardów) wartości. Dla kodu UPC-E również jest obliczana cyfra kontrolna, ale nie jest ona zapisywana na jednej z 6 pozycji, a jedynie wpływa

na sposób prezentacji kolejnych cyfr w kodzie, o czym dalej. Ponieważ ostatnia cyfra w kodzie UPC-E informuje natomiast o sposobie konwersji na UPC-A, więc jest też użyteczna, oznaczałoby to, że w UPC-E można zapisać milion kombinacji, zachowując przy tym zdolność do wykrycia 100% błędów w postaci jednej niepoprawnie odczytanej cyfry. W praktyce liczba dostępnych kombinacji jest nawet większa i wynosi 64 miliony, co wynika z faktu, że sam sposób zakodowania poszczególnych cyfr w UPC-E też ma znaczenie. To jeden z najbardziej skomplikowanych systemów kodowania w kodach jednowymiarowych.

Sposób druku kodu UPC-A nie jest trywialny. Cyfry podzielone są na dwa bloki, po 6 cyfr każdy. Pomiędzy tymi blokami umieszczony jest znak środka, który nie reprezentuje żadnej liczby. Na krańcach kodu znajdują się identyczne znaki startu/stopu, które również nie reprezentują żadnej liczby, a jedynie, wraz ze znakiem środka, wskazują czytelnikowi początek, środek i koniec kodu, ułatwiając analizę obrazu. Aby umożliwić poprawny odczyt kodu, na zewnątrz znaków startu i stopu muszą znaleźć się odpowiednio szerokie strefy czyste (*quiet zones*), dzięki którym obrazy znajdujące się naokoło kodu nie będą traktowane jako jego fragmenty.

Same cyfry zakodowane są za pomocą 7 przylegających do siebie, cienkich czarnych lub białych pasków dla każdej z nich. Jak łatwo policzyć, 7 pasków, które mogą przyjąć tylko jeden z dwóch kolorów (biały lub czarny) skutkuje powstaniem 2^7 , czyli 128 kombinacji, które wykorzystywane są do zapisu tylko 10 różnych cyfr. Nie jest to bynajmniej przeoczenie, a celowy zabieg, zwiększający poprawność odczytu. Każda cyfra musi być zakodowana z użyciem 7 pasków, które przez to, że nie ma pomiędzy nimi odstępów, mogą się ze sobą zlewać, tworząc wizualnie paski o grubości równej wielokrotności pojedynczego paska. Co więcej wprowadzono też regułę, że wizualnie, każda cyfra może być reprezentowana jedynie za pomocą dwóch białych i dwóch czarnych pasków, co oznacza, że maksymalna grubość takiego połączonego paska (czy to czarnego, czy białego), może być równa czterokrotnej grubości paska pojedynczego. Wtedy to, z konieczności zmieszczenia się w bloku o szerokości 7 pojedynczych pasków, pozostałe trzy paski muszą mieć szerokość pojedynczą (każdy). W efekcie dostępne jest tylko osiem kombinacji, w których pojawia się biały lub czarny pasek o wizualnej grubości czterech pojedynczych pasków, osiem kombinacji, w których jest tylko jeden pasek pojedynczych rozmiarów i trzy paski podwójnych rozmiarów oraz 24 kombinacje z dwoma paskami o grubości pojedynczej, jednym podwójnym i jednym potrójnym. Razem daje to już tylko 40, a nie 128 dopuszczalnych kombinacji! Ale to nie

Tabela 1. Sposób przekształcania kodu UPC-E do UPC-A

Ostatnia cyfra UPC-E	Wzór UPC-E	Odpowiednik UPC-A
0	XXNNN0	0 lub 1 + XX000-00NNN + cyfra kontrolna
1	XXNNN1	0 lub 1 + XX100-00NNN + cyfra kontrolna
2	XXNNN2	0 lub 1 + XX200-00NNN + cyfra kontrolna
3	XXXNN3	0 lub 1 + XXX00-000NN + cyfra kontrolna
4	XXXXN4	0 lub 1 + XXXX0-0000N + cyfra kontrolna
5	XXXXX5	0 lub 1 + XXXXX-00005 + cyfra kontrolna
6	XXXXX6	0 lub 1 + XXXXX-00006 + cyfra kontrolna
7	XXXXX7	0 lub 1 + XXXXX-00007 + cyfra kontrolna
8	XXXXX8	0 lub 1 + XXXXX-00008 + cyfra kontrolna
9	XXXXX9	0 lub 1 + XXXXX-00009 + cyfra kontrolna

wszystko. Połowa z tych kombinacji zaczyna się od paska białego, kończąc na czarnym, a połowa od czarnego, a kończąc na białym (co wynika z faktu, że każda kombinacja składa się wizualnie z czterech pasków, a więc z parzystej ich liczby). Gdyby zestawili obok siebie dwie cyfry, z których jedna byłaby zakodowana z użyciem kombinacji zaczynającej się od paska białego, a druga od czarnego, to ponieważ pomiędzy kodami dla poszczególnych cyfr nie ma odstępów, ostatni pasek pierwszej cyfry byłby czarny i zlewałby się z pierwszym, czarnym paskiem drugiej cyfry. Znacznie zwiększyłoby to trudność odczytu, gdyż w najgorszym wypadku prowadziłoby to do zlania się ze sobą 8 pojedynczych pasków, których odróżnienie np. od 7 złączonych ze sobą pasków wymagałoby dużo większej precyzji niż odróżnienie złączonych ze sobą 3 i 4 pasków. Wszak pasek o grubości 8-krotności paska pojedynczego różni się od paska o grubości 7-krotności paska pojedynczego tylko o 12,5%, podczas gdy pasek pojedynczy o potrójnego o 25%, co pozwala dopuścić mniejszą precyzję pomiaru szerokości pasków. Oczywiście pod warunkiem, że skaner jest w ogóle w stanie wykryć pasek pojedynczej grubości, co jednak jest w praktyce zadaniem łatwiejszym niż rozróżnienie paska 7-krotnego od 8-krotnego.

W związku z powyższym, aby uniknąć tych problemów, inżynierowie tworzący kod UPC uznali, że skoro dostępna liczba kombinacji przy ustalonych wcześniej zasadach wynosi 40, a do zakodowania cyfr dziesiętnych wystarczy tylko 10, można swobodnie odrzucić połowę z nich, tych zaczynających się od paska czarnego, zwiększając tym samym pewność odczytu kodu. Jednakże zrezygnowanie z wykorzystania 10 z pozostałych 20 dopuszczalnych kombinacji byłoby zwykłym marnotrawstwem, w związku z tym podjęto jeszcze jedną decyzję, ułatwiającą wykrywanie błędów podczas skanowania: ustalono, że suma grubości (liczonych jako wielokrotności grubości paska pojedynczego) czarnych pasków musi być nieparzysta, co zredukowało liczbę dopuszczalnych kombinacji do 10. Teoretycznie tak zaprojektowany kod pozwala zabezpieczyć się przed błędnym rozpoznaniem kodu

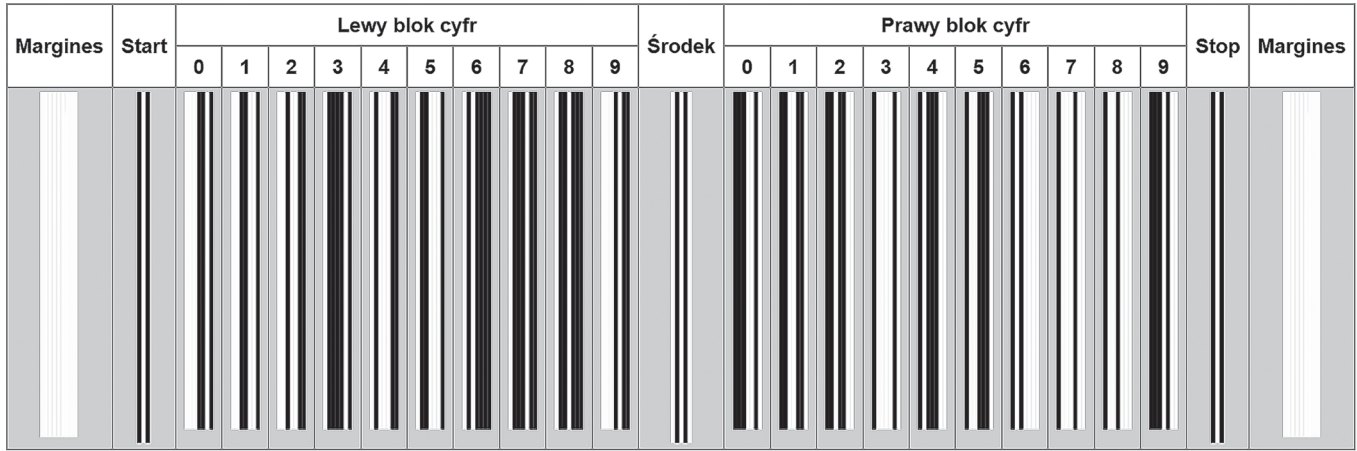


Rysunek 1. Treść „00222578463” zakodowana w postaci kodu UPC-A

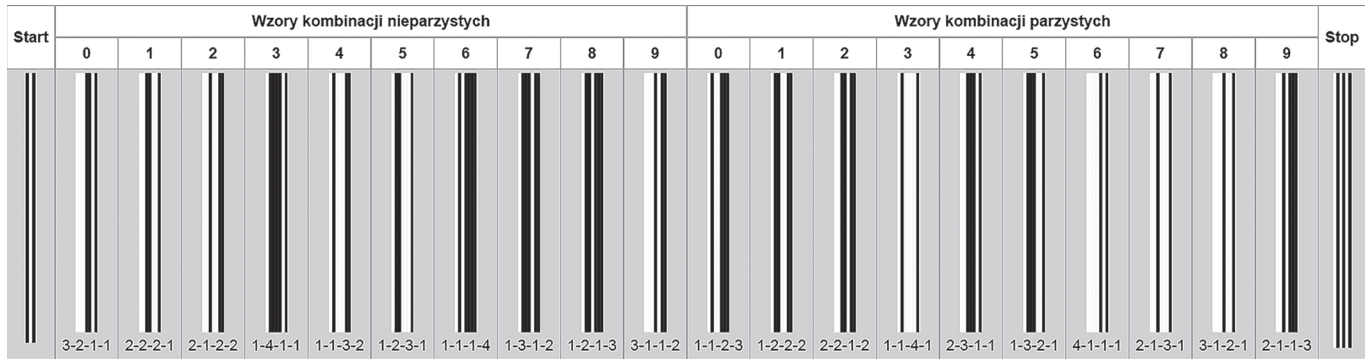
odczytywanego „do góry nogami”, gdyż wszystkie cyfry muszą się zaczynać od białego, a kończyć czarnym paskiem. Znano jednak, że to zbyt małe zabezpieczenie i zdecydowano, że cyfry znajdujące się po prawej stronie od znacznika środka będą musiały być kodowane w inny sposób niż te po lewej. Wizualnie będą zakodowane w sposób odwrotny niż ten dla cyfr z lewej strony paska. Oznacza to, że cyfry po prawej zawsze zaczynają się od paska czarnego, a kończą białym oraz że suma grubości czarnych pasków po prawej (liczona tak samo jak po lewej) stronie od środka będzie dla każdej z cyfr parzysta. W efekcie, wizualnie jedynka zapisana po prawej stronie będzie inwersją jedynki zapisanej po lewej stronie (**rysunek 1**). W ten sposób udało się też uniknąć problemu zlewania się ze sobą czarnych lub białych pasków z paskami startu, końca i środkowym.

Zestaw dopuszczalnych cyfr po obu stronach oraz format kodu UPC-A został przedstawiony na **rysunku 2**. Ponieważ znaki startu i końca mają szerokość 3 pojedynczych segmentów, znak środka ma szerokość 5 pojedynczych segmentów, puste przestrzenie po lewej i prawej stronie całego kodu muszą mieć szerokość równą przynajmniej 9-krotności pojedynczego segmentu, a każda z 12 cyfr ma wspomnianą już szerokość 7 segmentów, daje to łączną szerokość kodu $3 \cdot 2 + 5 + 12 \cdot 7 = 95$ szerokości najcieńszego paska, a razem z pustymi przestrzeniami kod musi zająć nie mniej niż 113 szerokości pojedynczego segmentu.

Zasady dotyczące kodowania UPC-E są bardzo podobne do UPC-A, ale dostępne 40 kombinacji rozmieszczenia pasków dla cyfr zostało użyte do zwiększenia liczby przechowywanych informacji. Kod UPC-E nie ma znaku środka, znak początku jest identyczny jak w przypadku UPC-A, a znak



Rysunek 2. Sposób kodowania cyfr w UPC-A



Rysunek 3. Sposób kodowania cyfr w UPC-E

Tabela 2. Dopuszczalne sposoby kodowania cyfr w UPC-E, w zależności od obliczonej cyfry kontrolnej dla odpowiednika UPC-A (E – parzysta suma segmentów czarnych pasków w cyfrze, 0 – nieparzysta suma segmentów czarnych pasków w cyfrze)

Cyfra kontrolna	Wzór kodowania UPC-A	
	dla UPC-A zaczynającego się od 0	dla UPC-A zaczynającego się od 1
0	EEEE00	000EEE
1	EE0E00	00E0EE
2	EE00E0	00EE0E
3	EE000E	00EEEE
4	E0EE00	0E00EE
5	E00E00	0EE00E
6	E000EE	0EEEE0
7	E0E0E0	0E0E0E
8	E0E00E	0E0E00
9	E00E0E	0EE0E0

końca ma szerokość sześciu segmentów i postać naprzemiennych pojedynczych białych i czarnych pasków. Cyfry nadal kodowane są za pomocą czterech naprzemiennych pasków, zajmujących za każdym razem 7 segmentów (40 kombinacji), ale z braku znaku środka, całkowicie wykluczono z użycia kody zaczynające się od czarnego paska – dzięki temu nie ma obaw, że dwie przylegające cyfry będą się ze sobą zlewać. Z pozostałych 20 kombinacji zamiast wykluczyć te, które mają parzystą sumaryczną grubość czarnych pasków przyjęto, że ich obecność

będzie niosła ze sobą dodatkową informację, a problem odczytu kodu „do góry nogami” będzie niwelowany dzięki kodowi końca, odmiennemu od kodu początku. Ta dodatkowa informacja obejmuje zarówno wartość cyfry kontrolnej, jak i informację o pierwszej cyfrze kodu UPC-A, do którego UPC-E można rozszerzyć. Aby pokazać jak to jest zrealizowane, trzeba najpierw zaprezentować sposób obliczania cyfry kontrolnej kodu UPC-A.

Cyfra kontrolna liczona jest poprzez zsumowanie nieparzystych cyfr kodu UPC-A, przemnożenie tej sumy przez 3 i dodanie do wyniku cyfr parzystych kodu (z pominięciem 12. cyfry, która jest przecież właśnie obliczana). Następnie bierze się resztę z dzielenia uzyskanej sumy przez 10 i jeśli jest ona różna od 0, odejmuje się ją od 10, co pozwala na uzyskanie cyfry kontrolnej. Cyfrę kontrolną dla kodu UPC-E oblicza się w identyczny sposób, ale dopiero po rozszerzeniu kodu do postaci UPC-A. Ponieważ zgodnie z tabelą 1., odpowiednik UPC-E w postaci UPC-A może zaczynać się od cyfry 0 lub 1, licząc cyfrę kontrolną dla UPC-E uzyskuje się dwa możliwe rozwiązania. Jeśli kod jest aktualnie tworzony, a nie skanowany, twórca wie, czy pierwszą cyfrą odpowiadającego UPC-E kodu UPC-A powinno być 0 czy 1 (wie, który kod chce zapisać), dzięki czemu wie, która z cyfr kontrolnych jest poprawna i na tej podstawie określa schemat zastosowania 20 wcześniej wspomnianych kombinacji pasków do utworzenia kodu

UPC-E. Jeśli kod jest odczytywany, a cyfra kontrolna sprawdzana, skaner rozważa obie obliczone wartości cyfry, a następnie sprawdza, która z nich pasuje do zastosowanego schematu kodowania. Dopuszczalne schematy zostały zebrane w tabeli 2 i odnoszą się do sposobu kodowania cyfr, przedstawionego na rysunku 3.

Szerokość kodu UPC-E również można przedstawić w postaci wielokrotności najmniejszego segmentu. 6 cyfr po 7 segmentów każda plus 3 segmenty startu i 6 segmentów stopu oraz brak przerw dają łącznie 51 segmentów szerokości, czyli o ponad połowę mniej niż kod UPC-A, który UPC-E może reprezentować (rysunek 4).

Zastosowanie kodów UPC obejmuje przede wszystkim znakowanie towarów przeznaczonych do sprzedaży detalicznej na rynkach w krajach anglojęzycznych, ale bywają one też wykorzystywane w farmaceutyce i w wydawnictwach. Oczywiście, aby zapewnić jednoznaczność, unikalną identyfikację produktów pochodzących od różnych wytwórców, konieczne było określenie



Rysunek 4. Treść „12345670” zakodowana w postaci kodu UPC-E

Tabela 3. Sposoby kodowania cyfr w EAN-13, w zależności od pierwszej cyfry kodu. Wygląd kodów L, G i R dla poszczególnych cyfr znajduje się w tabeli 4.

Pierwsza cyfra	Kody w grupie pierwszych 6 cyfr	Kody w grupie ostatnich 6 cyfr
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

zasad przypisywania numerów produktom oraz stworzenie organizacji, która by to nadzorowała. Aktualnie, kodami tymi zarządza międzynarodowa organizacja GS1, przy czym kody UPC są zastępowane kodami w formacie EAN, a jednocześnie wszystkie z nich, jeśli są wprowadzane do sprzedaży detalicznej, muszą pasować do standardu GTIN (Global Trade Item Numbers).

Kody EAN

W Europie i Azji również postanowiono opracować standard kodów kreskowych, bazując na osiągnięciach amerykańskich, czyli na kodzie UPC, ale tworząc nieco bardziej zaawansowane rozwiązanie. A ponieważ globalizacja pociągnęła za sobą konieczność ułatwienia sprzedaży importowanych towarów w dowolnym kraju, uznano, że projektowany kod będzie wstecznie kompatybilny z UPC. W ten sposób powstał kod EAN (dawniej European Article Number, a obecnie International Article Number), który jest aktualnie promowanym przez organizację GS1 i ma docelowo zastąpić kody UPC.

Podstawowym kodem EAN jest EAN-13, który zawiera 13 cyfr dziesiętnych, a więc o jedną więcej niż UPC-A. Każdy kod UPC-A będzie poprawnie odczytany przez czytnik kodów EAN-13 z tym, że dodatkowo, odczytana wartość zostanie poprzedzona cyfrą 0. Natomiast czytniki kodów UPC-A będą w stanie poprawnie odczytywać kody EAN-13, które zaczynają się od zera. Wynika to z faktu, że w kodach EAN-13, zastosowano niemal identyczny sposób kodowania cyfr, z tą różnicą, że pierwsza (czyli ta dodatkowa cyfra) nie jest bezpośrednio zapisywana tak jak pozostałe, ale wpływa na zmianę schematu używanych kombinacji pasków. W przypadku, gdy pierwszą cyfrą jest 0, schemat zapisu pozostałych cyfr jest identyczny, jak w kodzie UPC-A. Jeśli natomiast pierwsza cyfra jest inna, niektóre z cyfr w lewej połowie pozostałego kodu są zapisywane z użyciem kodów z parzystą sumą szerokości czarnych pasków – na analogicznej zasadzie, jak

Tabela 4. Sposób kodowania cyfr w EAN. Zera w kolumnach „Wzór” oznaczają białe paski, a jedyne paski czarne. Liczby w kolumnach „Szerokość paska” wskazują szerokości poszczególnych czarnych i białych pasków, podane jako wielokrotności pojedynczych segmentów. Warto zauważyć, że kody R są inwersją bitową kodów L, a kody G są kodami L, zapisanymi w odwrotnej kolejności. Te same zabiegi zastosowano w kodach UPC-A i UPC-E (rysunki 2. i 3.).

Cyfra	Wzór			Szerokość paska		
	Lewa		Prawa	Lewa		Prawa
	kod L (nieparzyste)	kod G (parzyste)	kod R (parzyste)	kod L (nieparzyste)	kod G (parzyste)	kod R (parzyste)
0	0001101	0100111	1110010	3211	1123	3211
1	0011001	0110011	1100110	2221	1222	2221
2	0010011	0011011	1101100	2122	2212	2122
3	0111101	0100001	1000010	1411	1141	1411
4	0100011	0011101	1011100	1132	2311	1132
5	0110001	0111001	1001110	1231	1321	1231
6	0101111	0000101	1010000	1114	4111	1114
7	0111011	0010001	1000100	1312	2131	1312
8	0110111	0001001	1001000	1213	3121	1213
9	0001011	0010111	1110100	3112	2113	3112

w UPC-E. Cyfry w prawej połowie kodu zawsze są zapisywane w ten sam sposób, identyczny jak w UPC-A. **Tabela 3** zawiera opis sposobu kodowania cyfr EAN-13, w zależności od wartości pierwszej cyfry (**rysunek 4**). Przykładowo, jeśli pierwszą cyfrą ma być 4, cyfry w lewej części kodu będą zapisywane z użyciem kolejno kodów LGLLGG (zgodnie z tabelą 3), a te w prawej części z użyciem kodów RRRRRR. Warto zauważyć podobieństwo do sposobu kodowania cyfr w UPC-E w zależności od wartości cyfry kontrolnej, w sytuacji gdy pierwsza cyfra odpowiadającego mu UPC-A to 0. Znaczenie kodów L, G i R zostało zademonstrowane w **tabeli 4**.

Twórcy systemu EAN również mieli świadomość ograniczeń dostępnej przestrzeni na niektórych przedmiotach wprowadzanych do sprzedaży i przygotowali skróconą wersję kodu w EAN-13 w postaci EAN-8 (**rysunek 5**). Kody EAN-8 nie są jednak kompatybilne w jakikolwiek sposób z UPC i nie rozwijają się do pełnych kodów EAN-13, choć budowane są na tej samej zasadzie, co zwykłe kody UPC, z tą różnicą, że zamiast po 6 cyfr po każdej stronie znaku środka, mają tylko po 4 cyfry (łącznie 8 cyfr).

Zarówno kody EAN-13, jak i EAN-8 zawierają cyfrę kontrolną. Konieczne jednak było zmodyfikowanie algorytmu jej obliczania, w stosunku do kodów UPC-A, tak by uwzględnił dodatkową cyfrę z EAN-13, a jednocześnie by obliczana wartość dla tej cyfry równej zeru była taka sama, jak w przypadku odpowiadającego kodowi EAN-13 kodu UPC-A. W EAN, podobnie jak w UPC, co druga cyfra jest mnożona przez trzy, wyniki mnożeń są sumowane a następnie dodawane są do tego pozostałe cyfry. Całkowitą sumę dzieli się przez dziesięć i bierze z tego resztę, a następnie, jeśli jest to wartość różna od 0, odejmuje ją od 10, uzyskując w ten sposób cyfrę kontrolną. Różnica polega na tym, które cyfry mnożone są przez 3, a które nie. Uniwersalna zasada



1 234567 890128 >

Rysunek 5. Treść „1234567890128” zakodowana w postaci kodu EAN-13

polega na tym, że przedostatnia cyfra kodu (a więc ostatnia, która nie jest cyfrą kontrolną) jest mnożona przez 3, a następnie począwszy od niej, liczona jest co druga, która będzie także mnożona przez 3. Oznacza to, że w przypadku kodu EAN-13, pierwsza cyfra nie jest mnożona przez 3, a druga, będąca pierwszą cyfrą ewentualnego odpowiednika w postaci UPC-A już tak. W przypadku kodu EAN-8, pierwsza cyfra jest mnożona przez 3. Ta sama zasada obowiązuje w przypadku odmiany EAN-18, określanej też mianami SSCC-18 (Serial Shipping Container Code) lub NVE (Nummer der Versandinheit), a stosowanej w logistyce.

Postać kodów GTIN

Choć z punktu widzenia elektronika nie ma znaczenia, co reprezentują wartości w skanowanych kodach EAN i UPC, warto wiedzieć, dlaczego nie należy swobodnie, w sobie znany sposób, umieszczać takich kodów na produktach (choć niektórzy producenci tak robią), które mogą znaleźć się w sprzedaży detalicznej. Kody EAN i UPC mogą być świetnym sposobem na zapisanie ciągu liczb, ale w praktyce reprezentują numery GTIN, przydzielane przez organizację GS1.

W przypadku numeru GTIN-13 zapisywanego w postaci EAN-13, pierwsze trzy cyfry określają narodową organizację, do której producent towaru wystąpił o przydzielenie puli kodów. Jeśli kod EAN-13 jest użyty do zapisania numeru ISBN (bywa wtedy nazywany mianem BookLand), pierwsze trzy cyfry będą miały wartość 978 lub 979 (dla



Rysunek 6. Treść „2578463” zakodowana w postaci kodu EAN-8



Rysunek 7. Treść „990222578463” zakodowana w postaci kodu EAN-99



Rysunek 8. Treść „257846” zakodowana w postaci kodu EAN-Velocity

nowszych publikacji), w przypadku numerów ISSN, pierwsze trzy cyfry również będą sprowadzały się do 979, a w dla czasopism, posiadających numer ISSN, prefiks to 977.

Drugi blok cyfr, o długości od trzech do ośmiu pozycji, to identyfikator producenta. Producenci wytwarzający szeroki asortyment dóbr wykupują (odpowiednio droższe) pule numerów, w których wspomniany blok ten jest krótki, podczas gdy mali wytwórcy rejestrują tańsze pule z długim blokiem.

Następne dwie do sześciu cyfr (w zależności od długości poprzedniego bloku) służą do oznaczania indywidualnych towarów. Ostatnia cyfra, to wspomniana wcześniej cyfra kontrolna, obliczana zgodnie z podanym algorytmem.

Organizacja GS1 zarezerwowała także pule numerów przeznaczonych do zastosowań wewnętrznych w firmach, a więc nie do wprowadzania do powszechnej dystrybucji. Kody te nie powinny być mylone z kodami produktów wprowadzonych do sprzedaży detalicznej w różnych sklepach. Jako



Rysunek 9. Treść „9771230352153:04” zakodowana w postaci kodu ISSN + 2



Rysunek 10. Treść „9788360233702:12345” zakodowana w postaci kodu ISBN + 5

ciekawostkę można wspomnieć, że kody EAN-13, zaczynające się od cyfr 99, zostały określone mianem EAN-99 (rysunek 7) i zarezerwowane przez GS1 na rzecz kuponów emitowanych i używanych tylko w ramach jednej jednostki – np. sklepu.

Kody GTIN EAN-13, wydawane w USA, ze względu na zgodność z UPC-A, zaczynają się od zera. Ponadto przygotowano też zestaw numerów GTIN do innych potrzeb – np. do oznaczania małych produktów z użyciem EAN-8. W przypadku tych ostatnich, kody zaczynające się od zera noszą miano EAN-Velocity (rysunek 8) i są przeznaczone do znakowania towarów na potrzeby sprzedaży detalicznej, w sytuacji gdy producent towaru nie oznakował go numerem GTIN. Kody EAN-Velocity mogą być używane tylko w ramach jednostki, która je wyemitowała.

Czasem można się spotkać z kodami określanymi mianem JAN, które są identyczne z EAN-13, ale nazwa JAN dotyczy kodów stosowanych w Japonii. Ich wyróżnienie wynika z faktu tamtejszej ogromnej popularności kodów kreskowych – Japonii przypisane są 2 różne kody krajów (49 i 45).

Warto też wspomnieć o kodach EAN 2 i EAN 5, które są używane jako rozszerzenie EAN-13, dodatkowo do oznaczania numerów wydania lub sugerowanej ceny detalicznej. Kod EAN-13 z kodem EAN 2 można znaleźć na okładce Elektroniki Praktycznej. W 5-cyfrowym kodzie EAN 5 pierwsza cyfra standardowo reprezentuje walutę (np. piątka – dolara amerykańskiego), a kolejne cztery cyfry to sugerowana cena wyrażona

w setnych częściach waluty (w przypadku dolara – w centach). Kody EAN 2 i EAN 5 zapisywane są z użyciem takich samych kombinacji pasków, jak w przypadku kodu EAN-13, z tym że dodatkowo stosowane są paski separatorów, a kontrola parzystości realizowana jest w postaci zmiany rodzaju używanych kombinacji pasków na poszczególnych pozycjach.

W końcu, należy też wspomnieć o standardowym sposobie drukowania kodów EAN-13, EAN-8, UPC-A i UPC-E. W każdym z tych przypadków, znaki startu i stopu oraz znak środka (jeśli występuje) są dłuższe (wystają w dół), niż pozostałe. Dodatkowo, tylko w UPC-A, znaki dla pierwszej i ostatniej cyfry również są dłuższe i równe długością znakom startu i stopu. W praktyce konieczne jest też drukowanie cyfr zawartych w kodzie, by w razie niemożności zautomatyzowanego zeskanowania kodu, można było go odczytać i wprowadzić ręcznie. Tu została przyjęta praktyka, która jednocześnie zabezpiecza kod przed wydrukowaniem go bez obowiązkowych marginesów. W przypadku 12-cyfrowego kodu UPC-A, skrajne cyfry znajdują się na zewnątrz kodu, rozszerzając tym samym cały piktogram. Pozostałe cyfry umieszczone są po 5, pomiędzy paskami pierwszej cyfry, środka i ostatniej cyfry. W kodach EAN-13, pierwsza cyfra jest na zewnątrz po lewej, pomiędzy paskami startu, środka i końca umieszczone są dwie grupy po 6 cyfr, a za znakiem końca umieszczony jest znak „>”, który wraz z pierwszą cyfrą, zapewnia zachowanie marginesu. W kodach UPC-E skrajne znaki umieszczone są na zewnątrz kodu, w celu zachowania marginesów, a w kodzie EAN-8 praktykuje się dodanie znaków „<” i „>” na zewnątrz kodu, podczas gdy pozostałe 8 cyfr jest umieszczonych w grupach po cztery, pod kodem.

Przykłady kodów ISSN+2 i ISBN+5 pokazano na rysunkach 9 i 10.

Code 39

Bardzo popularnym kodem jest Code 39, opracowany w latach siedemdziesiątych ubiegłego wieku przez pracowników firmy Intermec (rysunek 11, rysunek 12). Code 39 nazywany jest też mianem Code 3 of 9 oraz został ustandaryzowany jako ISO/IEC 16388. Nazwa kodu ma podwójne znaczenie. Po pierwsze, Code 39 w pierwotnej wersji pozwalał na zapisanie dowolnie długiego ciągu 39 różnych znaków: 26 liter, 10 cyfr i trzech znaków przestankowych: kropki, myślnika i spacji. Jednocześnie, każdemu z tych znaków odpowiadało 9 pasków (białych lub czarnych), z których trzy były grube, a pozostałe sześć – cienkie. Pasek rozpoznawany jest jako gruby, jeśli jest 2-3 razy grubszy niż pasek cienki (nazywany też pojedynczym segmentem). Dodatkowo zdefiniowano znak startu i końca, który



ELEKTRONIKA PRAKTYCZNA

Rysunek 11. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Code 39

Tabela 5. Sposób reprezentacji znaków w Code 39 wraz z odpowiadającymi znakom liczbami stosowanymi w Code 39 mod 43. Ważna jest zarówno szerokość białych przerw, jak i czarnych pasków

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



Rysunek 12. Treść „0222578463” zakodowana w postaci kodu Code 39

dzięki temu, że jest asymetryczny, pozwala na wykrycie, czy kod nie jest odczytany „do góry nogami”.

W Code 39 nie przewidziano specjalnej cyfry kontrolnej, ale zamiast tego paski dla poszczególnych znaków zaprojektowano tak, aby można było łatwo sprawdzić, czy w trakcie ich skanowania nie wkradł się błąd. Wszystkie znaki zawierają dwa grube czarne paski i jeden gruby biały oraz po trzy cienkie białe i czarne. Tak zdefiniowany kod (tabela 5) jest dosyć redundantny, a niewielka liczba dopuszczalnych znaków niweczyła sens stosowania Code 39 w wielu sytuacjach. Dlatego po pewnym czasie wprowadzono rozszerzenie o cztery dodatkowe znaki: plus, slash, dolar i procent, kodowane z użyciem trzech białych grubych pasków, jednego białego cienkiego i pięciu czarnych, cienkich. Ponieważ kod odpowiadający każdemu zapisywanemu znakowi jest niezależny od pozycji oraz w Code 39 nie ma cyfry kontrolnej, wprowadzenie drukowania tych kodów jest bardzo proste – wystarczy skorzystać z odpowiedniej czcionki, która dla poszczególnych znaków będzie miała przypisane odpowiednie kody graficzne. Pomiędzy znakami w Code 39 muszą znaleźć się pojedyncze przerwy, a znaki startu i stopu są standardowo przypisywane symbolowi gwiazdki (*).

Z czasem pojawiły się też trzy kolejne modyfikacje zmodyfikowanego wcześniej Code 39. Pierwsza z nich obejmowała wprowadzenie cyfry kontrolnej (Code 39 mod 43). By ją policzyć, konieczne jest przypisanie wszystkim 43 znakom rozszerzonego Code 39 wartości od 0 do 42, a następnie zsumowanie tych wartości odpowiadającym znakom w zapisanym kodzie i obliczeniu reszty z dzielenia takiej sumy przez 43. Reszta ta powinna zostać zapisana na koniec ciągu znaków, w postaci kodu odpowiadającemu znakowi o przypisanej wcześniej wartości równej obliczonej reszcie. Odczyt takiego formatu kodu, tj. sprawdzanie poprawności liczby kontrolnej wymaga zazwyczaj włączenia odpowiedniej opcji w skanerze kodów.

Znacznie ciekawszą modyfikacją jest Full ASCII Code 39 (nazywana też Code 39 Extended), w którym wprowadzono obsługę wszystkich 128 znaków ASCII (dla Code 39 Extended również można obliczać sumę kontrolną mod 43). Wielkie litery, cyfry oraz kropka, myślnik i spacja są zapisywane w sposób identyczny jak w Code 39, a pozostałe znaki ASCII składają się z dwóch sąsiadujących znaków Code 39, rozpoczynających się od procenta, slash lub plusa (tabela 6). Warto przy tym dodać, że choć znak startu i stopu pozostał bez zmian,



Rysunek 13. Treść „0222578463” zakodowana w postaci kodu Code 11



Rysunek 14. Treść „0-22-257-84-63” zakodowana w postaci kodu Code 11

wprowadzono dodatkowy znak gwiazdki, reprezentowany sąsiadującymi ze sobą symbolami slash i dużej litery J. Oczywiście oznacza to, że do zapisu małych liter oraz wcześniej niedostępnych znaków ASCII konieczne jest użycie łącznie 18 pasków (w tym 6 grubych), podzielonych na dwie grupy, oddzielone od siebie białą przerwą. W efekcie kod ten, jak i reszta cały Code 39, charakteryzuje się małą gęstością upakowania danych. Niemniej, ze względu na swoją popularność, jest wciąż bardzo chętnie wykorzystywany, szczególnie w starszych aplikacjach. Jeszcze inną odmianą jest Code 39 Reduced, w którym dostępne są tylko znaki alfanumeryczne, bez przestankowych, co skutkuje pulą 36 różnych znaków. Znak kontrolny w Code 39 Reduced oblicza się modulo 36.

Na podobnej zasadzie do Code 39, ta sama firma opracowała Code 11, przeznaczony do zapisu numerów telefonów (rysunek 13). W Code 11 dostępne jest 10 cyfr, znak myślnika oraz znak startu/stopu (rysunek 14). Każdy ze znaków zapisywany jest

Tabela 6. Sposób kodowania znaków w Full ASCII Code 39

Nr	Znak	Kod
0	NUL	%U
1	SOH	\$A
2	STX	\$B
3	ETX	\$C
4	EOT	\$D
5	ENQ	\$E
6	ACK	\$F
7	BEL	\$G
8	BS	\$H
9	HT	\$I
10	LF	\$J
11	VT	\$K
12	FF	\$L
13	CR	\$M
14	SO	\$N
15	SI	\$O
16	DLE	\$P
17	DC1	\$Q
18	DC2	\$R
19	DC3	\$S
20	DC4	\$T
21	NAK	\$U
22	SYN	\$V
23	ETB	\$W
24	CAN	\$X
25	EM	\$Y
26	SUB	\$Z
27	ESC	%A
28	FS	%B
29	GS	%C
30	RS	%D

Nr	Znak	Kod
31	US	%E
32	[spacja]	[spacja]
33	!	/A
34	"	/B
35	#	/C
36	\$	/D
37	%	/E
38	&	/F
39	'	/G
40	(/H
41)	/I
42	*	/J
43	+	/K
44	,	/L
45	-	-
46	.	.
47	/	/O
48	0	0
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	/Z
59	;	%F
60	<	%G
61	=	%H
62	>	%I
63	?	%J

Nr	Znak	Kod
64	@	%V
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
70	F	F
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O	O
80	P	P
81	Q	Q
82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W
88	X	X
89	Y	Y
90	Z	Z
91	[%K
92	\	%L
93]	%M
94	^	%N
95	_	%O
96	`	%W

Nr	Znak	Kod
97	a	+A
98	b	+B
99	c	+C
100	d	+D
101	e	+E
102	f	+F
103	g	+G
104	h	+H
105	i	+I
106	j	+J
107	k	+K
108	l	+L
109	m	+M
110	n	+N
111	o	+O
112	p	+P
113	q	+Q
114	r	+R
115	s	+S
116	t	+T
117	u	+U
118	v	+V
119	w	+W
120	x	+X
121	y	+Y
122	z	+Z
123	{	%P
124		%Q
125	}	%R
126	~	%S
127	DEL	%T, %X, %Y, %Z

za pomocą trzech pasków czarnych i dwóch białych. Niektóre znaki mają dwa szerokie czarne paski, a pozostałe tylko po jednym. Mogą też mieć szerokie białe paski. Wszystkie znaki są od siebie oddzielane pojedynczą białą przerwą. Standard przewiduje też możliwość dodawania jednej lub dwóch cyfr kontrolnych. Wykaz znaków dostępnych w Code 11 umieszczono w **tabeli 7**.

Code 93

Inżynierowie z Intermec zdawali sobie sprawę, że ich Code 39 trudno nazwać oszczędnym. Dlatego na początku lat 80. opracowali Code 93, który ponadto miał się cechować lepszym zabezpieczeniem odczytu. Code 93, w odróżnieniu od Code 39 nie składa się

z pasków o dwóch różnych szerokościach, ale tak jak kody UPC i EAN, z pasków o długościach równych maksymalnie czterokrotnej wielokrotności najcieńszego paska (pojedynczego segmentu). Każdy znak zawiera trzy paski czarne i trzy białe, których łączna szerokość wynosi 9 segmentów. Przy założeniu, że rozdzielczość czytnika takiego kodu jest taka sama, jak przy odczycie Code 39, oznacza to, że w Code 93 każdy ze znaków zajmuje w praktyce o kilka szerokości najcieńszego paska mniej. Ponadto w standardowym Code 93 uwzględniono 26 dużych liter, 10 cyfr i 7 znaków specjalnych: myślnik, kropkę, dolar, slash, plus, procent i spację. Przestrzeń dopuszczalnych kombinacji kodów, zgodna z opisanymi regułami i zaczynających się od czarnego

paska obejmuje 56 opcji, dlatego twórcy postanowili stworzyć kilka dodatkowych znaków, które ułatwią rozszerzenie Code 93 do obsługi pełnego zestawu znaków ASCII (Full ASCII Code 93 lub też Code 93 Extended), na podobnych zasadach, jak w przypadku Code 39 Extended. Lista znaków i odpowiadających im kodów w Code 93 została zebrana w **tabeli 8**. Znaki Full ASCII Code 93 można obliczyć korzystając z tabeli 7 i zamieniając prefiksy „\$”, „%”, „/” i „+” na „(\$)”, „(%)\”, „(/)” i „(+)\” (odpowiednio). Oznacza to zarazem, że same znaki dolara, procenta, slash i plusa, w Code 93 Extended nie wymagają prefiksów.

Warto dodać, że w typowym Code 93, przed znakiem stopu stosuje się dwa znaki kontrolne modulo 47. Powinno się też

Tabela 7. Znaki stosowane w Code 11

Kod	Znaczenie
	0
■	1
■	2
■	3
■	4
■ ■	5
■	6
■	7
■ ■	8
■	9
■	– (myślnik)
■	* (start/stop)

używać przynajmniej kilkumilimetrowych marginesów po bokach. Przykłady użycia Code 93 pokazano na rysunkach 15 i 16.

Code 128

Kolejnym po Code 39 i Code 93 standardem, umożliwiającym zapis znaków alfanumerycznych jest Code 128, który charakteryzuje się wyjątkowo dużą gęstością upakowania danych (rysunek 17). Jego jakość została doceniona w postaci standardu ISO/IEC 15417.

Code 128, podobnie jak Code 11, Code 39 i Code 93, obejmuje określony zestaw znaków. Są to 103 symbole danych, 3 symbole startu i dwa symbole stopu. Należy też pamiętać o pustych przestrzeniach naokoło kodu, o szerokości przynajmniej 10-krotności najcieńszego paska (pojedynczego segmentu). Poszczególne symbole składają się z trzech pasków białych i trzech czarnych. Znak stopu składa się z dwóch nachodzących na siebie symboli i ma cztery czarne naski. Znak stopu, czytany od tyłu, jest

Tabela 8. Znaki stosowane w Code 93

ID	Znak	Szerokości pasków	Zapis binarny wzoru	ID	Znak	Szerokości pasków	Zapis binarny wzoru
0	0	131112	100010100	28	S	211122	110101100
1	1	111213	101001000	29	T	211221	110100110
2	2	111312	101000100	30	U	221121	110010110
3	3	111411	101000010	31	V	222111	110011010
4	4	121113	100101000	32	W	112122	101101100
5	5	121212	100100100	33	X	112221	101100110
6	6	121311	100100010	34	Y	122121	100110110
7	7	111114	101010000	35	Z	123111	100111010
8	8	131211	100010010	36	-	121131	100101110
9	9	141111	100001010	37	.	311112	111010100
10	A	211113	110101000	38	SPACJA	311211	111010010
11	B	211212	110100100	39	\$	321111	111001010
12	C	211311	110100010	40	/	112131	101101110
13	D	221112	110010100	41	+	113121	101110110
14	E	221211	110010010	42	%	211131	110101110
15	F	231111	110001010	43	(\$)	121221	100100110
16	G	112113	101101000	44	(%)	312111	111010100
17	H	112212	101100100	45	(/)	311121	111010110
18	I	112311	101100010	46	(+)	122211	100110010
19	J	122112	100110100		Start/Stop *	111141	101011110
20	K	132111	100011010		(Odwrrotny stop)	114111	101111010
21	L	111123	101011000		Nie używane	411111	111101010
22	M	111222	101001100			111132	101011100
23	N	111321	101000110			111231	101001110
24	O	121122	100101100			113112	101110100
25	P	131121	100010110			113211	101110010
26	Q	212112	110110100			213111	110111010
27	R	212211	110110010			212121	110110110

rozpoznawany jako odwrrotny znak stopu, dzięki czemu skaner wie, że ma zamienić kolejność wczytanych danych. Pomimo, że Code 128 obejmuje tylko 103 symbole danych, jest w stanie zapisać wszystkie 128

znaków ASCII. W tym celu został podzielony na trzy zbiory:

- 128A, zawierający znaki ASCII od numeru 00 do 95, a więc cyfry, duże litery, znaki kontrolne i specjalne oraz funkcyjne (rysunek 18).
- 128B, zawierający znaki ASCII od numeru 32 do 127, a więc cyfry, duże i małe litery, znaki specjalne oraz funkcyjne.
- 128C, zawierający liczby od 00 do 99 oraz jeden znak funkcyjny (rysunek 19).

Co ważne, znaki funkcyjne pozwalają na przełączanie zestawów znaków w trakcie czytania kodu, dzięki czemu nie ma potrzeby stosowania podwójnych symboli dla nietypowych znaków ASCII. Tabela 9 zawiera listę dostępnych znaków w poszczególnych zbiorach kodu.

Każdy znak Code 128 (za wyjątkiem znaku stopu) zajmuje szerokość 11 segmentów,



ELEKTRONIKA PRAKTYCZNA

Rysunek 15. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Code 93



0222578463

Rysunek 16. Treść „0222578463” zakodowana w postaci kodu Code 93



0222578463

Rysunek 18. Treść „0222578463” zakodowana w postaci kodu Code 128 Mode A



ELEKTRONIKA PRAKTYCZNA

Rysunek 17. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu Code 128



0222578463

Rysunek 19. Treść „0222578463” zakodowana w postaci kodu Code 128 Mode C

Tabela 9. Lista znaków kodu Code 128 we wszystkich jego trybach

Wartość	Tryb			ASCII		Wzór kodu	
	128A	128B	128C	Numer	Znak	Zapis binarny	Szerokości
0	spacja	spacja	00	0032 lub 0212 / 252	Spacja lub Ó / ü	11011001100	212222
1	!	!	01	33	!	11001101100	222122
2	"	"	02	34	"	11001100110	222221
3	#	#	03	35	#	10010011000	121223
4	\$	\$	04	36	\$	10010001100	121322
5	%	%	05	37	%	10001001100	131222
6	&	&	06	38	&	10011001000	122213
7	'	'	07	39	'	10011000100	122312
8	((08	40	(10001100100	132212
9))	09	41)	11001001000	221213
10	*	*	10	42	*	11001000100	221312
11	+	+	11	43	+	11000100100	231212
12	,	,	12	44	,	10110011100	112232
13	-	-	13	45	-	10011011100	122132
14	.	.	14	46	.	10011001110	122231
15	/	/	15	47	/	10111001100	113222
16	0	0	16	48	0	10011101100	123122
17	1	1	17	49	1	10011100110	123221
18	2	2	18	50	2	11001110010	223211
19	3	3	19	51	3	11001011100	221132
20	4	4	20	52	4	11001001110	221231
21	5	5	21	53	5	11011100100	213212
22	6	6	22	54	6	11001110100	223112
23	7	7	23	55	7	11101101110	312131
24	8	8	24	56	8	11101001100	311222
25	9	9	25	57	9	11100101100	321122
26	:	:	26	58	:	11100100110	321221
27	;	;	27	59	;	11101100100	312212
28	<	<	28	60	<	11100110100	322112
29	=	=	29	61	=	11100110010	322211
30	>	>	30	62	>	11011011000	212123
31	?	?	31	63	?	11011000110	212321
32	@	@	32	64	@	11000110110	232121
33	A	A	33	65	A	10100011000	111323
34	B	B	34	66	B	10001011000	131123
35	C	C	35	67	C	10001000110	131321
36	D	D	36	68	D	10110001000	112313
37	E	E	37	69	E	10001101000	132113
38	F	F	38	70	F	10001100010	132311
39	G	G	39	71	G	11010001000	211313
40	H	H	40	72	H	11000101000	231113
41	I	I	41	73	I	11000100010	231311
42	J	J	42	74	J	10110111000	112133
43	K	K	43	75	K	10110001110	112331
44	L	L	44	76	L	10001101110	132131
45	M	M	45	77	M	10111011000	113123
46	N	N	46	78	N	10111000110	113321
47	O	O	47	79	O	10001110110	133121
48	P	P	48	80	P	11101110110	313121
49	Q	Q	49	81	Q	11010001110	211331
50	R	R	50	82	R	11000101110	231131
51	S	S	51	83	S	11011101000	213113
52	T	T	52	84	T	11011100010	213311
53	U	U	53	85	U	11011101110	213131
54	V	V	54	86	V	11101011000	311123
55	W	W	55	87	W	11101000110	311321
56	X	X	56	88	X	11100010110	331121
57	Y	Y	57	89	Y	11101101000	312113
58	Z	Z	58	90	Z	11101100010	312311

Tabela 9. c.d.

Wartość	Tryb			ASCII		Wzór kodu	
	128A	128B	128C	Numer	Znak	Zapis binarny	Szerokości
59	[[59	91	[11100011010	332111
60	\	\	60	92	\	11101111010	314111
61]]	61	93]	11001000010	221411
62	^	^	62	94	^	11110001010	431111
63	~	~	63	95	~	10100110000	111224
64	NUL	\	64	96	\	10100001100	111422
65	SOH	a	65	97	a	10010110000	121124
66	STX	b	66	98	b	10010000110	121421
67	ETX	c	67	99	c	10000101100	141122
68	EOT	d	68	100	d	10000100110	141221
69	ENQ	e	69	101	e	10110010000	112214
70	ACK	f	70	102	f	10110000100	112412
71	BEL	g	71	103	g	10011010000	122114
72	BS	h	72	104	h	10011000010	122411
73	HT	i	73	105	i	10000110100	142112
74	LF	j	74	106	j	10000110010	142211
75	VT	k	75	107	k	11000010010	241211
76	FF	l	76	108	l	11001010000	221114
77	CR	m	77	109	m	11110111010	413111
78	SO	n	78	110	n	11000010100	241112
79	SI	o	79	111	o	10001111010	134111
80	DLE	p	80	112	p	10100111100	111242
81	DC1	q	81	113	q	10010111100	121142
82	DC2	r	82	114	r	10010011110	121241
83	DC3	s	83	115	s	10111100100	114212
84	DC4	t	84	116	t	10011110100	124112
85	NAK	u	85	117	u	10011110010	124211
86	SYN	v	86	118	v	11110100100	411212
87	ETB	w	87	119	w	11110010100	421112
88	CAN	x	88	120	x	11110010010	421211
89	EM	y	89	121	y	11011011110	212141
90	SUB	z	90	122	z	11011110110	214121
91	ESC	{	91	123	{	11110110110	412121
92	FS		92	124		10101111000	111143
93	GS	}	93	125	}	10100011110	111341
94	RS	~	94	126	~	10001011110	131141
95	US	DEL	95	200 / 240	Ë / ð	10111101000	114113
96	FNC 3	FNC 3	96	201 / 241	É / ñ	10111100010	114311
97	FNC 2	FNC 2	97	202 / 242	Ê / ò	11110101000	411113
98	Shift B	Shift A	98	203 / 243	Ë / ó	11110100010	411311
99	Code C	Code C	99	204 / 244	Ì / ô	10111011110	113141
100	Code B	FNC 4	Code B	205 / 245	Í / õ	10111101110	114131
101	FNC 4	Code A	Code A	206 / 246	Î / ö	11101011110	311141
102	FNC 1	FNC 1	FNC 1	207 / 247	Ï / ÷	11110101110	411131
103		Start Code A		208 / 248	Ð / ø	11010000100	211412
104		Start Code B		209 / 249	Ñ / ù	11010010000	211214
105		Start Code C		210 / 250	Ò / ú	11010011100	211232
106		Stop (7 pasków/przerw)		211 / 251	Ó / û	1100011101011	2331112
—		Odwrotny Stop		—	—	11010111000	211133
		używany do wykrywania odczytu od prawej do lewej					

w których mieszczą się wspomniane 3 paski czarne i 3 białe. Maksymalna szerokość paska została ustalona na cztery segmenty (czterokrotność najcieńszego paska). Przyjęto też, że wszystkie znaki muszą się zaczynać od paska czarnego, a więc kończyć na białym, dzięki czemu nie ma potrzeby stosowania dodatkowych przerw pomiędzy nimi. Daje to 108 kombinacji, z czego jedna nie jest wykorzystywana, po to by nie była

mylona ze znakiem stopu, gdy ten jest czytany od tyłu. Znaki oznaczone jako „Shift A” i „Shift B” informują, że jeden następny znak w kodzie ma być traktowany jakby należał do Code 128A i Code 128B (odpowiednio). Znaki „Code A”, „Code B” i „Code C” informują, że cały następujący po nich ciąg ma być traktowany jako zapisany w odpowiednim, innym niż dotąd z trzech zbiorów Code 128. Jeszcze inne znaczenie mają kody

funkcyjne „FNC 1”, „FNC 2”, „FNC 3” i „FNC 4”. Pierwszy z nich służy do informowania czytelnika o znaczeniu zapisanej w kodzie treści, zgodnie z międzynarodowymi listami kodów. Definiują one format zapisu informacji, taki jak np. sposób podania kodu pocztowego czy numeru telefonu (określają, czy użyty jest przedrostek kraju itp.). Kod „FNC 4” służy natomiast do zapisywania rozszerzonych znaków standardu ISO-8859-1 (tzw.



(22)22578463

Rysunek 20. Treść „222578463” zakodowana w postaci kodu GS1-128

LATIN 1) i informuje, że do wartości kodu kolejnego znaku należy dodać 128 i zinterpretować go jako kod ze zestawu znaków LATIN 1. Dwa następujące po sobie znaki „FNC 4” informują, że wszystkie kolejne znaki mają być traktowane jako rozszerzone znaki standardu ISO-8859-1, a w przypadku gdy pojawią się w momencie gdy już są tak traktowane, stanowią komendę do powrotu do normalnego odczytu kodów.

Warto zwrócić szczególną uwagę na Code 128C, który jest bardzo przydatny do zapisu ciągów numerycznych. Pozwala na niemal dwukrotne skrócenie długości kodu, dzięki czemu możliwe jest generowanie i poprawne skanowanie istotnie dłuższych ciągów znaków niż np. w przypadku Code 93. Wystarczy użyć znaku startu „Start C” lub jeśli ciąg numeryczny zaczyna się dopiero w środku kodu, przełączyć odczytywany zbiór znakami „Code C”. Jednakże w części numerów nie da się zapisać w całości z użyciem jedynie Code 128C – tych, które zawierają nieparzystą liczbę znaków.

Code 128 obejmuje też obowiązkowy znak kontrolny. Liczony jest on jako reszta z dzielenia przez 103 sumy iloczynów numerów pozycji znaków w kodzie i wartości kodów tych znaków, powiększonej o wartość



(22)22578463

Rysunek 21. Treść „(22)2578463” zakodowana w postaci kodu GS1-128



ELEKTRONIKA PRAKTYCZNA

Rysunek 22. Treść „ELEKTRONIKA PRAKTYCZNA” zakodowana w postaci kodu GS1-128

kodu zastosowanego znaku startu. Oznacza to, że ponieważ wszystkie znaki mają wygląd niezależny od ich pozycji w ciągu, podobnie jak w Code 93 możliwe jest generowanie kodów Code 128 z użyciem odpowiedniej czcionki, o ile tylko następnie obliczona zostanie i dodana na koniec takiego ciągu, przed znakiem stopu, liczba kontrolna w postaci odpowiedniego znaku.

GS1-128

Specyficznym skodyfikowanym podzbiorem Code 128 jest GS1-128, znany też jako EAN/UCC-128. Wymaga on korzystania z znaku „FNC 1” do informowania o znaczeniu zakodowanego ciągu znaków, w oparciu o Identyfikatory Aplikacji GS1 (GS1 Application Identifiers). Identyfikatory te mają długość od 2 znaków wzwyż i mogą informować, że następujące po nich znaki przechowują np. datę ważności, numer kontenera, numer seryjny, datę pakowania,

gęstość, kwotę pieniężną, walutę, trasę, adres, kod kraju, wymiary, numer konta bankowego lub dane zapisane zgodnie z wewnętrznymi procedurami firmy, która kod stworzyła. Zwyczajowo, kody Identyfikatorów Aplikacji, są zapisywane w nawiasach, wraz z resztą kodu (już poza nawiasami), w postaci czytelnej dla człowieka, pod paskami. Fakt, że Code 128 cechuje się dobrym opakowaniem ciągów liczb sprawia, że kody GS1-128 bardzo dobrze nadają się do reprezentacji wartości numerycznych. Ciekawostką jest, że Identyfikator Aplikacji o wartości 01 oznacza, że następujące po nim 13 znaków to cyfry numeru GTIN, a więc powszechnie zapisywanego w kodach EAN-13, EAN-8 i UPC, a ostatnia, 14. (łącznie 16.) to cyfra kontrolna. Kod ten ma swoje osobne miano: GTIN-14 lub EAN-14.

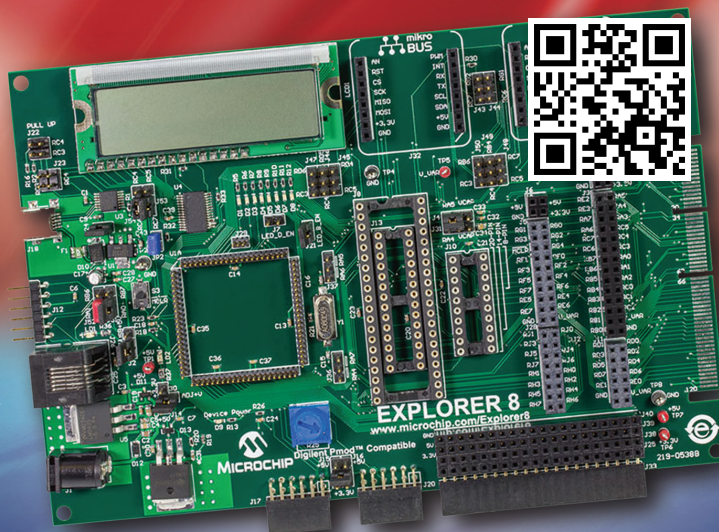
Przykłady użycia kodu GS1-128 pokazano na rysunkach 20...22.

Marcin Karbowniczek, EP

KONKURS Wygraj zestaw Microchip Explorer Development Kit!

Firma Microchip organizuje konkurs dla czytelników Elektroniki Praktycznej, w którym do wygrania jest zestaw deweloperski Microchip Explorer 8 Development Kit (model DM160228). Nagroda to pełnowartościowa płytki deweloperska dla projektów opartych o 8-bitowe mikrokontrolery PIC. Zestaw jest uniwersalnym rozwiązaniem, pozwalającym na podłączenie różnych zewnętrznych sensorów, interfejsów komunikacyjnych i HMI. Dostępna na płycie przestrzeń pozwala na swobodne rozszerzanie tworzonych projektów o dowolne podzespoły, sprawiając że jest to idealny zestaw dla inżynierów poszukujących narzędzia, które obsługiwałoby największą liczbę 8-bitowych mikrokontrolerów PIC. Płytki Explorer 8 to najnowszy produkt z szerokiej gamy profesjonalnych zestawów deweloperskich dla 8-bitowych mikrokontrolerów PIC. Powstała w oparciu o popularną płytkę PIC18 Explorer Board, przy czym różni się od niej o nowe funkcje:

- obsługuje wbudowane w mikrokontrolery, ale niezależne peryferia
- jest kompatybilna z wszystkimi mikrokontrolerami PIC



- pozwala projektować z użyciem układów z 8, 14, 20, 28, 40/44, 64 i 80 wyprowadzeniami.

Firma Microchip zaprojektowała zestaw Explorer 8 Development Kit, chcąc uczynić z niego podstawowe narzędzie deweloperskie dla 8-bitowych projektów. Wbudowane komponenty ułatwiają projektowanie interfejsów użytkownika, tworzenie przetwornic mocy, budowę aplikacji wpisujących się w nurt Internetu Przedmiotów, czy projektowanie ładowarek akumulatorów oraz wykorzystywanie 8-bitowych PICów w dowolnych innych aplikacjach. Ponadto zestaw można rozbudowywać za pomocą dwóch interfejsów Digilent Pmod, dwóch gniazd MikroElektronika Click i dwóch dodatkowych wyprowadzeń.

Explorer 8 Dev. kit obsługuje też różne narzędzia programistyczne, takie jak PICkit 3, ICD 3 i MPLAB REAL ICE In-Circuit Emulator. Aby wziąć udział w konkursie wystarczy się zarejestrować na stronie organizatora, pod adresem: <http://goo.gl/Uo6llW>