

## A vonalkódokról

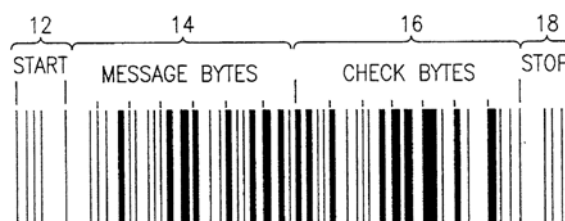
A vonalkód az adatoknak olyan grafikus elrendezése, melyet optikai leolvasóval (vonalkód olvasóval) egyszerűen vissza lehet fejteni. Ezeket általában áruk csomagolására nyomtatják, belekódolva bizonyos információkat a termékekről (pl. származási ország, a termék ára stb.), de napjainkban ennél sokkal elterjedtebbek más területeken is.

### Történeti áttekintés

A vonalkódokat a történelem során először a vagonok címkézésére használták, de igazi sikert csak a bevásárlóközpontok pénztárainak automatizálásánál értek el velük. 1948-ban Bernard Silver, a Drexel egyetem végzős hallgatója meghallotta amint egy helyi üzletlanc elnöke megkérte az egyik dékánt, hogy fejlesszen ki egy olyan rendszert, mely automata módon kiolvassa a termékinformációt a pénztárnál. Silver elmondta a hallottakat barátjának, Norman Joseph Woodland-nek, és együtt elkezdtek dolgozni ezen, több rendszert is kifejlesztve. Az első ilyen rendszerük ultraibolya tintát használt, de mivel az ilyen tinta túl drága volt, nem került használatba. Woodland-nek meggyőződése volt, hogy a rendszer továbbfejlesztett változata használható lesz, ezért folytatta munkáját. A következő ötlete a Morse-kódból származott és az első vonalkódot a tengerparton a homokba írta. A vonalkód leolvasására egy 500 wattos villanykörtét használt, mellyel átvilágította a papírt egy RCA935 fotódetektorral. Később szabadalmaztatta találmányát, melyet 1952-ben elfogadtak. Woodland munkáját az IBM-nél folytatta, ahol tovább próbálta fejleszteni a találmányát.

### Vonalkódok szimbolizmusa

Az üzenetek és vonalkódok közötti leképzést szimbolizmusnak nevezzük. Egy szimbolizmus specifikációja magába foglalja az üzenetnek a karakterenkénti/számjegyenkénti kódolását, valamint a start és stop karakterek kódolását vonalakba és szünetekbe, a „halk zóna” méretét a vonalkód előtt és után, valamint az ellenőrzőösszeg kiszámítását. A vonalkód felépítését láthatjuk az alábbi ábrán.



1. ábra  
*A vonalkód felépítése*

### Vonalkódok osztályozása

Két típusú vonalkódot különböztetünk meg:

- egydimenziós (lineáris) vonalkód,
- kétdimenziós (halmazott) vonalkód.

Az egydimenziós vonalkódok sötét és világos vonalakkból állnak. Vonalkód típustól függ, hogy mi határozza meg a kódolt értéket. Vannak olyan típusú egydimenziós vonalkódok, ahol a fehér vonalak vastagságát figyelmen kívül hagyják, máshol annak is van jelentősége, része a kódnak. Példát egydimenziós vonalkódra a 2. ábrán láthatunk. Könnyen belátható, hogy az egydimenziós vonalkódok csak a vízszintes tengely mentén hordoznak információt. Mégsem hanyagolható el a vonalkód magassága, mivel ha koszos a vonalkód, leolvasáskor pontatlan eredményt kaphatunk. Ezért ajánlott, hogy a vonalkód megfelelő magasságú legyen, hogy ki lehessen küszöbölni a leolvasási hibákat. Leolvasásukat általában lézerleolvasókkal végzik.

Az egydimenziós vonalkódokkal ellentétben a kétdimenziós vonalkódok a függőleges és vízszintes tengely mentén egyaránt tárolnak információt. Így az ilyen kódokban sokkal kisebb területen el lehet tárolni hibajavítási információkat is. Kezdetben a kétdimenziós vonalkódokat ott használták, ahol kis helyre kellett beszorítani az információt vonalkóddal, pl. gyógyszerdobozokra, fiolákra. Napjainkban már nemcsak ilyen helyeken használják, hanem elterjedtek a postai levelezésben és egyéb olyan helyeken, ahol több információt kell tárolni, pl. egy borítékon nevet, címet, postai irányítószámot kódolni, állami hivatalokhoz leadandó papírokon szintén több adatot kódolnak bele.

### Vonalkódok típusai

#### Egydimenziós vonalkódok

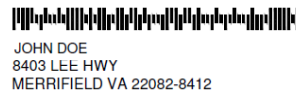
UPC  
(Universal Product Code)



EAN-13  
(European Article Number)



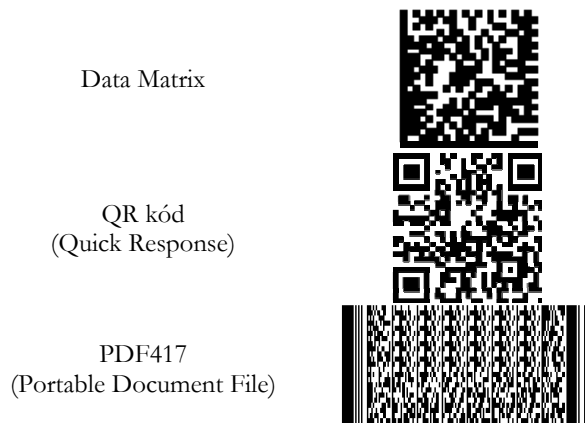
Intelligent Mail Barcode



#### Kétdimenziós vonalkódok

Azték kód





2. ábra

### Az Universal Product Code

Az UPC kódot elsősorban az Amerikai Egyesült Államokban és Kanadában használják az áruk címkézésére. 12 számjegy kódolását tartalmazza a következő struktúrával:  $S\underline{L}LLLLLMRRRRR\underline{R}E$ , ahol S a start jel (101), E a vég jel (101), M a közép jel (01010), L és R számjegyek, melyeket egyenként 7 biten kódolnak. Így egyszerűen meghatározható, hogy egy ilyen kód  $12 \cdot 7 + 2 \cdot 3 + 5 = 95$  biten van kódolva. Az aláhúzott L számjegyet prefixnek nevezzük, míg az aláhúzott R számjegyet hibellenőrző számjegynek. A következő táblázat tartalmazza a számjegyek kódolását a bal, illetve jobb oldali részén a vonalkódnak.

Számjegy	L minta kód	R minta kód
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

A bináris 1 a vonalkódban egy fekete vonalnak felel meg, a bináris 0 a vonalkódban egy szünetnek felel meg. Ha megfigyeljük a táblázatot, láthatjuk, hogy a bal oldali minta, egy adott számjegy esetében, a fordítottja a jobb oldali mintának. Ennek a tulajdonságnak köszönhetően leolvasható a vonalkód akár fejjel lefelé is, mert úgy is ugyanazt az értéket kapjuk.

Az L prefix azt jelzi, hogy a termék milyen típusú, pl. gyógyszerek esetében kötelező módon 3, de más termékek esetében gyártótól függően változhat. Az R hibajavítási számjegy kiszámítását egy példán keresztül szemléltetjük. Adott a kódolni kívánt számjegysorozat: „9876543210R”, melyből az R értéket kell meghatároznunk. Először összeadjuk a páratlan pozíciókon szereplő számjegyeket (1-től számozunk):

$S_1 = 9 + 7 + 5 + 3 + 1 + 9 = 34$ , majd megszorozzuk ezt hárommal és hozzáadjuk a páros pozíciókon szereplő számjegyeket:  $S_2 = 3S_1 + 8 + 6 + 4 + 2 + 0 = 122$ . Ezután meghatározzuk az  $m = S_2 \pmod{10}$  értéket. Ha  $m$  értéke nem nulla, akkor kivonjuk 10-ből és megkapjuk az  $R$  értékét. Ebben a példában  $m = 2 \neq 0 \Rightarrow R = 10 - m = 8$ .

A kód visszafejtése egyszerű, mivel minden számjegynek pontos kódja van, és pontosan meg van határozva a kód felépítése.

Az ilyen típusú vonalkódnak több változata is létezik, a bemutatott változat pontos megnevezése az UPC-A típusú vonalkód.

### A European Article Number

Az EAN-13 az UPC kódolás egy kiterjesztett változata. Ebben az esetben is 12 számjegyet kódolunk, habár a kód 13 számjegyből áll. Az első számjegy meghatározza a többi 12 számjegy kódolási módját.

Első számjegy	Első 6-os csoport	Második 6-os csoport
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGLLG	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

Érdekességképpen megemlítjük, hogy ez a vonalkód egy GTIN-13 számot kódol (Global Trade Item Number), aminek az első három számjegye az esetek többségében a termék származási országát adja meg. Magyarországi termékek esetében ez 599, míg Románia esetében 594.

A számjegyek kódolására a következő táblázatot kell használni:

Számjegy	L kód	G kód	R kód
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Az R kód oszlopban szereplő bináris értékek a bitenkénti tagadásai az L kód oszlopban szereplő értékeknek, a G kód oszlopban szereplő értékek pedig, az R kód oszlopban szereplő értékek tükrözései.

Akárcsak az UPC esetében, itt is az utolsó számjegy egy hibaellenőrző számjegy, melynek a kiszámítása eltér az ott bemutatott módszertől. A következőkben ezt a módszert mutatom be. Ehhez be kell vezetni egy súlyzó táblázatot, melyben 3-sok és 1-sek követik egymást. Legyen a következő kód: „590123412345R”, melyből az R értéket akarjuk meghatározni. Ehhez felépítjük a következő táblázatot.

<b>Pozíció</b>	1	2	3	4	5	6	7	8	9	10	11	12
<b>Súlyzó</b>	3	1	3	1	3	1	3	1	3	1	3	1
<b>Kód</b>	5	9	0	1	2	3	4	1	2	3	4	5
<b>Súlyzó * Kód</b>	15	9	0	1	6	3	12	1	6	3	12	5

Összeadva a táblázat utolsó sorában szereplő értékeket 73-at kapunk, melynek 10-el való osztási maradéka 3. Mivel nem nulla, kivonjuk a 10-ből és megkapjuk a hibaellenőrző kód értékét: 7. Így a teljes kód a következő: „5901234123457”.

A kód visszafejtésénél figyelembe kell venni a kód első számjegyét, melyből el lehet dönteni a kódolás módját. Ezután a visszafejtés hasonló az UPC visszafejtéséhez, mivel hasonló a strukturáltsága.

#### A PDF417 kód

Ez a kód a kétdimenziós vonalkódok családjába tartozik. Sorokból és oszlopokból épül fel, és maximum 2700 byte adatot tud tárolni, innen származik az elnevezése is: „Portable Document Format” (Hordozható dokumentum formátum). A kódolást két részre oszthatjuk: az első részben az adatot kulcsszóvá (codeword) alakítjuk (ez a magas szintű kódolás), a második részben ezt a kulcsszót vonalakká és szünetekké alakítjuk (ez az alacsony szintű kódolás).

Felépítését tekintve 3–90 sort tartalmazhat és 1–30 oszlopot. Létezik hibajavítás benne, mely lehetővé teszi, hogy ne csak detektálni, hanem kijavítani is tudjuk a hibákat. A hibajavítási szintek 0-8-ig terjednek, de a legtöbb esetben 2-es hibajavítási szint elégséges.

Ahogy az egydimenziós vonalkódoknál is létezett start és stop karakter, itt is léteznek. A következő táblázat megadja ezek bináris alakját.

Start karakter	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	0	0	
Stop karakter	1	1	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	1

A magas szintű kódolásnál az adatokat 3 módon tudjuk kulcsszavakba sűríteni.

Sűritési mód	Adat amit sűrít	Sűrités mértéke
Byte	ASCII karakterek 0-255	1.2 byte kulcsszónként
Szöveges	ASCII 9, 10, 13 és 32-127	2 karakter kulcsszónként
Numerikus	Számjegyek 0-9	2.9 számjegy kulcsszónként

A 900-tól 928-ig terjedő kulcsszavaknak speciális jelentésük van, mint pl. különböző sűritési módokra való áttérés.

A szöveges sűrítési módnak van 4 almódja:

1. nagybetűs,
2. kisbetűs,
3. vegyes: numerikus és írásjeles,
4. írásjeles.

Szöveges mód esetén létezik egy kódolási táblázat, mely tartalmazza az ábécé kis- és nagybetűit, írásjeleket és áttérési kódokat az almódok között. Ezt a táblázatot itt nem mutatjuk be, csak a szöveges módbeli kódolási műveletet szemléltetjük egy példán keresztül. Legyen a kódolandó üzenet: „Super !”. A táblázatból megkapjuk a karakterek kódjait: S: 18, LOW: 27, u: 20, p: 15, e: 4, r: 17, SPACE: 26, T\_PUN: 29, !: 10. A LOW azt jelenti, hogy áttérünk kisbetűs almódba, a T\_PUN pedig, hogy a következő karakter írásjel almódban van. Egy kulcsszóba 2 karaktert kódolunk a következő képlet szerint:  $CW = C_1 \cdot 30 + C_2$ . Ha páratlan számú karakterünk van, a végére teszünk „töltelék” áttérési kódokat, pl. T\_PUN. Visszatérve a „Super !” szövegre a következő eredményt

$$CW_1 = 18 \cdot 30 + 27 = 567$$

$$CW_2 = 20 \cdot 30 + 15 = 615$$

kapjuk:  $CW_3 = 4 \cdot 30 + 17 = 137$ , tehát a kulcsszósorozat, mely a „Super !”-t kódolja:

$$CW_4 = 26 \cdot 30 + 29 = 809$$

$$CW_5 = 10 \cdot 30 + 29 = 329$$

567, 615, 137, 809, 329.

Byte mód esetében 256 különböző byte-ot tudunk kódolni, vagyis a teljes kibővített ASCII táblázatot. A kódolás során 6 darab 256-os számrendszerbeli byte-ot átalakítunk 5 darab 900-as számrendszerbeli kulcsszóvá. Legyen a 6-os bytesorozat a következő:  $X_5 X_4 X_3 X_2 X_1 X_0$ , ahol  $X_0$  a legkevésbé fontosabb helyérték. Kiszámoljuk a következő összeget:  $S = X_5 \cdot 256^5 + X_4 \cdot 256^4 + X_3 \cdot 256^3 + X_2 \cdot 256^2 + X_1 \cdot 256 + X_0$ . A kulcsszó értéke az  $S \bmod 900$ , majd az  $S$ -et osztjuk 900-al és így tovább. Most egy példán keresztül szemléltetem az eljárást. Legyen a kódolandó bytesorozat az „alcoo!”, vagyis 97, 108, 99, 111, 111, 108. Az  $S$ -et az előbbi képlet szerint kiszámítva: 107118152609644-et kapunk. Az első kulcsszó tehát  $S \bmod 900$ : 244. Ezután az  $S$ -et elosztjuk 900-al és 119020169566-ot kapunk. A második kulcsszó: 766. Ezt addig ismételjük, míg  $S$  értéke el nem éri a 0-t. Visszafelé haladva egymás után állítva a kulcsszavakat, a következő kódot kapjuk az „alcoo!” esetében: 163, 238, 432, 766, 244.

A numerikus kódolásnál egy 10-es számrendszerbeli számot átalakítunk 900-as számrendszerbeli számmá. 44-es (vagy amennyi marad) számjegyű számmá szervezzük a számjegyeket, teszünk egy 1-est minden szám elé (dekódoláskor ez az egyes eltűnik), s ezeket a számokat 900-as számrendszerbeli számokká alakítjuk. Egy 44 számjegyű számból 15 kulcsszó lesz, míg a kisebb számjegyű számból: (számjegyek száma)/3+1 darab kulcsszó lesz. A kódolás hasonló a byte módban használt kódoláshoz: ha a kódolandó sorozat: 01234, akkor megtoldás után 101234-et kapunk. Ennek a 900-al való osztási maradéka 434, az új szám  $101234/900 = 112$ , ennek a 900-al való osztási maradéka 112. Így a kódolt sorozat 112, 434.

A hibajavító kód a „Reed Solomon” kódokon alapszik. Ha  $l$  szintű kódolást szeretnénk, azt jelenti, hogy  $2^{l+1}$  darab kulcsszóra lesz szüksége a hibajavító kódnak. Így a

maximum kódolandó adatok mérete ennyivel csökken. A Reed Solomon kódok egy olyan polinomiális egyenletre épülnek melynek fokszáma  $2^{l+1}$ , ahol  $l$  a hibajavítási szint értéke. Például  $l=1$  esetén a polinom:  $a + bx + cx^2 + dx^3 + x^4$ . Az algoritmus bonyolultsága miatt és a fontosságát tekintve a PDF417-es kódolásban tovább nem részletezem.

A lineáris vonalkódokhoz hasonlóan itt a kapott kulcsszósorozatban szereplő 1-ek sötét téglalapot, míg a 0-ák fehér téglalapot eredményeznek a grafikus reprezentációban. Erről az átalakításról nem tudunk részletesebb leírást adni, mivel ez a kódolás le van védve, és így nem nyilvános az eljárás.

### A vonalkódok használata és előnyei

A vonalkódok lassan a modern civilizáció nélkülözhetetlen részévé váltak. Használatuk nagyon elterjedt, és a mögöttük rejlő technológia is folyamatosan fejlődik. Lássunk néhány használati területet:

- az üzletekben található termékek azonosítására,
- munkahelyeken az alkalmazottak azonosítására, egybekötve a munkaidő nyilvántartásával,
- automatizált parkolóokban a bérletet váltott személyek azonosítására,
- az egészségügyben a páciensek azonosítására, belekódolva például a személy adatait, azon gyógyszerek listáját melyekre allergiás (ez igen hasznos lehet egy elájult személy esetében),
- dokumentumokon szereplő információk kódolására, így könnyen és gyorsan bevihetők az adatok számítógépbe,
- mobiltelefonokba beépített leolvasókkal vonalkódba kódolt linkek érhetőek el könnyedén (QR vonalkód).

A vonalkódok által kódolt adatok jelentősen megkönnyítik a termékek azonosítását, gyorsan felismerhetők és akár több byte-nyi információt is tárolhatnak.

### Könyvészet

- 1] Barcode, <http://en.wikipedia.org/wiki/Barcode>
- 2] Andrew Langrace Jr., One-dimensional barcode symbology and method of using same, United States Patent, US005479515A, 26.12.1995
- 3] What is quiet zone? [http://whatis.techtarget.com/definition/0,,sid9\\_gci860023,00.html](http://whatis.techtarget.com/definition/0,,sid9_gci860023,00.html)
- 4] What's a barcode? <http://www.dataid.com/autoidpe.htm>
- 5] 2 dimensional barcodes, <http://www.adams1.com/stack.html>
- 6] Universal Product Code, [http://en.wikipedia.org/wiki/Universal\\_Product\\_Code](http://en.wikipedia.org/wiki/Universal_Product_Code)
- 7] EAN-13, <http://en.wikipedia.org/wiki/EAN-13>
- 8] List of GS1 country code, [http://en.wikipedia.org/wiki/List\\_of\\_GS1\\_country\\_codes](http://en.wikipedia.org/wiki/List_of_GS1_country_codes)
- 9] The PDF417 code,
- 10] <http://grandzebu.net/index.php?page=/informatique/codbar-en/pdf417.htm>
- 11] 2D barcode FAQ, <http://www.barcodeman.com/faq/2d.php>

**Aszalos Attila**

Sapientia Erdélyi Magyar Tudományegyetem,  
Marosvásárhelyi Kar, Számítástechnika, IV. év