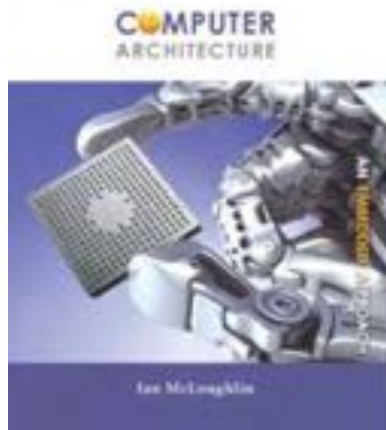


Computer Peripherals

School of Computer Engineering
Nanyang Technological University
Singapore

These notes are part of a 3rd year undergraduate course called "Computer Peripherals", taught at Nanyang Technological University School of Computer Engineering in Singapore, and developed by Associate Professor Kwoh Chee Keong. The course covered various topics relevant to modern computers (at that time), such as displays, buses, printers, keyboards, storage devices etc... The course is no longer running, but these notes have been provided courtesy of him although the material has been compiled from various sources and various people. I do not claim any copyright or ownership of this work; third parties downloading the material agree to not assert any copyright on the material. If you use this for any commercial purpose, I hope you would remember where you found it.

Further reading is suggested at the end of each chapter, however you are recommended to consider a much more modern alternative reference text as follows:



Computer Architecture: *an embedded approach*

**Ian McLoughlin
McGraw-Hill 2011**

Chapter 6. Bar codes

6.1 Introduction

A typical bar-code symbol is illustrated in Figure 0-1 Typical bar-code symbol. Numeric data are represented as a series of bars of varying thickness and separations. For convenience, the numeric data are often printed underneath the bar-code symbol, as shown.

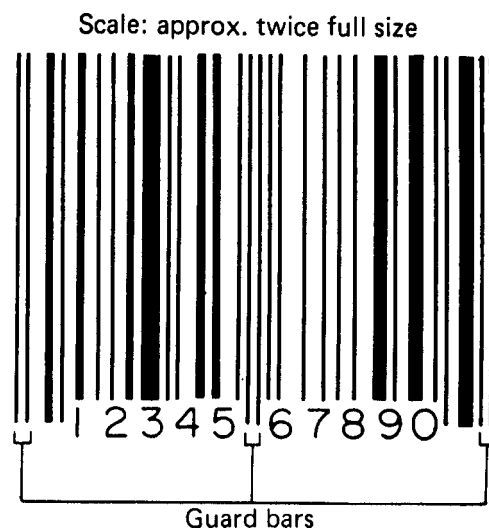


Figure 0-1 Typical bar-code symbol.

The codes are easily read by a light pen, or a scanner. The light pen (or wand) has a light-sensitive tip and when stroked along the bar-code symbol the light and dark bars produce a corresponding sequence of binary voltage levels. This sequence is then decoded to give the numeric data. With the scanner, the object bearing the bar code is moved across a window. Typically this window is set into a work surface or counter. The window is rapidly scanned from below by a narrow beam of light and the reflected light from the bar code is detected by a light sensitive device and converted into a binary sequence of voltage levels. The scanning pattern of the light beam is designed so that at least one valid scan along the code is normally obtained, no matter what the orientation of the code as it is moved past the window. With both the light pen and the scanner arrangement a reject condition is given to the operator if an invalid code is read. The operator may then try again.

Bar codes are usually made up of black bars and white bars in alternating sequence. The widths of these black and white bars usually add up to a fixed width for all characters. This allows the complete code to be scanned at any velocity within reasonable limits, because the timed widths of each bar can be expressed as a fraction of the time to scan a complete

digit. These fractions are relatively independent of velocity provided the scan velocity does not change significantly in the course of a scan.

6.2 Applications

One application of bar codes is in article numbering in supermarkets and other retailing operations. This can be useful at the point of sale, stock room or warehouse. The article identity can be quickly conveyed to a computer. Various improvements in efficiency of the retailing operation can thereby be obtained, such as an automatic update of stock levels, automatic re-ordering of new stock, better marketing information, and the ability to carry reduced levels of stock because of quicker replenishment of stock. This book carries a bar code on the rear outside cover. Bar codes are also used in library systems. Each book is labelled with a unique bar code which is read when the book is issued and returned. This eliminates the form filling and record keeping normally associated with these operations. Book records can now be kept in a computer data base, which permits automatic production of overdue-book notifications, generation of statistics, and other benefits. Other applications of bar codes include control of the dispensing of drugs within a hospital.

6.3 Symbologies

A bar code symbol is a parallel arrangement of varying width bars and spaces. "Symbology" is the term used to describe the unambiguous rules specifying the way that data is encoded into the bar and space widths.

Symbology is analogous to language. When humans communicate via the spoken or written word, any language can be used provided that both parties agree to and are proficient in the choice. The same concept follows in bar code: depending on the data to be communicated, several different symbologies can be used. Communication obviously cannot occur unless the readings and printing equipment use a compatible symbology. This is an important consideration because dozens of different bar code symbologies have been developed since the technology's inception in the late 1940s.

6.3.1 Characteristics of a Symbology

6.3.1.1 Character Set

The term "character set" describes the range of data characters that can be encoded into a given symbology. Some symbologies can encode only numbers and are therefore termed "numeric" symbologies. Other symbologies can encode alphanumeric information, while others support the entire 128 unique codes of the ASCII character set.

6.3.1.2 Symbology Type

Bar code symbologies fall into two general categories: discrete and continuous.

In a **discrete code**, each character can stand alone and can be decoded independently from the adjacent characters. Each character is separated from its neighbour by loosely tolerated intercharacter gaps, which contain no information. Every character has a bar on each end. In the decoding process, each character is treated individually. See Figure 3.1.2.1.

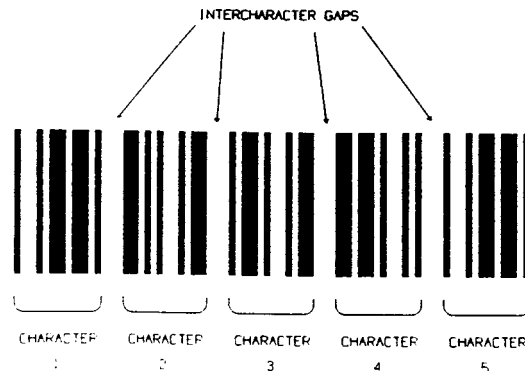


Figure 0-2 : Discrete Symbology.

Because the intercharacter gaps are loosely tolerated, a discrete code can be printed by a variety of different techniques. A discrete code can easily be produced by a printer that prints an entire character at one time, such as a movable font device.

A **continuous code** has no intercharacter gaps. Every character starts with a bar and ends with a space or vice versa, as shown in Figure 0-3 : Continuous Symbology. The end of one character is indicated by the start of the next character.

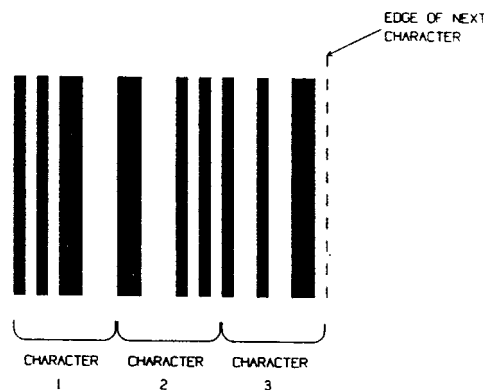


Figure 0-3 : Continuous Symbology.

In a continuous bar code symbology there are no intercharacter gaps. Every character begins with a bar and ends with a space. Some form of termination pattern starting with a bar must provide the last “edge of next character” for the symbol's last character.

Since there are no intercharacter gaps, a continuous code requires less symbol length to encode a given amount of data. Offsetting this density advantage is the fact that the range of available demand printing technologies is more restricted for continuous codes than it is for discrete symbologies.

6.3.1.3 Number of Element Widths

In a bar code symbol, data is conveyed in the widths of the bars and spaces. Two basic types of bar codes exist: those that employ only two element widths (wide and narrow), and those that employ multiple widths.

In a **two-width** symbology, the ratio between wide and narrow element widths is called N . N is typically allowed to vary over some range (usually from 2.0 to 3.0), but must be constant for a given symbol. As N gets larger, the allowable printing tolerance also increases.

In a **multiple-width** symbology, bars and spaces can assume several different width values. Most multiple-width symbologies are modular, meaning that the length of a character is subdivided into a predetermined number of modules, and a bar or space width is always an integral number of modules. Multiple-width symbologies are usually continuous and are often decoded using edge-to-similar-edge algorithms. This technique involves the measurement of distances between similar edges of adjacent elements rather than the measurement of actual element widths.

Continuous codes with multiple element widths are often called (n,k) codes, where n refers to the number of modules in a character width and k to the number of bars (and spaces). The total number of possible patterns in an (n,k) code can be determined from the expression:

$$\frac{(n-1)!}{(2k-1)!(n-2k)!}$$

6.3.1.4 Fixed or Variable Length

Some symbologies by their very structure encode only messages of a fixed length. Other symbologies should only be used in a fixed length environment because of data security considerations, while other symbologies can be used to encode true variable length data.

6.3.1.5 Density

Bar code symbologies differ in the amount of data that can be encoded in a given unit of length. In order to allow meaningful comparisons, the value of X (width of a narrow bar or space) needs to be considered when examining relative densities.

Note that density is normally specified only for the data characters. The overall length of a symbol needs to include start/stop characters, quiet zones, and any check characters.

6.3.1.6 X

X is the term used to describe the nominal width of a bar code symbol's narrow elements (both bars and spaces). It is often calculated as the average width of the symbol's narrow elements and by convention is expressed in mils (thousandths of an inch).

6.3.1.7 Self Checking

A symbology is termed self-checking if a single printing defect will not cause a character to be transposed into another valid character in the same symbology.

6.3.2 Classification on Barcode Symbologies

6.3.2.1 Numeric-only barcodes

1. Code 11
2. Interleaved 2 of 5

3. Industrial 2 of 5
4. Standard 2 of 5
5. Codabar
6. Plessey
7. MSI
8. Postnet

6.3.2.2 Retail barcodes

1. EAN-8
2. EAN-13
3. UPC-A
4. UPC-E

6.3.2.3 Alphanumeric barcodes

1. Code 39
2. Code 93
3. Code 128
4. LOGMARS

6.3.2.4 2-Dimensional barcodes

1. PDF417
2. DataMatrix
3. Data Code
4. Maxicode
5. Code 49
6. 16K

6.4 UPC

An American format, used for article numbering, is that of the universal product code (UPC). One version of the UPC is:

S	xxxxx	xxxxx	C	
			+	----- (check digit)
		+	-----	(product number)
	+	-----		(manufacturer number)
+	-----			(number-system character).

The manufacturer and product numbers occupy respectively the left and right halves of the symbol. The numbers are also printed underneath the symbol.

The number-system character identifies the symbol version and also the type of article. The check digit is used to detect errors. The check digit and number-system character are not printed underneath the symbol.

The Universal Product Code (UPC) has been successfully employed in the supermarket industry since 1973. UPC is a coding system as well as a symbology; it is designed to uniquely identify a product and its manufacturer.

UPC is a fixed length, numeric, continuous symbology employing four element widths. Two common types of UPC are Version A, which encodes 12 digits, and Version E, which encodes 6 digits. Figure 0-4 : UPC Version A Symbol Encoding The Data "012345678905" illustrates the data arrangement in a Version A symbol.

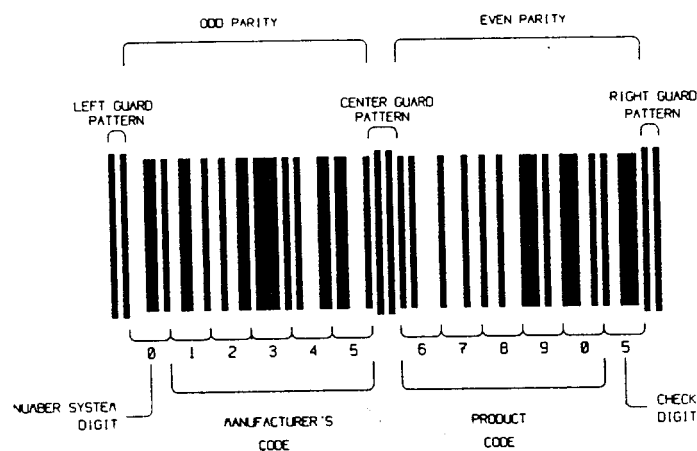


Figure 0-4 : UPC Version A Symbol Encoding The Data "012345678905"

6.4.1 UPC Version A

A typical **UPC Version A** symbol consists of two halves representing a total of 12 numeric digits. The two six-digit halves are surrounded by left, centre and right guard patterns. The various guard bars can be thought of as start/stop patterns. The left half uses the odd parity encodings of digits, and the right half uses the even parity encodings of digits.

The first numeric digit identifies the UPC number system digit being used:

- 0: regular UPC codes
- 1: reserved
- 2: random weight items marked at the store
- 3: National Drug Code and National Health Related Items code
- 4: no format restrictions, for in-store use on non-food items
- 5: for use on coupons
- 6: reserved
- 7: regular UPC codes
- 8: reserved

- 9: reserved

The next group of 5 digits identifies the manufacturer (the manufacturer's code). This number is assigned by the Uniform Code Council (UCC). The first five digits of the right half of the pattern are the product code digits and are assigned by the manufacturer, and the final digit is a Modulo 10 checksum check digit used for error detection. The check digit whose value is evaluated mathematically based on all of the other numbers encoded in the symbol. A weighting scheme is used in its calculation, so that the check digit also protects against transposition errors if the data is manually entered. Although UPC is a continuous symbology the left and right halves of the pattern can be independently decoded.

6.4.1.1 Modulo-10 checksum

The UPC bar code standard incorporates a **modulo-10 error check character**, which needs to be taken into account in the data recovery software. The modulo-10 algorithm is as follows:

1. Add the values of the digits in odd positions 1, 3, 5, 7, 9, and 11 to S1.
2. Multiply this result by 3: $S2 = S1 \times 3$.
3. Add the values of the digits in even positions 2, 4, 6, 8, and 10 to S3.
4. Sum the results of steps 2 and 3: $S4 = S3 + S2$.
5. The check character is the smallest number which, when added to the result in step 4, produces a multiple of 10: $\text{Check Digit} = 10 - \text{Remainder} (S4/10)$

Example: Assume the barcode data = 01234567890

1. $0 + 2 + 4 + 6 + 8 + 0 = 20$
2. $20 \times 3 = 60$
3. $1 + 3 + 5 + 7 + 9 = 25$
4. $60 + 25 = 85$
5. $85 + X = 90$ (next highest multiple of 10), therefore $X = 5$ (checksum)

A UPC-A code may be augmented with a 2-digit or 5-digit extension. These extensions are handled as separate barcodes and are generally used to create ISBN's (International Standard Book Numbers).

The UPC-A data is encoded as two bars and two spaces in a character that is seven modules wide. It is a (7,2) code, which has 20 unique patterns. Bar and space widths may be 1, 2, 3, or 4 modules wide. Figure 0-5 illustrates the pattern for "6" encoded for either the left or right half of the symbol.

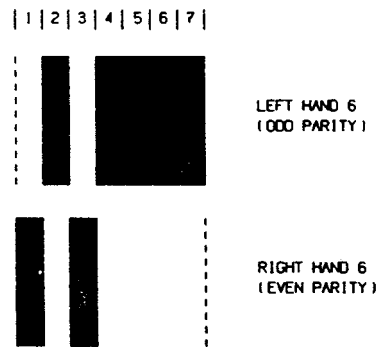


Figure 0-5 : UPC version A Encodation of the Digit “6”.

Encodation differs for the left and right halves of the symbol, as shown in Figure 0-6 : UPC Version A Encodation Patterns. Note that the term "parity" refers to whether the sum of the bar module widths in a given character is an even or an odd number.

Because the bar height of either half of a UPC Version A symbol is greater than the width of the six digits, omnidirectional scanning is possible if orthogonal scanning lines are employed. The scanner can independently decode the right and left halves of the symbol because an examination of the data's parity indicates which half was decoded. The guard bars are usually printed with greater length than other bars in the symbol. This is done to maximise the allowable scanning tilt angle.

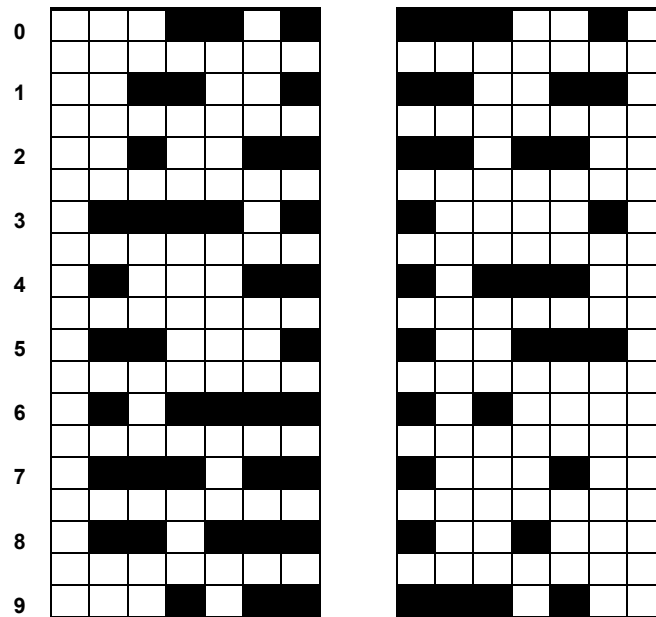


Figure 0-6 : UPC Version A Encodation Patterns.

6.4.2 UPC Version E

UPC Version E encodes six digits and is suited for identifying products in small packages. UPC-E is a variation of the UPC-A symbol that is used for number system 0. By suppressing zeroes, UPC-E codes can be printed in a very small space and are used for labelling small items. The symbol layout is illustrated in Figure 0-7 : UPC version E symbol.

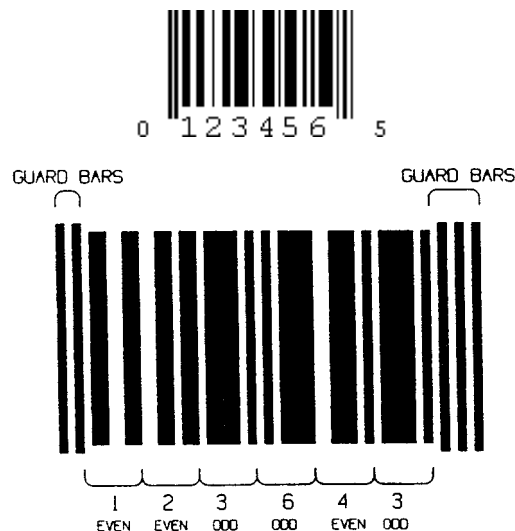


Figure 0-7 : UPC version E symbol.

UPC version E is a 6-digit variation of the UPC symbology used for number system. It is often used to label small items. In this example the number 12300-00064 has been compressed to the shorter numeric code 123643. The last digit (the 3) indicates the type of compression.

In addition to the requirement that the first digit of the barcode (number system) must be zero, there are four rules that determine what UPC codes can be printed using the compressed UPC-E format:

- If the last 3 digits of the manufacturer's number are 000, 100, or 200, the valid product code numbers are 00000 - 00999 (1,000 numbers).
- If the last 3 digits of the manufacturer's number are 300, 400, 500, 600, 700, 800, or 900, the valid product code numbers are 00000 - 00099 (100 numbers).
- If the last 2 digits in the manufacturer's number are 10, 20, 30, 40, 50, 60, 70, 80, or 90, the valid product code numbers are 00000 - 00009 (10 numbers).
- If the manufacturer's number does not end in zero, the valid product code numbers are 00005 - 00009 (5 numbers).

Guard bars precede and follow the data in the pattern. The parity of the encodation follows the pattern shown. Because of the data compression process used, version E symbol is often referred to as a zero-suppressed symbol

The six digits are enclosed between two left-hand guard bars and three right hand guard bars. A symbol check character is encoded into the parity of the six data digits. Of the six digits, three have even parity and three have odd parity.

The nominal X dimension employed in all UPC symbols is 13 mils. An allowable magnification factor of 0.8 to 2.0 results in a range of printable X values between 10.4 and 26 mils.

6.5 EAN

6.5.1 IMPORTANT NOTE

On June 9, 1997 the Uniform Code Council announced that January 1, 2005 will be the 'sunset date' for expansion of UPC to EAN-13 (known as UPC-13 in the USA). This expansion is designed to provide a larger pool of numbers for US manufacturers and to eliminate the need for manufacturers who export goods to the US and Canada to double-label their products.

Most scanners sold as early as 1985 can auto-discriminate between these two codes, so the hardware will generally not be an issue. The primary concern is that product databases used by retailers will need to accommodate all EAN numbers; many database designs currently store only 12 digits. In order to guard against any confusion between an existing UPC code, a new UPC-13 code, and a previously assigned EAN-13 code it is necessary to use all 13 digits as a lookup key.

6.5.2 EAN codes

The European Article Numbering system (abbreviated as EAN) is a superset of UPC. An EAN scanner can decode UPC, but the inverse is, unfortunately not necessarily true.

The European article number (EAN) system includes a two-digit code which identifies the country, and is designed to be compatible with the American UPC.

EAN has two versions: EAN-13 and EAN-8, encoding 13 and 8 digits respectively. An EAN-13 symbol contains the same number of bars as UPC Version A, but encodes a thirteenth digit into a parity pattern of the left-hand six digits. This thirteenth digit, in combination with the twelfth digit, defines two flag characters that represent a country code. For compatibility with UPC, flags 00, 01, 03, 04, and 06 through 09 are assigned to the U.S.

6.5.2.1 EAN-13 code

EAN-13 is used world-wide for marking retail goods. It encodes 13 characters: the first two are a country code, followed by 10 data digits and a checksum. 2-digit and 5-digit supplemental barcodes may be added for a total of 14 or 17 data digits. Similar to UPC, the checksum is a Modulo 10 calculation:

1. Add the values of the digits in the even-numbered positions: 2, 4, 6, etc.
2. Multiply this result by 3.
3. Add the values of the digits in the odd-numbered positions: 1, 2, 5, etc.
4. Sum the results of steps 2 and 3.
5. The check character is the smallest number which, when added to the result in step 4, produces a multiple of 10.

Example: Assume the barcode data = 123456789012

1. $2 + 4 + 6 + 8 + 0 + 2 = 22$
2. $22 * 3 = 66$
3. $1 + 3 + 5 + 7 + 9 + 1 = 26$
4. $66 + 26 = 92$
5. $92 + X = 100$ (nearest equal or higher multiple of 10), therefore $X = 8$ (checksum)



Figure 0-8 EAN-13

6.5.2.2 EAN-8

EAN-8 is a shortened version of the EAN-13 code. An EAN-8 symbol has a left-hand guard pattern, four odd parity digits, a centre guard pattern, four even parity digits, and a right-hand guard pattern. An EAN-8 symbol encodes two flag digits (2-digit country code), five data digits, and one check digit. An EAN-8 symbol is shown in Figure 0-9 : EAN 8 Symbol Encoding the Data "4015347." 2-digit and 5-digit extension barcodes may be added.

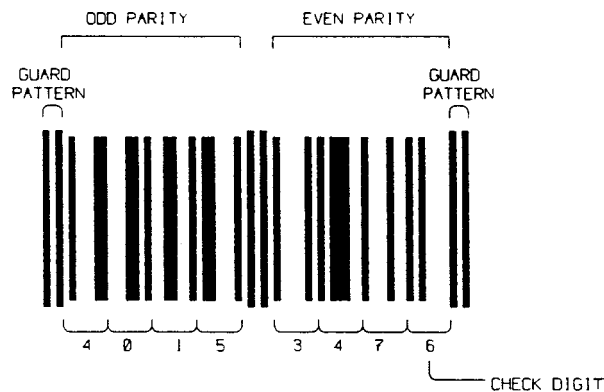


Figure 0-9 : EAN 8 Symbol Encoding the Data "4015347."



Figure 0-10 EAN-8

UPC/EAN has proven to be extremely successful in the retail marketplace. In normal usage, every code from a supermarket scanner is verified against an in store computer's database before being acted upon. In this mode, very high data security has been reliably demonstrated.

6.6 Code 39

Code 39 was the first alphanumeric symbology to be developed. Code 39 is widely used in many industries and is the standard for many government barcode specifications, including the U.S. Department of Defense. Code 39 is defined in American National Standards Institute (ANSI) standard MH10.8M-1983, and is also known as USD-3 and 3 of 9. It is now the 'de facto' non-retail symbology. It is a discrete, two widths, selfchecking, variable length symbology that can readily be printed by a variety of technologies.



Every Code 39 character has five bars and four spaces. Of these nine elements, 3 are wide and 6 are narrow.



Figure 0-11 : Code 39 Character "I."

Figure 0-11 is a typical Code 39 character encoded according to the pattern for "I" shown in Figure 0-12 Code 39 Encodation Patterns. The "1" and "0" notations show how the typical Code 39 pattern may be represented as a sequence of 9 binary bits in a computer memory.

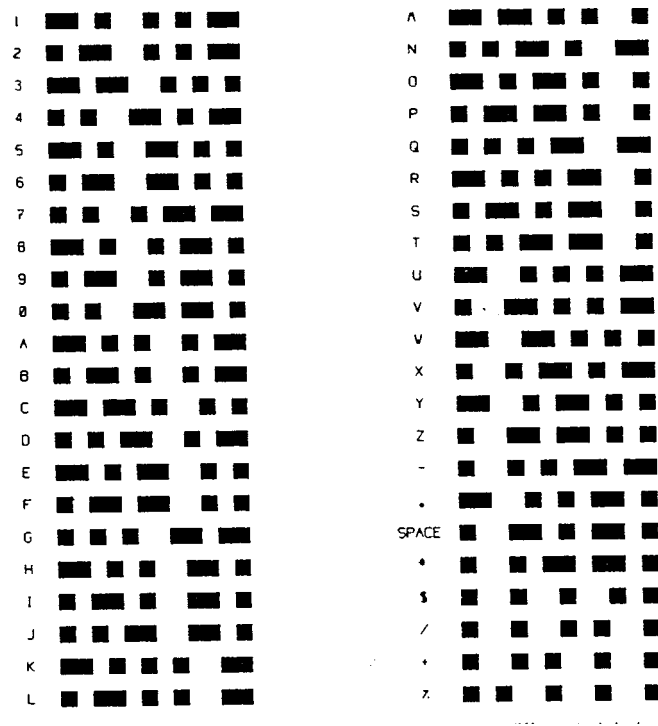


Figure 0-12 Code 39 Encodation Patterns.

Code 39 encodes 44 different alphabetic but only 43 is used, numeric and graphic characters. Every character encodation has three wide elements and six narrow elements out of 9 total elements, hence the name "Code 39".

A Code 39 symbol begins and ends with an asterisk (*), which is this symbology's start/stop code. Figure 0-13 : Code 39 Symbol Encoding The Data "CODE" is an example of a Code 39 symbol encoding the data message "CODE."



Figure 0-13 : Code 39 Symbol Encoding The Data "CODE"

Each character is separated from its neighbour by a loosely toleranced intercharacter gap that contains no information. Because of the mirror image relationship between the start/stop character and the letter "P," an upper limit is specified for the intercharacter gap width in order to prevent short reads, especially in the case of partial scans.

Although there are only 43 data characters in Code 39's character set, it is possible to encode all 128 ASCII characters using Code 39's Full ASCII feature. If a reader is in its Full ASCII mode, the symbols \$, /, %, and + are used as precedence codes with the 26 letters.

Breaking up long bar code messages into two or more shorter symbols for ease of reading is sometimes advantageous. If the first data character of a Code 39 symbol is a space, the bar code reader appends the information contained within the symbol (excluding the leading space) to its buffer. This operation continues for all symbols that contain a leading space. When a Code 39 symbol is read that does not contain a leading space, the data is appended to the reader's buffer, and the entire buffer is transmitted and cleared for new data.

Code 39 is self-checking (a single printing defect cannot cause a substitution error) and is normally not used with a check character. An optional modulo 43 check character is defined for use in specific applications that require exceptional data security. As a general policy, the use of Code 39's check character is encouraged, especially if the print quality is less than optimum.

6.6.1 Code 39 Specification

Code 39 is a discrete, variable length, self-checking, alphanumeric symbology employing two-element widths. Every character has five bars and four spaces; three of the elements in any given character are wide, and six are narrow. Every symbol starts and ends with a unique start/stop character. A summary of Code 39's main characteristics is as follows:

- ◆ Character Set: 26 uppercase letters, 10 digits, 7 special characters; Extendable to full 128 ASCII character set using a two-character precedence code scheme.
- ◆ Symbol Length: Variable
- ◆ Check Characters: Optional
- ◆ Overhead Characters: 2 per symbol
- ◆ Other Features: Concatenation ability
- ◆ Density: Maximum of 9.8 characters per inch when printed using a 7.5 mil X dimension Code 39 Symbol Description Each

6.6.2 Code 39 Symbol Description:

Code 39 symbol consists of:

- ◆ Leading quiet zone
- ◆ Start character
- ◆ Data characters
- ◆ Stop character
- ◆ Trailing quiet zone

Figure 0-13 : Code 39 Symbol Encoding The Data "CODE" illustrates a typical Code 39 symbol.

6.6.3 Code 39 Encodation

Each Code 39 character consists of five bars and the four intervening spaces; three of the nine elements are wide, and 6 are narrow. Each character has either two or zero wide bars. Figure 0-12 Code 39 Encodation Patterns. shows the encodation of bar and space patterns for each character code. The asterisk character is exclusively used as a start/stop code.

6.6.4 Code 39 Dimensions

The minimum value of X is 7.5 mils for symbols used in open systems. X shall be constant throughout a symbol.

The ratio of the nominal-wide element width to the nominal narrow-element width is designated as N and must remain constant throughout a symbol. The allowable range of N is 2.0 to 3.0, but N must exceed 2.2 if X is less than 20 mils.

The minimum width of the intercharacter gap is $(X - t)$, where t is the applicable printing tolerance and the maximum is:

- ◆ 5.3X for values of X less than 10 mils
- ◆ 3X or 53 mils (whichever is greater) for values of X equal to or greater than 10 mils

For open systems, the minimum bar height is 15 percent of the symbol length or 0.25 inches, whichever is greater. The minimum quiet zone width is 10X. For optimum hand-scanning, the quiet zone width should be at least 0.25 inches. The overall length of a Code 39 symbol is a function of X, N, the intercharacter gap width, and the number of data characters, as given by:

$$L = (C+2) (3N+6) X + (C + 1) I$$

where

L = length of symbol, excluding quiet zones

C = number of data characters

X = X dimension

N = wide-to-narrow ratio

I = intercharacter gap width

Many printers generate symbols with an intercharacter gap width of X. In this case, the expression for symbol length can be simplified to

$$L = ((C + 2) (3N + 7) - 1) X$$

6.6.5 Code 39 Dimensional Tolerances

The allowable printing width tolerances are given by the expression

$$t = \pm \frac{4}{27} \left(N - \frac{2}{3} \right) X$$

where

t = the allowable width deviation about the nominal dimension of a given element

X = X dimension

N - wide-to-narrow ratio

The tolerance t applies to the printed width of all elements, both wide and narrow. These tolerances should not accumulate within a single character. The overall character width shall not deviate from nominal by more than $2t$.

For measuring purposes, the edge of a bar/space transition is defined as the centre of a circular sampling aperture of diameter no greater than $0.8X$ when the apparent reflectance of the sample viewed through the aperture is halfway between the maximum and minimum reflectance obtained by that aperture on the adjacent bar and space.

6.6.6 Code 39 Optical Properties

Reflectivity measurements are made at a wavelength of 633 nanometers with illuminating and viewing axes separated by 45 degrees, with one of the axes positioned perpendicular to the sample surface. The aperture of each of the two optical systems should subtend an angle of no more than 15 degrees. The measured diffuse reflectance is defined as the ratio of the light reflected from the sample to that reflected from a specially prepared Magnesium Oxide or Barium Sulphate standard measured under the same conditions. One of the optical paths must restrict the sample field to an area equal to that of a circle of $0.8X$, and the other path's sample field must have an area greater than that of a circle of $8X$.

The maximum bar reflectance is 30 percent. The minimum space reflectance is 25 percent.

The minimum reflectance difference (MRD) is equal to the reflectivity difference between the darkest space and the lightest bar. The minimum value of MRD is 37.5 percent if X is less than 40 mils, and 20 percent if X is equal to or greater than 40 mils.

6.6.7 Code 39 Full ASCII Mode

If suitably programmed, a Code 39 reader can decode the 128 ASCII character set by using paired character sequences made up of the character \$, ., %, or / followed by one of the 26 letters, as shown in Table 1 CODE 39 ASCII chart.

Table 1 CODE 39 ASCII chart

CODE 39 ASCII CHART							
ASCII	CODE	ASCII	CODE	ASCII	CODE	ASCII	CODE
NUL	%U	SP	SPACE	@	%V	`	%
SOH	\$A	!	/A	A	A	a	+A
STX	\$B	“	/B	B	B	b	+B
ETX	\$C	#	/C	C	C	c	+C
EOT	\$D	\$	/D	D	D	d	+D
ENQ	\$E	%	/E	E	E	e	+E
ACK	\$F	&	/F	F	F	f	+F
BEL	\$G	‘	/G	G	G	g	+G
BS	\$H	(/H	H	H	h	+H
HT	\$I)	/I	I	I	i	+I
LF	\$J	*	/J	J	J	j	+J
VT	\$K	+	/K	K	K	k	+K
FF	\$L	,	/L	L	L	l	+L
CR	\$M	-	-	M	M	m	+M
SO	\$N	.	.	N	N	n	+N
SI	\$O	/	/O	O	O	o	+O
DLE	\$P	0	0	P	P	p	+P
DC1	\$Q	1	1	Q	Q	q	+Q
DC2	\$R	2	2	R	R	r	+R
DC3	\$S	3	3	S	S	s	+S
DC4	\$T	4	4	T	T	t	+T
NAK	\$U	5	5	U	U	u	+U
SYN	\$V	6	6	V	V	v	+V
ETB	\$W	7	7	W	W	w	+W
CAN	\$X	8	8	X	X	x	+X
EM	\$Y	9	9	Y	Y	y	+Y
SUB	\$Z	:	/Z	Z	Z	z	+Z
ESC	%A	;	%F	[%K	{	%P
FS	%B	<	%G	\	%L		%Q
GS	%C	=	%H]	%M	}	%R
RS	%D	>	%I	^	%N	~	%S
US	%E	?	%J	_	%O	DEL	%T,%X,%Y, %Z

Note: Character pairs /M and /N decode as a minus sign and a period respectively.
Character pairs /P through /Y decode as the digit 0 through 9.

6.6.8 Code 39 Check Character

An optional check character is defined for Code 39 applications requiring higher levels of data security. When used, the check character is positioned between the final data character and the stop character. The check character is determined as follows:

Table 2 Code 39 Character set

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

- ◆ Using Table 2 Code 39 Character set. determine the numerical value of each of the data characters.
- ◆ Sum the values of all the data characters.
- ◆ Find the modulo 43 remainder of the sum from above.
- ◆ The check character is that character whose value from Table 1 CODE 39 ASCII chart is equal to the remainder.

As an example, if the data characters are “TEST”, the values would be summed as $29+14+28+29=100$. 43 goes into 100 twice, with a remainder of 14. The letter “E” has a value of 14 is therefore the check character.

6.6.9 Code 39 Symbol Concatenation

It is sometimes advantageous to break up long messages into multiple, shorter symbols. If the first data character of a Code 39 symbol is a space, the reader may be programmed to append the information contained in the remainder of the symbol to a storage buffer (data not transmitted). This operation continues for all successive symbols with a leading space, with messages being added to the end of previously stored messages. When a message is read which does not contain a leading space, the contents are appended to the buffer, the entire buffer is transmitted, and the buffer is cleared.

6.6.10 Code 39 Dimensional Verification

The following procedure should be used to determine if a Code 39 symbol meets the specified dimensional tolerances:

1. Measure and record the perceived width for every element, the intercharacter gaps, and the two quiet zones.
2. Decode the symbol, confirming that there are valid start, stop, and data characters.
3. As determined by decoding, sort the widths of all the characters' elements (excluding intercharacter gaps and quiet zones) into four groups: narrow bars, narrow spaces, wide bars, and wide spaces.
4. If it is desired to evaluate symbol quality against values of X and N specified in an application standard, those values can be used directly in step 9, skipping steps 5 through 8.
5. Compute the average width in each group:
 - ANB = average narrow bar
 - ANS = average narrow space
 - AWB = average wide bar
 - AWS = average wide space
6. Compute X as follows:

$$X = \frac{ANB + ANS}{2}$$
7. Compute N as follows:

$$N = \frac{AWB + AWS}{2X}$$
8. Confirm that X and N are within allowed ranges:
 - a. Check that X is greater than or equal to 0.0075"; if not, declare X = 0.0075".
 - b. Check that N is greater than or equal to 2.0; if not, declare N = 2.0.
 - c. If X is less than 0.020", then check that N is greater than or equal to 2.2; if not, declare either X = 0.020" or N = 2.2.
 - d. Check that N is less than or equal to 3.0; if not, declare N = 3.0.

9. Compute the printing tolerance:

$$t = \pm \frac{4}{27} \left(N - \frac{2}{3} \right) X$$

10. Confirm that all measured widths fall within allowed limits, otherwise the symbol is not in conformance:
- Check that all narrow elements fall within $X - t$ to $X + t$.
 - Check that all wide elements fall within $NX - t$ to $NX + t$.

If X or N were changed in step 8c, it might be necessary to use the opposite assumption in order to meet the requirement that all the measured widths fall within the allowed limits.

- Confirm that the sum of the widths of any number of adjacent elements within any character does not deviate from nominal (using X and N as measured or assumed above) by more than $2t$, otherwise the symbol is not in conformance.
- Confirm that all intercharacter gap widths (from step 1) are equal to or greater than $X - t$, otherwise the symbol is not in conformance.
- If X is less than 10 mils, confirm that all intercharacter gaps are less than $5.3X$, otherwise the symbol is not in conformance. If X is greater than or equal to 10 mils, confirm that all intercharacter gaps are less than 0.053" or $3X$, whichever is greater, otherwise the symbol is not in conformance.
- Confirm that each quiet zone is at least $10X$ wide, otherwise the symbol is not in conformance.

Note: In step 8a, the 7.5 mil limit is consistent with present practice for open systems. In closed applications, a lower limit may be used.

6.7 PDF-417



PDF-417 is a two-dimensional barcode which can store up to about 1,800 printable ASCII characters or 1,100 binary characters per symbol. The symbol is rectangular; the shape of the symbol can be adjusted to some extent by setting the width and allowing the height to grow with the data. It is also possible to break large amounts of data into several PDF-417 symbols which are logically linked. There is no theoretical limit on the amount of data that can be stored in a group of PDF-417 symbols.

The capacity of PDF-417 can be helpful in applications where the data must travel with the labelled item, where a host database is not always available for quick look-up. PDF-417 is being used for hazardous materials labelling; storing technical specifications and calibration

data on electronic instruments; encoding fingerprints and photographs on the backs of drivers' licenses.

The maximum data density is determined by the smallest elements which can be reliably printed and scanned. Using the smallest recommended element size of 0.0075 inch wide and 0.010 inch high, the maximum data density in the binary mode is 686 bytes per square inch (106.2 bytes per square centimetre). In the printable ASCII mode the density is 1,144 characters per square inch (177.2 characters per square centimetre).

PDF-417 symbols require a 2-D scanner; they cannot be read using an ordinary linear barcode scanner. A number of scanners are on the market using both laser and CCD camera technologies. PDF-417 symbols can be printed using most professional-grade thermal transfer label printers on the market.

6.8 DataMatrix



DataMatrix is a two-dimensional barcode which can store from 1 to about 2,000 characters. The symbol is square and can range from 0.001 inch per side up to 14 inches per side. As an example of density, 500 numeric only characters can be encoded in a 1-inch square using a 24-pin dot matrix printer. The sample shown above encodes 20 ASCII characters.

DataMatrix is being used to encode product and serial number information on electrical rating plates; to mark of surgical instruments in Japan; to identify lenses, circuit boards, and other items during manufacturing.

DataMatrix symbols require a 2-D scanner; they cannot be read using an ordinary linear barcode scanner. A number of scanners are on the market using both laser and CCD camera technologies. DataMatrix symbols can be printed using most professional-grade thermal transfer label printers on the market.

6.9 BAR CODE READER

The bar codes are read either by holding the product stationary and passing a bar wand over the pattern, or alternatively by passing the product across an opening from which a light beam strikes the bar code. Figure 0-14 UPC scanner. shows this latter type of scanner, in which several laser light sources are directed in the general direction of the product, at different orientations (typically a star pattern).

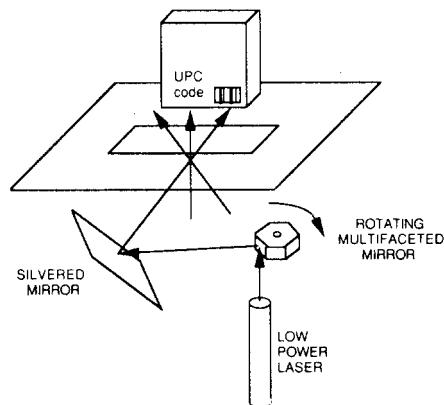


Figure 0-14 UPC scanner.

In this manner, one of the laser beams will usually strike the bar code pattern as an angle suitable for reading; if none of the laser beams strike the bar code at a suitable angle, then the product is passed by the opening again until a successful read occurs. There is a backup procedure, in that repeated failures can always be rectified by entering the numeric data manually.

The bar code reader (wand or fixed scanner) will produce an asynchronous waveform output of widely varying and unknown data rate, in the range 0.2-20 cm/second. A zero output could either be an unwanted noise spike, or the beginning of an authentic scan. If it represents the beginning of a scan, then the total scan time will lie in the range 3-300 milliseconds ($11 \times 0.033 \text{ cm/scan rate}$).

At the commencement of the first LO to HI transition, a 'bar timing loop' is started, and at the next HI to LO transition, a 'space timing loop' is started. This procedure is repeated 30 times, until 60 time intervals have been recorded (30 bars and 30 spaces), as indicated in Figure 0-16. The appropriate algorithm is shown in Figure 0-15.

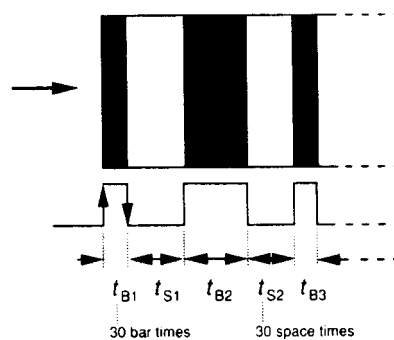


Figure 0-15 Bar Code scan times

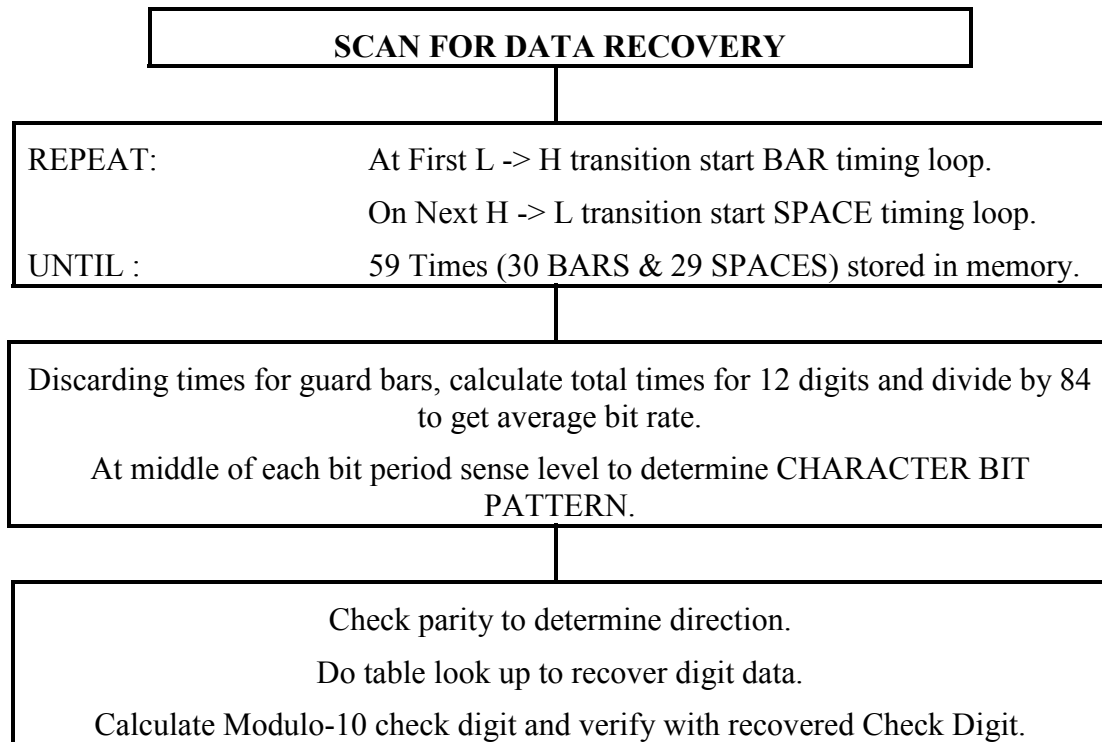


Figure 0-16 Pseudo-code for UPC data recovery

Once 60 times have been stored in the table, they need to be converted first to bar widths, and then to bit patterns (digit information).

The per character scan rate will be fairly constant, so that the average time per module (black or white) will be one seventh of the total character scan time (since each character is comprised of seven modules). The left, right and centre guard bands must be omitted from bar width calculations.

6.9.1 Edge-to-similar-edge algorithm



Figure 0-17 T distance in the edge-to-similar-edge algorithm

UPC is intended to be decoded using the edge-to-similar-edge algorithm. For each character, the first three timing T_1 , T_2 and T_3 are fixed in term of number of modules and T_4 is dependent on the next character and hence not use in recognition. Unfortunately, the

left hand encodation of digits 1 and 7 possess the same T distances. A similar problem exists with the digits 2 and 8; and when they occur in the right half of the symbol. To avoid this ambiguity, a UPC reader also needs to examine element widths when these problematic characters are encountered. (See Figure 0-18 T timing for odd and right parity UPC.)

	odd parity				even parity		
	T1	T2	T3		T1	T2	T3
0	5	3	2				
1	4	4	3				
2	3	3	4				
3	5	5	2				
4	2	4	5				
5	3	5	4				
6	2	2	5				
7	4	4	3				
8	3	3	4				
9	4	2	3				

Figure 0-18 T timing for odd and right parity UPC.