1. Introduction

A bar code is a series of parallel, adjacent bars and spaces with encoding text information that is easily read by electronic readers called Barcode Scanner. Bar coding allows data to be collected rapidly and with extreme accuracy. Predefined bar and space patterns or "symbologies" are used to encode small strings of character data into a printed symbol. Bar codes can be thought of as a printed type of the Morse code with narrow bars (and spaces) representing dots, and wide bars representing dashes. A bar code reader decodes a bar code by scanning a light source across the bar code and measuring the intensity of light reflected back by the white spaces. The pattern of reflected light is detected with a photodiode which produces an electronic signal that exactly matches the printed bar code pattern. This signal is then decoded back to the original data by the electronic circuits. Due to the design of most bar code symbologies, it does not make any difference if bar code is scanned a bar code from right to left or from left to right.

The basic structure of a bar code consists of a leading and trailing quiet zone, a start pattern, one or more data characters, optionally one or two check characters and a stop pattern.

There are a variety of different types of bar code encoding schemes or "symbologies", each of which were originally developed to fulfill a specific need in a specific industry. Several of these symbologies have matured into de-facto standards that are used universally today throughout most industries.

The different symbologies have different capabilities for encoding data. For example the UPC symbology used to identify retail products always contains 12 numeric digits whereas the general purpose Code 39 or Code 128 bar code symbologies can encode variable length alphanumeric data up to about 30 characters in length. These types of bar codes are called "linear symbologies" because they are made up of a series of lines of different widths. **There are several types of Barcode Scanners available in market which may or may not be able to read all of the different linear bar code symbologies. Therefore one should make attention before buying bar code scanner if it is compatible with various types of symbologies.**

2. Why Use Barcodes?

The industrial use of barcodes can be traced back as far as the 1960s. Some of these early implementations were used to identify rail road cars. Common barcodes started appearing on grocery shelves in the early 1970s as the UPC code to automate the process of identifying grocery items. Today, barcodes are just about everywhere and are used for identification in almost all types of business. **When barcodes are implemented in business processes, procedures can be automated to increase productivity and reduce human error. Barcoding should be used when ever there is a need to accurately identify or track something.**

The primary purpose of a bar code is to identify something by labeling the item with a bar code containing a unique number or character string. Bar codes are typically used with a database application where the data encoded in the bar codes is used as an index to a record in the database that contains more detailed information about the item that is being scanned. For example, when a checkout clerk scans a bar code on a product in a grocery store, the bar code data is fed to a computer that looks up the information in a central database and returns more detailed information about the item that was scanned including possibly a description of the item and a price.

By using bar codes, the grocery store does not need to put a price tag on each item in the store and they can also change the price for a particular item by modifying a single entry in the central database. They can also track how much of a product is currently in stock so that they know when to re-order more of each item as the number of items in stock falls.

Bar codes also provide a quick and error free means for inputting the data into an application running on a computer. By using bar codes, the potential for errors from manual data input is eliminated. Another typical application for bar codes is therefore for inputting data without having to type. For example one can easily encode name or address data in a bar code on an ID badge and then scan the ID badges to input a persons name into a computer program instead of typing the information.

3. Barcode Symbologies and its Usage

Bar codes are like a printed version of the Morse code. Different bar and space patterns are used to represent different characters. Sets of these patterns are grouped together to form a "symbology". There are many types of bar code symbologies each having their own special characteristics and features. Most symbologies were designed to meet the needs of a specific application or industry.

The type of barcode to use for a particular situation depends upon: (1) the implementation (2) the data one need to encode in the barcode and (3) how you wish to print the barcode. Each type of symbology (or barcode type) is a standard that defines the printed symbol and how a device that a barcode scanner reads and decodes the printed symbol.

If an industry standard has been established for implementation of barcoding then the user most likely do not have a choice in selecting barcode symbology. Industry standards are usually established when multiple parties or companies are involved in the id process. It should be noted that the standard is not necessarily the same as the barcode symbology. Barcode standards define how to use the barcode symbology in a particular situation. As mentioned above, the UPC symbology was designed for identifying retail and grocery items, PostNET was designed to encode Zip Codes for the US Postal Service and ISBN is a standard for labeling books and periodicals that uses the EAN-13 symbology. Below, we list here established barcode standards and what they are used for:

Established Standard:	**For Identification of:**	**Barcode Symbology:**
UPC-A or UPC-E	items for sale in the USA and Canada	UPC/EAN
EAN-8 or EAN-13	items for sale worldwide	UPC/EAN
ISBN, ISSN & Booklan	books and periodicals	EAN-13 with UPC/EAN
UCC-128, EAN-128 or SSCC-18	shipping cartons	Code 128
SCC-14	shipping cartons	Interleaved 2 of 5 or Code 128
EAN-14	shipping cartons	Interleaved 2 of 5 or Code 128
SSCC-18	shipping cartons	Code 128
SISAC	serial numbers for serial publications	Code 128
SICI Code	serial numbers for serial publications	Code 128
POSTNET	US mail addresses for the US Post Office	POSTNET
USPS Special Services	US mail return receipts and registered mail	Interleaved 2 of 5 or Code 128
MICR	bank checks	MICR E-13B or CMC-7
LOGMARS	United States Department of Defense standard	Code 39

- **CODE128**: Code 128 is a continuous, multilevel, full ASCII code with three character sets. The barcode selection will always calculate the **mandatory** check character (modulus 103). To encode alpha-numeric UCC/EAN-128, the character set is set to "**0**" for automatic. Then, ASCII 202 or character Ê is entered as the FNC1 before each AI. For example, the UCC number of (8100) 712345 (21) 12WH5678 should be entered as: **Ê**8100712345**Ê**2112WH5678

- **CODE39:** Code 39 is an alphanumeric barcode that can encode numbers, upper case letters, and the following special symbols: _ . * $ / % +. The **font tools** will calculate the optional check character (modulus 43) which is recommended for LOGMARS. LOGMARS (Logistics Applications of Automated Marking and Reading Symbols) is an application of Code 39 is defined by Military Standard MIL-STD-1189B.

- **CODE39EXT**: Extended Code 39 encodes the full 128 character ASCII character set. The **font tools** will calculate the optional check character (modulus 43) which is recommended for LOGMARS.

- **INTERLEAVED25 *or* ITF**: Interleaved 2 of 5 code is a numeric only bar code. The **font tools** will calculate the optional modulus 10 check character.

- **UPCA**: UPC-A is used for marking retail products. Version A encodes a twelve digit number. The first number encoded is the number system character, the next ten digits are the data characters, and the last digit is the check character.

- **UPCE**: The UPC-E code is a compressed barcode which is intended for use on small items. Compression works by squeezing extra zeroes out of the barcode and then automatically re-inserting them at the scanner. Only barcodes containing zeroes are candidates for the UPC-E symbol.

- **EAN8**: EAN-8 is a shortened version of the EAN-13 code. It includes a 2 or 3 digit country code, 4 of 5 data digits (depending on the length of the country code) and a checksum digit.

- **EAN13**: EAN-13 encodes 13 digits: the first two or three are a country code which identify the country in which the manufacturer is registered (not necessarily where the product is actually made). The country code is followed by 9 or 10 data digits (depending on the length of the country code) and a checksum digit.

- **MSI**: The MSI Code is a numeric barcode that has been used primarily in libraries and retail applications such as grocery store shelves. The **font tools** will calculate the modulus 10 check character.

- **CODE11**: Code 11 is a numeric, high density code with one special character "-". The **font tools** will calculate check character. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

- **CODE93**: Code 93 is a more compact version of Code 39. It encodes exactly the same characters as Code 39, but uses 9 barcode elements per character instead of 15.

- **IND25**: Industrial 2 of 5 (also called Code 2 of 5) is a numeric-only barcode that has been in use a long time. Unlike Interleaved 2 of 5, all of the information is encoded in the bars; the spaces are fixed width and are used only to separate the bars. The code is self-checking and does not include a checksum.

- **CODABAR**: Codabar is a discrete, numeric code with special characters (-$:/.+). The **font tools** will calculate the optional modulus 16 check character.

- **POSTNET and PLANET:** Enter the entire number to be encoded without any spaces or dashes. For barcodes to be easily scanned at the US post offices, they should be between 22 and 24 bars per inch. The default X setting of .03CM should produce about 24 bars per inch.

- **UCC128:** This symbology option adds in the FNC1 function and formats the output to Code 128 set C. Use this function for UCC/EAN-128 applications when you have already generated your MOD10 check digit if required and the input data is an even number. Supports UCC-128, EAN-128, SSCC-18 and SCC-14. For example, to encode an SSCC-18 barcode, you would enter 00000123455555555558 as the data input. The FNC1 code is automatically included after the start digit. Use **Code 128 Auto** to encode additional FNC1 codes or data containing text or odd numbers.

UPC-A with Supplemental UPC-E

UPC-A is a 12 digit, numeric symbology used in retail applications. UPC-A symbols consist of 11 data digits and one check digit. The first digit is a number system digit that normally represents the type of product being identified. The following 5 digits are a manufacturers code and the next 5 digits are used to identify a specific product.

When specifying UPC-A messages, you normally specify 11 digits and let your bar code printing software calculate the 12th check digit for you. (**Our Acct 2000 bar code software automatically calculates check digits for you.**)

UPC-E is a smaller seven digit UPC symbology for number system 0. It is often used for small retail items. For UPC-E bar codes, you normally specify 6 digits and let your bar code printing software calculate the seventh check digit for you.

Both UPC-A and UPC-E allow for a supplemental two or five digit number to be appended to the main bar code symbol. This supplemental message was designed for use on publications and periodicals. If you enter a supplemental message, it must consist of either two or five numeric digits. The supplemental is simply a small additional bar code that is added onto the right side of a standard UPC symbol.

Differences between Type A and Type E

UPC-E is also called "zero suppressed UPC" because UPC-E compresses a normal 12 digit UPC-A number into a six digit code by "suppressing" the number system digit, trailing zeros in the manufacturers code and leading zeros in the product identification part of the bar code message. A seventh check digit is encoded into a parity pattern for the six main digits. UPC-E can thus be uncompressed back into a standard UPC-A 12 digit number.

The main difference between a UPC-A symbol and a UPC-E symbol is the size. Below is a UPC-A bar code on the left and the same data encoded as a UPC-E symbol on the right.

UPC-A

These two bar codes are equivalent.

UPC-E

Barcode Accuracy and Misreads:

In a recent study, several different bar code symbologies were tested to determine their accuracy and are listed below. Keep in mind that a well-trained data entry operator will usually make a data entry error once every 300 keystrokes. Therefore, implementing the least accurate symbology is a huge step forward to increase production and reduce data entry errors. The most inaccurate symbology (UPC) is used in the retail industry.

Barcode Type	Worst Case Accuracy	Best Case Accuracy
DataMatrix	1 error in 10.5 million	1 error in 612.9 million
PDF417	1 error in 10.5 million	1 error in 612.4 million
Code 128	1 error in 2.8 million	1 error in 37 million
Code 39	1 error in 1.7 million	1 error in 4.5 million
UPC	1 error in 394 thousand	1 error in 800 thousand

4. Bar Code Printing

Printing 100% compliant barcodes from any printer to any label or document will ensure perfect readability. With good quality label printers with built-in barcode support this is usually possible as long as you carefully follow the manufacturer's quality control directions. But good thermal printers can cost several thousand dollars and are only designed for label printing. What about printing from general purpose printers, such as laser printers? In this case the software used is critical to guarantee perfect barcodes.

The guidelines below are based on the industry standard recommendations of the Uniform Code Council for EAN/UPC symbols but the concepts apply equally well to all barcode symbologies.

As a user it is important to understand these concepts in selecting the best hardware and software for barcode printing.

Thermal Transfer Bar Code Label Printers

These are commonly used printers for barcode labeling. It is most important with this type of printer to make sure that you follow the manufacturer's recommendations for setting up the printer and testing the bar codes. Direct thermal and thermal transfer printers require specific settings for best results depending on the combination of label and ribbon materials. The manufacturer will supply the directions for the correct adjustment for your printer type.

It is also very important to verify bar codes using an ANSI based verifier. This should be done after any change in the label material or any change in the printer or it settings. It should also be done or a regular basis to ensure quality is maintained and when printing a long run.

To maintain the quality of printed bar codes the manufacturer's directions for cleaning the print head and guide surfaces should be followed. It will also be necessary to replace the thermal print head eventually as these wear out. When this happens the bar codes will no longer be readable as one or more of the dot elements will not heat properly. It is so important to verify printed symbols on a regular basis that some thermal printers can be equipped with on-line verifiers.

With thermal printers the quality of the label design software used will not effect the quality of the printed bar codes. This is because the software is just sending a command to print a bar code - a command that triggers the printer's internal software to actually generate and print the correct bar code.

General purpose printers, especially laser printers, are excellent for producing bar code labels on sheets of pre-cut labels (such as "Avery" labels) or on continuous feed labels. Obviously, for anyone needing extremely high quantities of bar codes on a daily basis thermal transfer printers with their high speed would be better, but for many users general purpose printers, especially laser printers, are preferably.

Laser printers are also perfect for creating bar codes on documents, such as medical and legal records, coded "mail-merge" letters, etc. Imagesetters, such as Linotronic printers, are used to create high quality, 2540+ dpi, film positives and negatives for commercial printing.

With general purpose printers, including laser, desk top, ink jet, ion deposition, dot matrix and imagesetter, the software used is critical to ensuring readable symbols.

There is a wide variety of software packages for creating symbols using general purpose printers. **Unfortunately, many of these packages are capable of producing symbols with totally unacceptable quality. So one should keep in mind before implementing software package that should compatible to any types of printers using.**

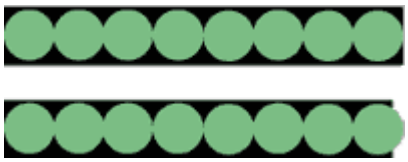
5. UCC Guidelines for Producing Quality Symbols

The software used should be able to satisfy the following requirements:

1. Dots Per Bar Width

When specifying the narrow bar (module) width the user should be able to specify an integer multiple of the printer dot width. This is best explained with an example. For a 600dpi printer the actual dot width is 1/609.6* = 0.00164". The module width should be specified as an integer multiple of printer dots, i.e. 8 dots per module gives a width of 3.12 Mils and a Magnification Factor of 100.95% 7 dots per module gives a width of 11.48 Mils and a Magnification Factor of 88.33%.

(*A 600 dpi printer with an actual 609.6 dpi)



Module (bar) width defined as 8 printer dots or 13.12 Mils or a Magnification Factor of 100.95%

Module (bar) width defined as 13 Mils or a Magnification Factor of 100%. This is not an Integer number of printer dots. Rounding errors could occur when printing.

The easiest way to achieve the correct module width for your printer is to have software that:

i. *Allows you to specify the target printer resolution or finds the resolution for you for the printer you have selected.*

ii. *Then allows you to specify the module width in "printer dots" (it would already know the "dots per inch"). You would select the number of printer dots that most closely matches the Magnification Factor, or module width in Mils, that you desire. If you wanted 100% magnification an "8 printer dot" width for a 600 dpi printer would be the closest.*

2. Bar Width Reduction

The software you select also needs to have an option for "Bar Width Reduction." This is important when printing to any printer that has "dot spread". This includes ink jet printers and "wet ink" printers (but not laser printers). For these printers one dot of bar width reduction is recommended per bar width. For instance for a 300 dpi ink jet printer with a module width of 13.12 Mils (4 printer dots per module width) Bar Width Reduction of one dot would be 3.28 Mils or 25%.

Bar Width Reduction is also required when bar codes are going to be printed on a commercial (wet ink) press. In this case the artwork containing the bar code is usually output on film from an Image setter. Image setters have resolutions of 2400+ dpi so the required bar width reduction can be specified exactly. The actual amount of reduction required is determined by the paper and ink used and this should be specified by the printer. If printing to a smooth hard surface, such as a glass bottle, Bar Width Gain may be required.

3. Bar Code Graphic Type

There are 3 main types of bar code graphics type

1. Bar Code Fonts
2. Bit-map (Raster) Bar Code Generators
3. Vector Bar Code Generators

i) Bar Code Fonts

Although fonts are not normally thought of as graphics, they can be used to produce bar codes. A font is a collection of graphic elements that are assigned to each of the characters in the ASCII or ANSI character set. Because most bar code symbologies encode data by mapping specific characters to specific bar and space patterns, it is possible to use fonts to create bar codes. Unfortunately, fonts have many inherent problems when they are used to create bar codes.

The most important problem with fonts when they are used to create bar codes is that they are not "intelligent". Almost every bar code symbology has features like start and stop patterns, check digits, guard patterns, quiet zones and bearer bars. When you use a font to create a bar code you cannot simply select the text for the message that you want encoded and select a bar code font. You first have to insert special characters for the start and stop patterns as well as manually calculate and insert a special character for the check digit. In almost all cases you have to use a special program provided with the font to calculate and add check digits and insert start and stop patterns. Another problem with fonts is that they cannot be scaled in a single direction in most application programs. Some specialized desktop publishing programs allow stretching of fonts in a single direction however most database or word processing programs do not. This means that if you increase the size of a font, both the height and the width change. When creating bar codes, it is extremely important that the width of the bars and spaces remain constant. Typically the height of a bar code font is not adequate and it must be scaled up. When you do this by increasing the font size, the width of the bars and spaces as well as the overall width of the bar code increases proportionally which causes the bar code to be rendered out of spec. In general fonts are the poorest choice for creating bar codes. They offer the least control of all possible methods for creating bar codes.

"Bar code fonts have been known to create EAN/UPC symbols with serious quality defects. The problems may be caused by the inherent design of the font, operator input, or a combination of both..." Uniform Code Council, Inc., Guidelines for Providers of EAN/UPC Symbol Design Software, 1997, Page 13.

"For these reasons, extreme caution should be used when producing EAN/UPC symbols with bar code fonts. They should only be used by highly experienced bar code design professionals utilizing appropriate controls"

This is true when using fonts to create any bar code symbology.

ii) Bit-map Bar Code Generators

A bitmap is an array of dots or "pixels" where each pixel (picture element) has a value that represents the color of the pixel. The width of a bitmap is defined by the number of pixels across a row. The height of the bitmap is defined by the number of rows of pixels. Any graphic that is made up of rows of dots is called a "Raster" graphic therefore a bitmap is a raster graphic. The overall printed dimensions of a bitmap is dependent on the dot resolution of the device used to print it. For example if you create a bitmap graphic that is 300 pixels wide and has 300 rows of pixels and then you print this bitmap on a printer that has a dot resolution of 300 dots per inch, you will end up with a printed image one inch square. If you display the same bitmap on a computer screen that has a dot resolution of 100 dots per inch, you end up with an image that is three inches square. This means that bitmaps are "device dependent" where the resolution of the rendering device (i.e. printer or screen) must be taken into consideration when you create the image. If you need to preserve the size of an image from one device to another, (i.e. screen to printer) you must "stretch" or "shrink" a bitmap to fit the desired size. The process of stretching and shrinking involves either adding or removing pixels to or from the original image. As you can imagine resizing a bitmap to a desired size when moving from one output device to another generally causes a severe degradation of the original image quality. Therefore, when creating precise graphics (like bar codes) it is extremely important that the image is created with the same dot resolution of the printer. If you do not know the dot resolution of the printer that will be used to print the bar code then you cannot fully guarantee that the image will be readable by all bar code readers.

A type of bitmap called a "Device Independent Bitmap" has been developed as an attempt to solve the device dependency problem however you still end up with distorted images when you render a bitmap on a device that does not have the exact same dot resolution that the bitmap was created with.

Another problem with bitmaps is that they require large amounts of memory. For example a bitmap that is 300 pixels wide and 300 lines down (only 1 square inch on a 300 DPI laser printer) and has three bytes per pixel of color information (standard RGB colors) will require 270,000 bytes of memory or disk space. For bar codes, this is a huge amount of space for a very simple graphic made up of a relatively small number of rectangles

iii) Vector Bar Code Generators (MetaFile)

The absolute best way to create bar codes is to use a vector style graphic. Instead of containing an actual raster style image (like a bitmap), a vector graphic contains a sequence of drawing instructions that describe how to render the image. For example it might contain an instruction that tells the output device to move to a point exactly two inches down and to the right from the upper left corner of the screen or page and draw

solid black rectangle that is exactly a quarter of an inch wide and one inch tall.

The Windows metafile (WMF) and the Encapsulated Postscript (EPS) graphics formats are "vector" graphic formats. The prominent features of the WMF format is that it is completely device independent, it supports extremely precise dimensions for all graphic elements (down to 1/100th of a millimeter) and the amount of memory required to store a metafile is extremely small. Best of all, every printer that has a Windows printer driver must support printing metafiles therefore there is never an issue with being able to print metafiles on a particular printer. The characteristics of metafiles are ideal for creating bar codes.

As an added bonus, most programming languages and commonly used Windows programs provide built in support for handling metafiles. Metafiles are in fact built from the natural "graphics language" used by Windows for creating almost all graphic elements used in every Windows program. Widows also provides built in clipboard support for the metafile format which makes it extremely easy to move them between applications.

The only problems that one might encounter with metafiles is that some service bureaus have trouble printing them on a small number of high resolution film mastering devices. Typically the problems that they encounter are because they are using older, outdated printer drivers. Service bureaus generally prefer the EPS graphics however most Windows programs do not have the ability to render EPS images on screen making EPS files difficult to work with. For example PageMaker for Windows will allow you to insert EPS files into a document however when you do so, PageMaker only displays a rectangle where the image would appear in the document. You will not see the image on your screen unless the EPS file has a TIFF preview embedded in the EPS file. (A TIFF is a type of raster graphic similar to a bitmap. Both EPS and WMF files can contain raster images embedded as part of the vector graphic.)

Imaging Device Resolution in DPI*	
Bit-map Images	100 - 400 dpi (device dependent)
Vector Images	100 - 4000+ dpi (device independent)
Thermal Bar Code Printers	100 - 400 dpi (with resident bar code formats)
General Purpose Printers	300 - 1200 dpi (laser, inkjet, etc..)
Image setters	1200 - 4000+ dpi (i.e. Linotronic printers)

**Taken from Guidelines for Providers of EAN/UPC Symbol Design Software*

6. Bar Code Verification

As with Thermal Transfer printers, it is important to test the output from General Purpose printers using a bar code verifier. The UCC recommends using the test symbols:

0 12345 01234 1

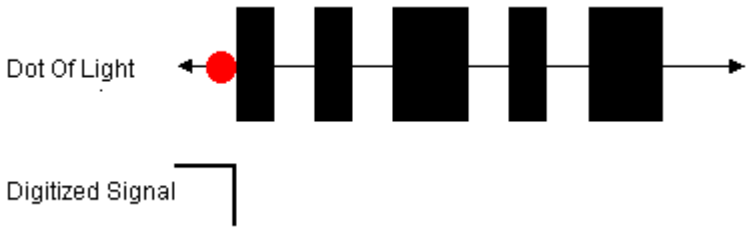
6 78912 56789 0

If they fall below a grade B check that you have specified a Magnification Factor (or module width) that corresponds to an integer number of printer dots per module (bar) width. Bar Width Reduction may also be required as discussed above.

If the user does not have a bar code verifier he can submit the bar codes to a qualified testing organization. If nothing else, at least they should be tested with a bar code reader.

7. How a Bar Code Reader Works

Bar Code readers use an array of hundreds of tiny light sensors lined up in a row in the head of the reader. Each sensor can be thought of as a single photo diode that measures the intensity of the light immediately in front of it. Each individual light sensor in the Bar code reader is extremely small and because there are hundreds of sensors lined up in a row, a voltage pattern identical to the pattern in a bar code is generated in the reader by sequentially measuring the voltages across each sensor in the row.



8. Interfacing PC with our Acct 2000 Bar Coding System

All application programs support bar code reading as long as you have the right equipment. Bar code readers are available with two types of output - either "keyboard wedge" output or RS232 output. The bar code readers with keyboard wedge output plug directly into the keyboard port on your PC and they also provide a pigtail connector so that you can plug in your keyboard at the same time. **When you scan a bar code with**

the keyboard wedge bar code reader, the data goes into the computer just as if it were typed in on the keyboard. This makes it extremely easy to interface the bar code reader to any application that is written to accept keyboard data.

The keyboard wedge interface is extremely simple however it has a few drawbacks. If you swipe a bar code, the cursor has to be in the correct input field in the correct application otherwise you end up reading bar code data into whatever application has the focus. This can cause all sorts of potential problems as you can imagine. The keyboard output also is limited in that you cannot modify the data in any way before sending it into the program that is to receive the data. For example, if you needed to parse a bar code message into multiple pieces or remove some of a bar code message or add in a date or time stamp you would not be able to with a normal keyboard wedge reader.

The other possible output option is to get a bar code reader with an RS232 or "Serial" interface. With these types of bar code readers, you connect the reader to an available serial port on the back of your PC. You would then need a program called a "Software Wedge" to take the data from the bar code reader and feed it to the application where you want the data to go. The disadvantage to this approach is that it is a little more complex however you gain much more control over how and where your data ends up when you read a bar code.

Our Acct 2000 Bar coding system product line is designed just for this purpose. The system is an executable program that can pass serial data back and forth to other programs using either DDE (Dynamic Data Exchange) or by converting incoming serial data to keystrokes (i.e. it stuffs the keyboard buffer with the incoming serial data). With this system, you can control exactly where the data goes in the target application and you can also perform all sorts of modifications on the data before it is sent to the application including parsing or translating the data as well as adding additional keystrokes or date and time stamps to the data.

Acct 2000 Bar Coding System is extremely easy to use and is designed to have you up and running sending and receiving serial data directly from within your application in just a second. Because the system can pass data using DDE, you can set your application up to insure that the bar code data always goes where it is supposed to go and you can also have your application running in the background and still accept bar code input while you run some other program in the foreground. This system is without question the most robust way to interface a bar code reader to a PC with the least amount of effort.

9. Recommended Bar Code Scanner

The barcode scanners we recommend and sell all have built-in decoders and can read several different bar code types. We have seen some low priced scanners on the market, but they require complicated decoders. After you order and program the decoder, you will end up spending more than if you ordered one with a built-in decoder. Also most barcode scanners cannot read common linear symbologies such as code 39, UPC, EAN, Code 128 and Codabar and some scanner manufacturers usually ship new scanners with most symbologies disabled, so one should make sure of all those matters before buying any bar code scanner.

The scanners we sell is of Birch company which is designed and manufactured in Taiwan having used world wide. It receives their power from the PC keyboard or USB port so no external power supply is required. When a barcode is scanned in, the data is sent to your PC as if it had been typed by the keyboard. Besides they supports all the industry standards of Bar Code Symbologies. Some of the general features of our providing scanner are as below:

BS-960L Laser Gun

General Features

- 4-in-1 decoder built-in
- Adjustable laser head angle
- Professional performance including data formatting,
- lower power consumption.
- Discrimating most barcode symbologies.

Output Interface

- Keyboard, RS-232, Wand Emulation, OCIA or other for request.

Laser module

- Metrologic (BS-960L)

Readable barcode

■ EAN-8, EAN-13	■ CODE 39	■ CODE 32
■ UPC-A,UPC-E	■ Industrial-25	■ Interleaved-25
■ Codabar/NW7	■ MSI/Plessey	■ Code 11
■ Code 128	■ Telepen	■ IATA
■ China Post 25	■ S-Code	■ Matrix-25

Depth of Field

- 300mm (BS-960), 200mm (BS-960L)

Scanning Angle

- 0~33 degrees (5 steps)

Resolution

- Max 0.127mm (5 mils)

Scanning Rate

- 100scan/second

Light source

- Laser diode (under 5MW)

Read Confirmation

- Red LED and adjustable Volumn Buzzer

Host & Terminals

- IBM XT/AT Compatible,IBM PS/2 25/30/55,IBM 5550/5576,Mac ADB, NEC 98xx, OCIA and WYSE Terminal

Connector

- Dsub-9p/ Dsub-25p/ Din-5p/ Din-6p/ Din-8p/ Mini Din-4p/ Mini Din-6p/ Other for request

Weight

- 300g/Decoder Built-in

Operating Temp.

- 0~40 degree C, Humidity 10~90%RH

Storage Temp.

- 10~60 degree C, Humidity 5~90%RH

Voltage

- DC 5V +-5%

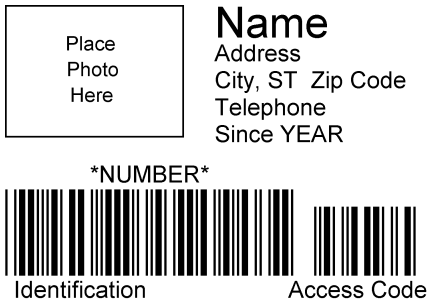
Power Consumption

- Operating (Max.100mA), Stand-by (Max.20-50mA), Power safe mode (<100uA)

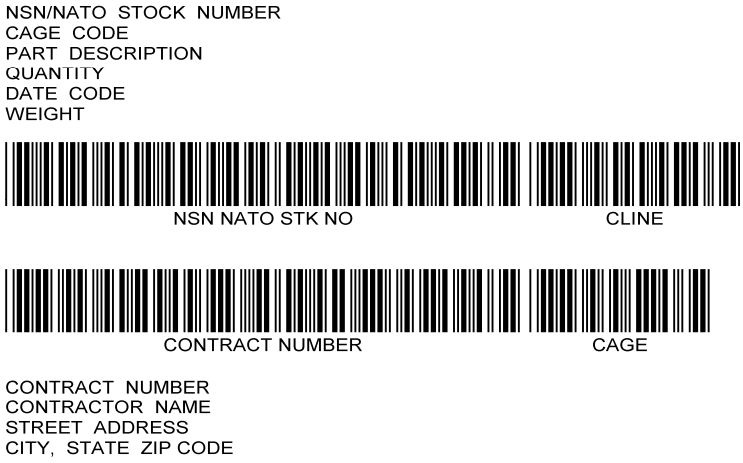
Bar Code Samples and its usage in various applications

BarCoding used in Books and Periodicals (ISBN)

Barcoding used in ID Card



Bar Code used in cargo items

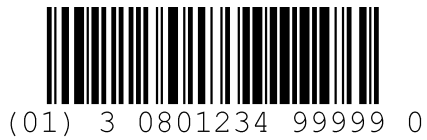


Bar Code used in General items and their symbologies

UPC A

UPC E

EAN



UPC EAN 128