Abstract

What is a Bar Code?

Bar Code Structure

Types of Bar Codes

Bar Code Readers

Serial Bar Code Readers

Mainframe Bar Code Readers

Portable Readers

Radio Frequency Readers

Picking

Put-Aways

Receiving

Shipping

Programming for the RF Terminal

One-Way communication

Two-way Communication

Before you begin programming

Host to Terminal Programming

Sign on

Sign Out

Addressing a terminal not Signed on

Programming Example

Wand Scanners
Abstract

The use of bar coding has been growing dramatically over the last 15 years. With the adoption of UPC as the standard for retail grocery stores in the late 70’s, bar codes are a fast, easy, and accurate data entry method. The correct use of bar codes decrease time required of the employee and increases an organization’s efficiency. One thing to remember with bar codes is that the application software, which accepts the bar code data is in full control of the success or failure of an application. In other words, the bar codes are another way to record data. With the introduction of the IBM PC in the early 80’s bar coding applications expanded along with the pc explosion. Worth Data was and is a pioneer in providing bar code hardware and printing software to the PC and Mac user. Understanding the purpose of bar codes and how they are used helps computer users to see how important this technology is beneficial in recording data.

What is a Bar Code?

The data in a bar code is a reference number the computer uses to look up associated records. These records contain descriptive data and other pertinent information. For example, the bar codes found on food items at grocery stores do not contain the price or description of the food item. Instead, the bar code has a product number, which contains 12 digits. When the bar code data is
transmitted to the computer, the computer finds the database record associated with that item number. In the disk file is the price, vendor name, quantity on hand, and the description. The computer does a price lookup by reading the bar code and then it creates a register of the items. At the end, the price is either added to the subtotal of the purchased item or the price is subtracted from the total on hand. Another example of bar coding data is the quality of reporting application, which is when the bar code has only a single digit in it, but it may be titled “Failed Vibration Test”. The computer associates the single digit with the test result. Typically, bar codes have only ID data in them. The ID is used by the computer to look up all pertinent detailed data associated with the ID data. Information that is typically not bar coded is social security numbers, license plate numbers, names, and addresses.

**Bar Code Structure**

A bar code is a series of varying width vertical lines, called bars and spaces. Bars and spaces together are named elements. There are different combinations of the bars and spaces which represent different characters. When a bar code scanner is passed over the bar code, the light source from the scanner is absorbed by the dark bars and not reflected, but it is reflected by the light spaces. A photocell detector in the scanner receives the reflected light and converts the light into an electrical signal. As the wand is passed over the bar code, the scanner creates a low electrical signal for the spaces, reflected light, and a high electrical signal for the bars. The duration of the electrical signal determines wide vs. narrow elements. This signal can be decoded by the bar code reader’s decoder into the characters that the bar code represents. The decoded data is then passed to the computer in a traditional data format.

**Types of Bar Codes**

There are many different types of bar codes. The bar codes are categorized into three subcategories. First category, bar codes are only numeric, UPC, EAN, and Interleaved 2 of 5. Second category, bar codes have a fixed length, such as UPC-A with 12 widths, UPC-E with 6 digits, EAN-13 with 6 digits, and EAN-8 with 8 digits. Third category, bar codes have numbers and alphabetic characters, such as Code 93, Code 128, and Code 39. Code 128 allows you to encode all 128 characters. Many bar codes were invented some time ago and have been superseded by newer bar codes. Some industries standardize the older bar codes before the more efficient bar codes. Therefore,
there is a continuing requirement for their use in particular industries. The following are a couple of bar codes.

![Bar Codes](https://www.cs.iusb.edu/internship/papers/kharris/bar_code.htm)

(Atoms 1)

**Bar Code Readers**

There are three basic types of bar code readers: fixed, portable batch, and portable RF. Fixed readers remain attached to their host computer and terminal and transmit one data item at a time as the data is scanned. Portable batch readers are battery operated and store data into memory for later batch transfer to a host computer. Some advanced portable readers can operate in non-portable mode too, often eliminating the need for a separate fixed reader. Portable RF readers are battery operated and transmit data real-time, on line. More importantly the real-time two-way communication allows the host to instruct the operator what to do next based on what just happened. A basic bar code reader consists of a decoder and a scanner. Also, a cable is also required to interface the decoder to the computer or terminal. The basic operation of a scanner is to scan a bar code symbol and provide an electrical output that corresponds to the bars and spaces of a bar code. A decoder is usually a separate box which takes the digitized bar space patterns, decodes them to the correct data, and transmits the data to the
computer over wires or wireless. This transmission can be done immediately or on a batch process.

**Serial Bar Code Readers**

Another method of data transmission from the bar code reader to the computer is by RS-232 Serial ASCII format. For example, the Unix system can be used as a multi-user computer, with serial ASCII terminals for each user. A bar code reader can be attached between the terminal and host computer transmitting ASCII data just like the terminal. In fact, the car code data looks just like keyed data. Serial readers can be places several hundred feet from the computer where as keyboard wedge readers cannot be placed beyond 10 feet. Also, multiple serial readers can be attached to the same computer. The PC runs a program to poll the readers one at a time.

**Mainframe Bar Code Readers**

Mainframe computers often have terminals with unique data connectors and data formats, which are different from ASCII or PC key codes. The IBM system 36-38, AS/400, 4300 and 9000, have these types of terminals. To use bar codes with these computer systems, a keyboard wedge reader is used. They are specifically designed for the terminal to be attached to. The alternative is to have a PC with a terminal emulation card in it attached to the mainframe. Then a less expensive PC bar code reader and laser printer can be used on the PC.

**Portable Readers**

Portable readers are handheld battery operated readers which store the data in memory for later uploading. In addition to a bar code scanner, a portable reader usually has an LCD display to prompt the user what to do. Also, they usually have a keyboard to enter variable data such as quantities. Ease of programmability is a key issue in the selection of a portable. Many vendors believe that it is easy as long as the computer user can program in C++. Other variables to consider are battery life, ease of reading the display, size and weight of the unit, who repairs it, and where it is to be repaired in the event of a malfunction. Worth Data has pioneered and patented voice prompt messages to supplement the display messages in a portable unit. This unit actually charges the batteries or uploads data, and can customize any or all voice prompts for your applications.
Radio Frequency Readers

Radio Frequency Readers are the ultimate solution to many applications needs, such as any computer remote application that can benefit from the computer checking and instruction of the operator. Warehousing applications such as picking put-aways, shipping and receiving are typically better performed by RF readers because the computer can instruct the operator where to go and what to do. Plus, the computer files are current as to exact status and location of available inventory. RF Readers are like on-line terminals, but wireless. The user can roam around his local facility scanning, keying data, and getting a response from the computer with each entry. Therefore the computer edits the data for errors as well as prompts the user for what to do next considering the data that has just been entered. The classic RF applications and associated advantages are: Picking, put-aways, receiving, and shipping.

**Picking**

The first RF application to discuss is picking. Picking is the routing of the picker, computer instructed substitutions, real-time status of the order.

**Put-Aways**

The second RF application is put-aways, which is when inventory is available for sale or for manufacturing immediately.

**Receiving**

The third RF application is receiving, which is where purchase order shortages can be immediately determined. Critical parts can be routed to manufacturing immediately. Here is an example, in a real world situation. At Reese Products there are four distribution centers. One of these centers is located in Corona California. An order has been place at the Corona cite which cannot be filled with the stock on hand. The RF application would determine this and immediately send a message to the manufacturing division requesting more inventory to fill the order. Manufacturing would then build the parts and have them shipped to the Corona cite to complete the order.

**Shipping**
The final RF application is shipping, which is eliminating wrong or incomplete shipments by computer checking before loading.

**Programming for the RF Terminal**

The RF Terminal has a 6X24 LCD screen and up to 99 voice messages which the host user program can activate. Messages from the host user program are written to the serial port to which the applicable Base Station is attached. Up to 16 RF Terminals can be controlled by one base station, so the host user program must address the applicable RF Terminal by its ID character. When the host receives a message from the Base Station it will receive data with the Terminal ID that is also included. Programs can be written in any language that has access to the serial port, reading/writing, regardless of the platform. No more than one Base Station is allowed for each serial port. Multi-dropping or daisy-chaining multiple Base Stations off a single COM port is not supported.

The RF Terminal operates in two basic ways, one-way communication and two-way communication.

**One-Way communication**

In the RF Terminal, One-Way communication initiates where all data is transferred. The Base Station itself simply acknowledges the receipt of the data by echoing it back to the terminal. The host computer has no dialog whatsoever with the Base Station or Terminal. The host computer’s main function is to simply do something with the data coming from the Base Station through the serial port.

**Two-way Communication**

Two-way communication entails messages from the host user program being sent to the Base Station via the serial port; and then from the Base Station to the RF terminal. The terminal responds back to the Base Station with data and its terminal ID. The data is then transmitted from the base to the host computer where it is processed and the next command is sent out. Each RF terminal has a unique terminal ID, allowing a single base station to handle up to 16 terminals. Two-way dialog is established when a terminal signs on to the RF network. The host computer application waits until a terminal signs on, and then it begins processing by sending the first prompt out to the terminal via the Base Station. If the terminal does not receive a prompt from the host, it goes into sleep mode, and it wakes up and
checks with the Base Station periodically to see if it has any messages waiting. This conserves battery power and reduces radio traffic. The difference between two-way modes is that it requires programming to communicate with the terminal and one-way mode does not.

**Before you begin programming**

Before you begin programming, there are some factors you should take into consideration during the planning process.

1.) Plan for system failures, this includes hardware, software, and operator failures. In order to create an efficient application, you must put some thought into what you will do when different parts of the system fail.

2.) Plan for expansion. You may start small but try to create an application that will allow for easy expansion and addition, especially of terminals.

3.) Plan for a site evaluation. Site testing does not require that you have an application up and running, and it can save you time when you do sit down to create your program if you already know what you will be dealing with in terms of Base Stations and relays.

4.) Use the demo program. The demo programs can at least allow you to see how the system functions and whether you can anticipate any system wide problems. The demo programs should also be used as a response time benchmark.

**Host to Terminal Programming**

Nick Johnson states, the basic format of a message that is transmitted from host to base to terminal is fairly simple:

<table>
<thead>
<tr>
<th>Byte position</th>
<th>Function</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RF Terminal ID</td>
<td>0-F</td>
</tr>
<tr>
<td>2+</td>
<td>Command(s)</td>
<td>**</td>
</tr>
<tr>
<td>Last</td>
<td>Termination of message</td>
<td>EOT (ASCII 4)</td>
</tr>
</tbody>
</table>

The RF terminal ID is always the first byte and is always one character in length. The 16 different possible values are the numbers 0-9 and the letters A-F. The command section of the message always starts with the second byte and can consist of one or more commands including data to be displayed or voice message to be broadcasted.

Here is a listing of examples of valid commands:
<table>
<thead>
<tr>
<th>Command Characters</th>
<th>Command Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>*@</td>
<td>Re-initializes all terminals</td>
</tr>
<tr>
<td>3@</td>
<td>Re-initializes Terminal #3</td>
</tr>
<tr>
<td>1@</td>
<td>Bn Make Terminal #1 beep n (1-9) times</td>
</tr>
<tr>
<td>2@C0*</td>
<td>Clears the entire screen (4 lines or 6 lines) on Terminal #2.</td>
</tr>
<tr>
<td>0@C1</td>
<td>Clears line 1 on Terminal #0</td>
</tr>
<tr>
<td>1@C2</td>
<td>Clears line 2 on Terminal #1</td>
</tr>
<tr>
<td>2@C3</td>
<td>Clears line 3 on Terminal #2</td>
</tr>
<tr>
<td>0@C4</td>
<td>Clears line 4 on Terminal #0</td>
</tr>
<tr>
<td>3@C5*</td>
<td>Clears line 5 on Terminal#3 (if 6 line display), Clears all lines if 4 line display.</td>
</tr>
<tr>
<td>1@C6*</td>
<td>Clears line 6 on a 6 line display. Will do nothing on a 4 line display.</td>
</tr>
<tr>
<td>1@Dn</td>
<td>Displays date and time on line n (1-4) in US (mm/dd/yy, hh:mm:ss) or Euro (dd/mm/yy, hh:mm:ss) format on Terminal #1</td>
</tr>
<tr>
<td>1@Vnn</td>
<td>Play voice message #nn (01-99) on Terminal #1</td>
</tr>
<tr>
<td>1@Sdataxxxx</td>
<td>Output dataxxxxxxxxx to serial port on Terminal #1 -max 255 chs</td>
</tr>
</tbody>
</table>

A typical "prompt" command sequence follows the format below:

**0@n,m,o,data**

<table>
<thead>
<tr>
<th>Where:</th>
<th>n</th>
<th>m</th>
<th>o</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>is the line number (1-4) you want the prompt displayed on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>is the character position (1-20) where you want the prompt displayed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o</td>
<td>is the character that determines whether the prompt is for display only (0) or is waiting for data input (1) See the table below for valid characters for this position.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data</td>
<td>is the data you want displayed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example, the command *@1,1,1, Enter Quantity* would display “Enter Quantity” starting at position 1 on line 1, and then it waits for the operator to enter their data.

These are valid entries for the third position character:
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No data input for this Command, Display ONLY</td>
</tr>
<tr>
<td>1</td>
<td>Data input required from the keypad or scanner</td>
</tr>
<tr>
<td>2</td>
<td>Only keypad input allowed, start un-shifted</td>
</tr>
<tr>
<td>3</td>
<td>Only keypad input allowed, start SHIFTED</td>
</tr>
<tr>
<td>4</td>
<td>Only scanner input allowed</td>
</tr>
<tr>
<td>5</td>
<td>Only accept YES (Enter key) or NO (0 key) keypad response. (Terminal sends 1 for YES, 0 for NO)</td>
</tr>
<tr>
<td>A</td>
<td>same as 1, but time stamped as prefix (hhmmss)</td>
</tr>
<tr>
<td>B</td>
<td>same as 2, but time stamped as prefix (hhmmss)</td>
</tr>
<tr>
<td>C</td>
<td>same as 3, but time stamped</td>
</tr>
<tr>
<td>D</td>
<td>same as 4, but time stamped</td>
</tr>
<tr>
<td>S</td>
<td>SHIFTED keypad input or scanner input</td>
</tr>
<tr>
<td>P</td>
<td>un-shifted keypad entry with no display (for passwords)</td>
</tr>
<tr>
<td>P</td>
<td>SHIFTED keypad entry with no display (for passwords)</td>
</tr>
</tbody>
</table>

Here are some rules for creating commands: Re-initialize commands * @ or x @ clear the buffer for terminals in the Base Station. Following a re-initialization, the host program should re-display all the screen data necessary to start the application. A message with multiple commands is legal and useful. For example, the command “@ 1,1,0,PLEAS2 ENTER @ 2,1,1,QTY” would display PLEAS2 ENTER on line 1, display QTY on line 2, then wait for data input. All 4 lines can be filled with one message. Prompt commands request data entry that can only be combined with clear statements @Cx and data display statements. Multiple data entry prompt commands cannot be combined. For example @1,1,ITEM@2, 1,1QTY is an illegal statement, it has two data entry prompt commands combined. If this command was sent to the RF terminal, it would be ignored as a command and displayed on the terminal display exactly as written. Every statement must end with a data entry prompt command, whether the statement is a single command by itself or several commands.
combined together. Any illegal statement will be ignored as a command but will be displayed on the addressed RF terminal display exactly as written. If Terminal ID was not included in the statement, it will try to display the invalid statement on ID 0. Once the ENTER key is pressed on the terminal displaying the invalid statement, the terminal sends the base station a “?” character. The base station then in turn sends the message n?CR, where n is the terminal ID and CR is a carriage return, back to the host computer. (Johnson)

**Sign on**

To login to the host computer, the user presses a key on the RF terminal at power-up to get to the sign on screen. As the user signs on, the base station sends back the following sign on message to the host.

<table>
<thead>
<tr>
<th>Byte position</th>
<th>Function</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RF Terminal ID</td>
<td>0-F</td>
</tr>
<tr>
<td>2+</td>
<td>SIGN ON</td>
<td>SI (ASCII 15) if 4 line display terminal or; if 6 line display terminal configured as 4 line display in firmware version 9075 or greater.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYN (ASCII 22) if 6 line display configured as 6 line display in firmware version 9075 or greater</td>
</tr>
<tr>
<td>Last</td>
<td>Termination of message</td>
<td>CR (ASCII 13)</td>
</tr>
</tbody>
</table>

(Adams)

After a terminal signs on, the host should be prepared to acknowledge the sign on and give the terminal instruction such as, Standby for assignment, press enter to acknowledge or nothing to do, press enter and see supervisor or pick item 1234. If there is something for the terminal to do, the host should send instruction to the terminal. If there is nothing to do at the time of the sign on, the host should acknowledge the sign on and tell the terminal to stand by or see supervisor. The user will notice that in lines 1 and 2 there is a request for the operator to press the enter key. This is required for the message to be a valid command. All messages must end with a request for data input. Then the host
would expect a response from the terminal of terminal ID and CR.

**Sign Out**

When a RF terminal is powered down manually or the user presses the F1 key to exit data entry mode to go to one of the other modes, the host receives the following sign out message.

<table>
<thead>
<tr>
<th>Byte position</th>
<th>Function</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RF Terminal ID</td>
<td>0-F</td>
</tr>
<tr>
<td>2+</td>
<td>SIGN OUT</td>
<td>SO (ASCII 14)</td>
</tr>
<tr>
<td>Last</td>
<td>Termination of message</td>
<td>CR (ASCII 13)</td>
</tr>
</tbody>
</table>

(Adams)

**Addressing a terminal not Signed on**

If the host attempts to send a message to a terminal that is not signed on, the base station sends back the following message to the host computer.

<table>
<thead>
<tr>
<th>Byte position</th>
<th>Function</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RF Terminal ID</td>
<td>0-F</td>
</tr>
<tr>
<td>2+</td>
<td>Terminal NOT Signed In</td>
<td>DC1 (ASCII 17)</td>
</tr>
<tr>
<td>Last</td>
<td>Termination of message</td>
<td>CR (ASCII 13)</td>
</tr>
</tbody>
</table>

(Adams)

Changing the base station setup can change the ASCII 17 character to ASCII 16 for XON/XOFF sensitive systems.

This program is a demo, which is a simple program to control a single RF terminal in a simulated picking application. It does not have any call for any voice messages to be broadcasted because it can run without voice messages needed to be recorded. Lines 56 and 57 are replacement lines for 40 and 50 if you want to add voice to the demo making it very powerful.
Programming Example

Nick Johnson, author of "Bar Code Programming" states, this is a very simple DEMO.BAS program, which asks for item and quantity.

10 DIM P$(10)
20 "COM1:9600,N,8,1,RS,DS,CS,CD" AS #1
30 P$(1)="@B1@C5@1,7,0,PICKING"
40 P$(2)="@C3@@3,2,1,ITEM"
50 P$(3)="@C4@4,2,1,QUANTITY"

' to add voice, replace lines 40 and 50 with the two lines below:
56 'P$(2)="@C3@V01@3,2,1,ITEM"
57 'P$(3)="@C4@V02@4,2,1,QUANTITY"

60 B$="":T1$=""
70 IF EOF(1) THEN 7
80 A$=INPUT$(1,#1)
90 IF RIGHT$(A$,1)=CHR$(13) THEN 110
100 B$=B$+A$:GOTO 70
110 IF RIGHT$(B$,1)=CHR$(19) THEN 180
120 IF RIGHT$(B$,1)=CHR$(14) THEN 210
130 IF RIGHT$(B$,1)=CHR$(15) THEN T1$=LEFT$(B$,1)+P$(1)+P$(2):GOTO 150
140 T1$=LEFT$(B$,1)+P$(1)
150 $T$ = $T_1$ + CHR$(4)$  

'add EOT for terminator 

160 PRINT #1, $T$;  

'send message to Terminal 

170 I = I + 1: IF I = 4 THEN I = 2  

'handle indexes 

180 PRINT "TRICODER/BASE DATA : "; $B$  

'display data received 

190 PRINT "HOST RESPONSE : "; $T_1$  

'display data transmitted 

200 GOTO 60  

'wait for another message 

PRINT 

210 "TRICODER"; LEFT$(B$, 1); "SIGNED OUT"  

(Johnson 1) 

Wand Scanners 

Wand Scanners are the lease expensive and the oldest type of bar code scanner. A wand is typically made from .5” stainless steel tubing or from plastic optics within the front of a cord to the back. The wand scanner must be moved by the user’s hand across and in contact with the bar code. While the wand is moving across the bar code, the reflected light is converted to electrical signals through a photocell in the wand. 

Wand Resolutions 

Wands come in a variety of resolutions, such as, low medium and high resolution. The resolutions allow for reading bar codes printed by different methods and for reading very small element widths. Low-resolution wands have a large diameter aperture for the reflected light to pass through to the photocell. Therefore, if there is a void in a dot matrix printed bar, the bar is still interpreted as a bar. A high-resolution wand has a smaller diameter aperture and it sees the same void in a bar as a space. Thus, it cannot read dot matrix code as well as lower resolution wands. A low-resolution wand will have an aperture opening so large that it will view very narrow bars and spaces at the same time. As a result the low-resolution wand is unable to decode a high-density bar code. A
high-resolution wand will see only one bar or space element at a time, which allows it to decode a high-density bar code correctly.

**Scanner light source**

Wands also vary with the type of light used. Today, most wands have visible red light as the light source reflecting from the spaces and bars. If you can see the light being emitted from a visible wand’s tip, it is red. Visible light can read any visible bar code. It can read thermal printed bar codes, whereas wands with infrared light cannot read thermal printed bar codes. The light emitted from an infrared wand’s tip is not visible. One advantage of infrared light wands is reading bar codes that cannot be photocopied. A bar code can be printed with infrared absorbing ink and covered with a black laminate window. The window is infrared blind, giving you a security bar code for use on badge that you do not want to be photocopied. A black on black bar code requires infrared scanners to be read.

**Switch Scanners**

With the typical wand on most bar code readers, power is always applied to the wand unless power is turned off at the computer or terminal. This problem should be of no concern to the user, because the led will last 20 years. However, with a portable bar code reader, the battery power is conserved. A wand with a switch is the best solution since no power is lost as long as the switch is open or off. When the user wants to read, the switch is depressed to provide power to the wand. When finished reading, the switch is released turning off power.

**Printing Bar Codes**

There are several methods of getting printed bar codes: three of these printed bar codes are 1.) Photocomposed bar codes from a label manufacturer. 2.) Printing the bar codes with inexpensive labeling software. 3.) Printing the bar codes on a specialized bar code label printer.

**Pre printed labels**

Photocomposed labels are usually very high quality and you can buy 5000 for around $300. Typically, libraries use pre-printed labels because the labels need to last for 25 years and the volume is usually 100,000 per library. High quality, durable, laminated photocomposed labels are usually used.
Printing on PC Printers

With the proper PC software, today’s dot matrix, inkjet and laser printers are capable of printing excellent quality bar codes. Dot matrix printer printers cannot print high-density bar codes, but laser printers can. Laser printers actually print the best quality bar codes of any commonly available printing technology.

Ink jet printers

These printers are getting better at printing bar codes for all densities but high. They print pages of labels, which can be referred back to the page label stock. If printed labels are going to be exposed to water, do not use the inkjet printers, because the ink is water-soluble.

Laser Printing

Laser printers can produce bar codes with outstanding quality. The quality is consistent, even when the toner gets low. Labels are sectionalized on an 8.5 “ X 11” page in multiple columns and/or rows. Since laser printers feed one sheet at a time it is impractical to print one label at a time. There is an unprintable area to the left, right, top and bottom of any form. This makes full labels impossible unless you sacrifice the top row and maybe the bottom row of labels. Since labels are changing continuously and rapidly, do not expect laser printers to be able to keep up with the needs of bar code labeling. For this, we must look at a special type of printer designed to print bar codes.

Thermal Transfer Printing

Thermal transfer printers are used most frequently. These printers print one label at a time or when it is needed they can print a roll of labels so that the labels can be applied by applicators directly to the boxes. These types of printers are fast and produce top quality bar codes. Thermal transfer refers to the print head heating up and melting a ribbon onto the label surface.

Conclusion

In conclusion, bar codes are all around us. They are becoming such a standard in our lives that it is really important to understand what exactly is contained in the bars and spaces of a bar code. Bar codes are a fast, easy and accurate data entry method. The correct use of bar codes can decrease
employee time required and increase an organization’s efficiency.

Works Cited

Atoms, Russ “Introduction to bar codes”
URL: HTTP://www.azalea.com/Faq/FAQ.pdf

Doleys, John Director of Information Systems at Reese Products.
Personal Interview. 12 Dec. 2000

Downing, Sarah Head of bar codes at Reese Products.
Personal Interview. 12 Dec. 2000

Johnson, Nick “Bar Code Programming”
URL: HTTP://www.spatula.net/proc/barcode/index.src