Bar Codes in Factory Data Collection

Terry W. Simpkins
Spectra-Physics
Retail Systems Division
959 Terry Street
Eugene, OR 97402

Over the last couple of years, much has been written about the impending bar code revolution in U.S. business. While bar codes are being used extensively in retail applications (primarily grocery stores), they have been relatively slow to catch on in manufacturing circles. There are several reasons why acceptance has been slow, including a perceived difficulty in implementation. In Hewlett-Packard 3000 shops, difficulty is not a legitimate excuse anymore. This paper describes one implementation of factory bar-code use, the goals of the project, and the methods used to meet those goals.

Our primary goal was to capture data throughout the assembly process of a new product. The data we wished to capture included: unit serial number, several component part serial numbers, results of various tests, and actions taken to resolve problems and test failures. The constraints placed on the process included: 100% accuracy in data entry, quick data entry, and response time that would not impede the assembly process with volumes up to 300 units per day. Once test data was collected on a significant population of units, we would generate SPC charts on the various production processes.

The general layout of the production process is outlined in figure #1. As you can see, it is a linear process with a vary well-defined workflow. This is a major contributor to the simplicity of the process. The assembly method required data entry at nine positions along the production line and data inquiry was needed at several other locations.

Two of the data entry stations presented a special challenge because they are dedicated PCs performing automated testing on the units. These PCs made physical connections to the units, performed functional and specification compliance tests, graded performance, passed or rejected the unit, and recorded the data collected and decisions made. If the units are rejected the data collected is examined at the network station to facilitate troubleshooting and determine appropriate repair of the units. This means a multi-user/multi-tasking environment is required. The software on the PC required to perform these tests would be written in-house.

Early on in the needs definition, it was decided to continue using an existing PC-based package to generate the required SPC graphs. This meant we needed the ability to

extract data from whatever system was used to collect and store the data and transfer it to a PC via ASCII file. Four alternatives were identified and evaluated:

1) Use the shop-floor control and quality modules of our integrated manufacturing package. This did not provide the functionality or response time guarantee required. Additionally, it was determined the coding required to interface with the dedicated PC test stations would be too difficult to maintain through multiple releases of the vendor's package. Also, this option would require the new software to run on our current Series 70; however, current workload on the 70 would not allow us to guarantee the required response time.

2) Develop a PC/LAN-based system designed specifically to accommodate our needs. This alternative was rejected due to the lack of in-house MS-DOS expertise and the lack of a true multi-tasking, multi-user database solution for the PCs.

3) Purchase a third-party solution. No existing solution was identified that would meet our specifications without substantial modification. Additionally, all packages we found were written in a language we didn't know, or they were priced at an unacceptably high level (i.e., over $30,000).

4) Create our own "home-grown" system designed to provide the needed functionality, and no more. This is the approach we adopted.

We chose to base our system on the HP3000 since that is where our expertise base. The assumption being slightly more expensive hardware and software would be offset by the reduction in training and development time, as well as the cost of on-going support.

We next decided on the Powerhouse product being our language since it is the fourth GL we own and use. These two decisions allowed us to begin development with no delay. The system requirements were broken down into several modules:

1. Basic data collection
   - Part Numbers and Serial Numbers
   - Manual Tests
   - Rework Activity
2. Automated Tests
3. Data Inquiry/Look Ups
4. Label Printing

By using Powerhouse, we were able to create prototype screens for data collection, manual tests, and rework very quickly. These prototypes allowed us to fine-tune our design as we went and thereby avoid significant "mid-course" corrections and massive rewrites.

The data communications requirements of talking to the PCs were more complex. We used COBOL for the programs that would accept data from the PCs and update the host database.

This allowed easy access to all intrinsics and again required very little learning time (for an in-depth discussion on programmatic terminal access, see the "Data Comm" column in the February and April 1985 issues of Interact).

For creation of bar code menus we chose the HP laserjet and the ASK Bar-Scan product. Bar-Scan allows easy creation of bar coded menus and documents. Initially, we used Quiz to create bar-codes on an HP2563 printer for testing. The decision to switch to a stock piece of third-party software was made to reduce development and support requirements. Conversion to a laserjet improved appearance, speed, and reduced the cost. The font cartridge required to create code 3 of 9 bar codes on the laserjet is HP92286W1, commonly referred to as W1.

To read the bar codes we selected the Intermec model 9510 series of "online readers." This model was chosen for several reasons. The Intermec 9510 supports light wand or laser scanners, and can be easily programmed to add carriage returns or enter keys after the number is read. This makes it much easier to use standard bar codes in existing applications and eliminate the manual use of keyboards. The 9510 is modular in design, you may replace the wand reader, the cord, the decoder, or the power supply individually instead of the entire unit. Most importantly, the Intermec unit goes between the host and the terminal using standard 25-pin connectors which allows the same unit to be used with several different terminal types.

Because the host can't tell where the data came from (wand or keyboard), no special programming efforts or techniques are required, except for configuring the reader. The Powerhouse screens developed for data collection are the same if we use keyboards for manual entry or wands to read bar codes.

To overcome our response time constraint we decided to give this system its own dedicated HP3000. While this seems extreme, the cost was actually very competitive to a networked PC approach. Figure 2 lists the hardware configuration and costs.

Because this system is virtually stand-alone, we determined an INP link to our current Series 70 would easily satisfy our communication needs. This link will eventually be converted to LAN/3000. The MICROXE platform was chosen because of its expandability and its longer expected life. The standard MICRO being recently replaced by the LX and GX models.

To print unit serial number labels, we selected the HP2934. Because of its low cost, high quality bar codes and the outstanding reliability of the unit, it was an obvious choice given our desire to keep configurations as simple as possible.

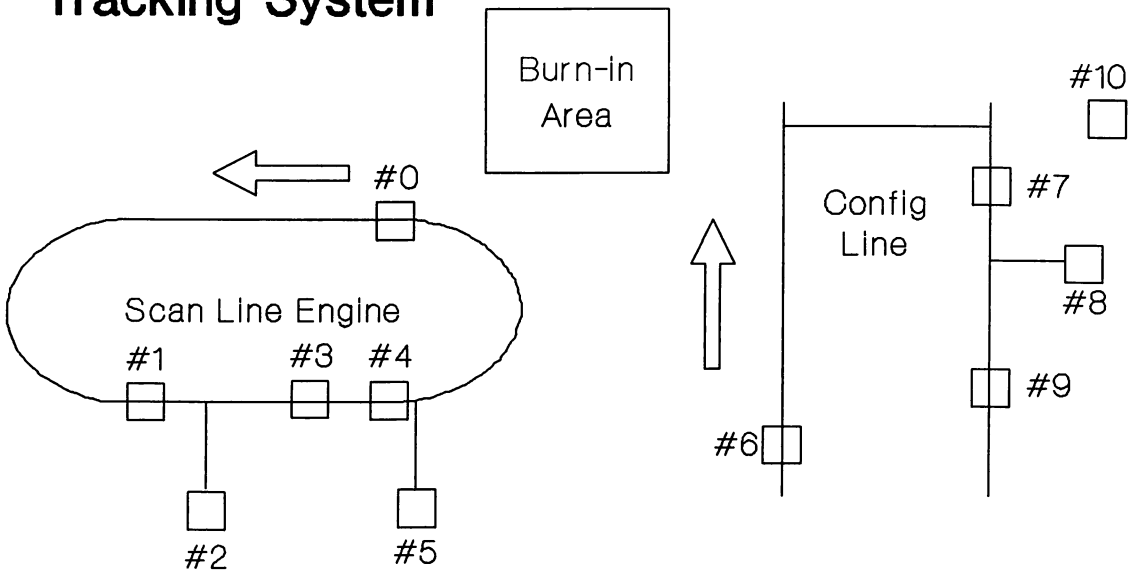
As you can see, bar coding applications are easy! Scenarios are possible that require no software changes to the data input function and several methods exist for creating bar coded output with minimal effort and expense. Our system consumed less than 2 man-months from design to turn on. This relatively short time is directly attributable to the use of a 4th generative language for program development and the Intermec readers being transparent to the HP3000.

To help you get started and understand how to create your own bar-coding system, Figure 3 contains a schematic of the database. Figure 4 shows the layout and source code for two of the screens used to collect data.

The advantages to be gained by using bar codes can be significant if absolute data accuracy is important in your applications, so get on the wagon now!

Freedom Assembly Tracking System

Figure 1



* #0 Start a Unit
* #1 Beamwalk Test
* #2 Beamwalk Rework
* #3 Motor & Main PCB Installation

#4 Scan Engine Test (2 PCs)

* #5 Scan Engine Rework

* #6 Configuration

#7 Final Test

* #8 Final Test Rework

#9 Photon Dr-Test

* #10 Shipping
* #11 Service (not shown)

* = Terminal/wand

Bar Codes in Factory Data Collection

Hardware Configuration List Figure 2

HP PART #	DESCRIPTION	List Price	Qty	Ext Price
32545A	MICRO XE w/ 4MB & PIC	$30,500	1	$30,500
40290A	ATP/M	$3,780	2	$7,560
opt 125	4-25 PIN MODEM & 4-RS232 DIRECT CONNECT			
7958A	130MB DISC DRIVE	$6,450	1	$6,450
C1001G	HP 700/92 TERMINAL	$895	9	$8,055
9144A	1/4" CARTRIDGE TAPE	$2,350	1	$2,350
30284A	NS Pt-to-Pt 3000/V LINK	$0	1	
opt 110	SYNC MODEM	$3,085	1	$3,085
32344A	NS3000/V NETWORK SERVICES	0	1	$0
opt 310	for MICRO XE	2040	1	$2,040
2934A	HP Printer (200cps)	2995	1	$2,995
	INTERMEC Model 9510 Wand Readers (complete)	660	9	$5,940 $0
		Total =		$68,975

This cost can be drastically reduced through one of several means:

1) Purchase used equipment in the after market or from HP's Remarketed Division
2) Use a VAR who will give you a volume discount
3) Purchase HP DEMO equipment.
4) Utilize current equipment where possible

Note: If we hadn't needed a dedicated processor, our total cost would have been: $16,990.

```
DATA BASE: FREE  .JOE

DATA BASE LANGUAGE ATTRIBUTE: NATIVE-3000

SET NAME:
   M-FAILURE,MANUAL

      ITEMS:
         FAIL-CODE,          X4          <<KEY ITEM>>
         DESCRIPTION,        X72

      CAPACITY: 251          ENTRIES: 20


SET NAME:
   M-REWORK,MANUAL

      ITEMS:
         REWORK-CODE,        X4          <<KEY ITEM>>
         DESCRIPTION,        X72

      CAPACITY: 101          ENTRIES: 4


SET NAME:
   M-SERIAL,MANUAL

      ITEMS:
         SERIAL-NO,          X8          <<KEY ITEM>>

      CAPACITY: 5003         ENTRIES: 271


SET NAME:
   M-TARGET,MANUAL

      ITEMS:
         PROD-DATE,          X6          <<KEY ITEM>>
         FIRST-UNIT,         X8
         LAST-UNIT,          X8
         TARGET-QTY,         J1
         ACTUAL-QTY,         J1

      CAPACITY: 503          ENTRIES: 10


SET NAME:
   M-UNIT,MANUAL

      ITEMS:
         UNIT,               X8          <<KEY ITEM>>
         TUBE,               X8
         PRE-AMP,            X8
         MOTOR,              X8
         MAIN-PCB,           X8
         DECODER,            X8
         MODEL,              X6
         POWER-BRICK,        X10
```

```
          CABLE,                X10
          SOFTWARE,             X10
          T1-FLAG,              X2
          T2-FLAG,              X2
          T3-FLAG,              X2
          TIME-0,               X12
          TIME-1,               X12
          TIME-3,               X12
          TIME-4,               X12
          TIME-6,               X12
          TIME-7,               X12

       CAPACITY: 1009           ENTRIES: 85


SET NAME:
   M-XREF,MANUAL

       ITEMS:
          XREF,                 X6          <<KEY ITEM>>

       CAPACITY: 101            ENTRIES: 11


SET NAME:
   D-FAILURE,DETAIL

       ITEMS:
          FAIL-CODE,            X4          <<SEARCH ITEM>>
          SEQ,                  X2          <<SORT ITEM>>
          F-LINE,               X72

       CAPACITY: 2021           ENTRIES: 5


SET NAME:
   D-TEST-1,DETAIL

       ITEMS:
          UNIT,                 X8          <<SEARCH ITEM>>
          TIME-T1,              X12         <<SORT ITEM>>
          FAIL-CODE-T1,         X4
          REWORK-T1,            X4
          TUBE,                 X8
          PRE-AMP,              X8

       CAPACITY: 3042           ENTRIES: 82


SET NAME:
   D-TEST-2,DETAIL

       ITEMS:
          UNIT,                 X8          <<SEARCH ITEM>>
          TIME-T2,              X12         <<SORT ITEM>>
          FAIL-CODE-T2,         X4
          REWORK-T2,            X4
          TUBE,                 X8
          PRE-AMP,              X8
          MOTOR,                X8
```

```
            MAIN-PCB,              X8
            TD-POWER-2-FLAG,       X2
            TD-POWER-2,            J1
            LINE-SEP-FLAG,         X2
            LINE-SEP,             4J1
            FWHM-FLAG,             X2
            FWHM-FF,              20J1
            FWHM-W,               20J1
            FWHM-L,                J1
            CENTROID-FLAG,         X2
            CENTROID-FF,          20J1
            CENTROID-W,           20J1
            GLITCH-FLAG,           X2
            GLITCH,               20J1
            FIRST-DRV-FLAG,        X2
            FIRST-DRV,            20J1

         CAPACITY: 3000            ENTRIES: 65


SET NAME:
   D-TEST-3,DETAIL

         ITEMS:
            UNIT,                 X8          <<SEARCH ITEM>>
            TIME-T3,              X12         <<SORT ITEM>>
            FAIL-CODE-T3,         X4
            REWORK-T3,            X4
            DECODER,              X8
            POWER-UP-FLAG,        X2
            AC-IN,                J1
            DC-OUT,               J1
            RIPPLE,               J1
            TD-POWER-3-FLAG,      X2
            TD-POWER-3,           J1
            MO-SPEED-FLAG,        X2
            MO-SPEED,             J1
            READ-A-FLAG,          X2
            READ-E-FLAG,          X2
            PORT-FLAG,            X2
            INTERFACE-FLAG,       X2
            BEEP-FLAG,            X2
            LED-FLAG,             X2
            EEPROM-FLAG,          X2
            EPROM-VERSION,        X12

         CAPACITY: 3008            ENTRIES: 1
```

PATH IDENTIFYING INFORMATION

MASTER SET NAME	ASSOCIATED DETAIL SET NAME	SEARCH ITEM NAME	SORT ITEM NAME
M-FAILURE	D-FAILURE	FAIL-CODE	SEQ
M-REWORK			
M-SERIAL			
M-TARGET			
M-UNIT	D-TEST-1	UNIT	TIME-T1
	D-TEST-2	UNIT	TIME-T2
	D-TEST-3	UNIT	TIME-T3
M-XREF			

DETAIL SET NAME	SEARCH ITEM NAME	SORT ITEM NAME	ASSOCIATED MASTER SET NAME
D-FAILURE	!FAIL-CODE	SEQ	M-FAILURE
D-TEST-1	!UNIT	TIME-T1	M-UNIT
D-TEST-2	!UNIT	TIME-T2	M-UNIT
D-TEST-3	!UNIT	TIME-T3	M-UNIT

```
**********************************************************************
*MODE:F ACTION:L                                          MMK010R
*                          BEAMWALK TEST
*
*    UNIT #        6000101
*    TUBE #        DE400101
*    PRE AMP       DC300101
*01 RESULT CODE    OK
*
*    TEST PASSED.
*
*
*
*
*
*
*
*
*
*
**********************************************************************
```

```
**********************************************************************************
*MODE:F ACTION:L                                                          MMK060R
*                                    CONFIGURATION
*
*   UNIT #
*
*   MODEL #
*
*   DECODER NUMBER
*
*   POWER BRICK
*
*   CABLE NUMBER
*
*   SOFTWARE
*
*
*
*
*
*
*
*
*
*OPERATION INCONSISTENT WITH ACCESS TYPE  (FSERR 40) (BUFFER*01)
**********************************************************************************
```

Figure 4
Page 2

```
SCREEN MMK010R
FILE D-TEST-1 ALIAS TEST-1 NODELETE
FILE M-UNIT SECONDARY NOITEMS NODELETE
     ACCESS VIA UNIT USING UNIT OF TEST-1
FILE M-SERIAL DESIGNER ALIAS SERIAL-T
FILE M-SERIAL DESIGNER ALIAS SERIAL-P
FILE D-TEST-1 DESIGNER ALIAS TEST-X
FILE M-FAILURE REFERENCE
     ACCESS USING FAIL-CODE-T1
FILE M-XREF REFERENCE
ITEM TIME-T1 FINAL ASCII(SYSDATE,6) + ASCII(SYSTIME / 100, 6)
ITEM T1-FLAG &
     FINAL "P1" IF FAIL-CODE-T1 ="OK" AND T1-FLAG = "  " &
      ELSE "P " IF FAIL-CODE-T1 ="OK" AND T1-FLAG <> "  " &
      ELSE "F "
ITEM TUBE OF M-UNIT FINAL TUBE OF TEST-1
ITEM PRE-AMP OF M-UNIT FINAL PRE-AMP OF TEST-1
ITEM TIME-1 FINAL ASCII(SYSDATE,6) + ASCII(SYSTIME / 100, 6) &
     IF TIME-1 = "              "
HILITE DATA INVERSE, DISPLAY INVERSE HALFTONE, &
      MESSAGE INVERSE, ERROR INVERSE BLINKING AUDIBLE
TITLE "MMK010R" AT 1,72
TITLE "BEAMWALK TEST" AT 2,41 CENTERED
SKIP 1
FIELD UNIT OF TEST-1 NOID SIZE 7 REQUIRED &
      LOOKUP ON M-UNIT MESSAGE 10
SKIP 1
FIELD TUBE OF TEST-1 NOID REQUIRED &
      LOOKUP ON M-XREF USING "TU" + TUBE [1:2] MESSAGE 11
SKIP 1
FIELD PRE-AMP OF TEST-1 NOID REQUIRED &
      LOOKUP ON M-XREF USING "PA" + PRE-AMP [1:2] MESSAGE 12
SKIP 1
FIELD FAIL-CODE-T1 LABEL "RESULT CODE" REQUIRED &
      LOOKUP ON M-FAILURE MESSAGE 13
SKIP 1
FIELD DESCRIPTION NOID NOLABEL DATA AT ,4 DISPLAY
PROCEDURE PROCESS UNIT
  BEGIN
    WHILE RETRIEVING TEST-X BACKWARDS USING UNIT OF TEST-1
      BEGIN
        IF FAIL-CODE-T1 OF TEST-X <> "OK" AND REWORK-T1 OF TEST-X = "
          THEN BEGIN
            ERROR 14
            BREAK
            END
        END
    END
PROCEDURE EDIT TUBE
  BEGIN
    IF FIELDTEXT <> TUBE OF M-UNIT
      THEN BEGIN
        GET SERIAL-T USING FIELDTEXT OPTIONAL
        IF ACCESSOK
          THEN ERROR 15
        END
    END
```

```
PROCEDURE EDIT PRE-AMP
  BEGIN
    IF FIELDTEXT <> PRE-AMP OF M-UNIT
      THEN BEGIN
        GET SERIAL-P USING FIELDTEXT OPTIONAL
        IF ACCESSOK
          THEN ERROR 16
        END
    END
PROCEDURE PREUPDATE
  BEGIN
    IF TUBE OF M-UNIT <> "        "
      THEN GET SERIAL-T USING TUBE OF M-UNIT
    LET SERIAL-NO OF SERIAL-T = TUBE OF TEST-1
    IF PRE-AMP OF M-UNIT <> "        "
      THEN GET SERIAL-P USING PRE-AMP OF M-UNIT
    LET SERIAL-NO OF SERIAL-P = PRE-AMP OF TEST-1
    END
PROCEDURE UPDATE
  BEGIN
    STARTLOG TEST-1
    PUT NOTDELETED M-UNIT
    PUT TEST-1
    PUT M-UNIT
    PUT SERIAL-P
    PUT SERIAL-T
    STOPLOG
    END
PROCEDURE DESIGNER 01
  BEGIN
    ACCEPT FAIL-CODE-T1
    DISPLAY DESCRIPTION
    END
```

```
SCREEN MMK060R AUTOUPDATE
FILE BUFFER NODELETE OPEN WRITE EXCLUSIVE
     TARGET 0
FILE M-UNIT SECONDARY NOITEMS NODELETE
     ACCESS VIA UNIT USING UNIT OF BUFFER
FILE D-TEST-2 DESIGNER
FILE M-SERIAL DESIGNER
FILE M-XREF REFERENCE
TEMPORARY FIRST-FLAG CHAR*1 INITIAL " " RESET AT STARTUP
ITEM MODEL OF M-UNIT FINAL MODEL OF BUFFER
ITEM DECODER OF M-UNIT FINAL DECODER OF BUFFER
ITEM TIME-6 OF M-UNIT &
     FINAL ASCII(SYSDATE,6) + ASCII(SYSTIME / 100, 6) &
           IF TIME-6 OF M-UNIT = "                    "
HILITE DATA INVERSE, DISPLAY INVERSE HALFTONE, &
       MESSAGE INVERSE, ERROR INVERSE BLINKING AUDIBLE
TITLE "MMK060R" AT 1,72
TITLE "CONFIGURATION" AT 2,41 CENTERED
SKIP 1
ALIGN (,4,21)
FIELD UNIT OF BUFFER SIZE 7 REQUIRED LOOKUP ON M-UNIT MESSAGE 10
SKIP 1
FIELD MODEL OF BUFFER SIZE 5 REQUIRED
SKIP 1
FIELD DECODER OF BUFFER REQUIRED &
      LOOKUP ON M-XREF USING "DC" + DECODER [1:2] MESSAGE 61, &
        ON M-XREF USING "DC" + DECODER [1:2] + MODEL [1:2] MESSAGE 63
SKIP 1
FIELD POWER-BRICK OF BUFFER LABEL "POWER BRICK" REQUIRED
SKIP 1
FIELD CABLE OF BUFFER REQUIRED
SKIP 1
FIELD SOFTWARE OF BUFFER REQUIRED
PROCEDURE PROCESS UNIT
  BEGIN
    LET FIRST-FLAG = "Y"
    WHILE RETRIEVING D-TEST-2 BACKWARDS USING UNIT OF BUFFER
      BEGIN
        IF FAIL-CODE-T2 <> "OK"
          THEN BEGIN
            IF FIRST-FLAG = "Y"
              THEN BEGIN
                ERROR 62
                BREAK
                END
              ELSE BEGIN
                IF REWORK-T2 = "    "
                  THEN BEGIN
                    ERROR 65
                    BREAK
                    END
                END
              END
        LET FIRST-FLAG = "N"
        END
    END
PROCEDURE PROCESS MODEL
```

```
  BEGIN
    IF NOT MATCHPATTERN(TRUNCATE(MODEL),"6(6|7|8|9)(0|1)(0|1|2)(0|2)")
      THEN ERROR 64
    END
PROCEDURE PROCESS DECODER
  BEGIN
    IF DECODER OF M-UNIT <> DECODER OF BUFFER
      THEN BEGIN
        GET M-SERIAL USING DECODER OF BUFFER OPTIONAL
        IF ACCESSOK
          THEN ERROR 60
        END
    END
PROCEDURE PREUPDATE
  BEGIN
    IF DECODER OF M-UNIT <> "          "
      THEN GET M-SERIAL USING DECODER OF M-UNIT
    LET SERIAL-NO OF M-SERIAL = DECODER OF BUFFER
    END
PROCEDURE UPDATE
  BEGIN
    STARTLOG M-UNIT
    PUT NOTDELETED M-UNIT
    PUT M-UNIT
    PUT M-SERIAL
    STOPLOG
    END
```