

Application Note

USING THE BAR CODE
SCANNER TO VALIDATE
PRODUCT IDENTIFICATION

Using the Bar Code Scanner to Validate Product Identification

This document explains how to use PASS 6.0 software to program the Dynalab Analyzer to scan bar-coded product information from a harness label, and validate that the product ID is correct.

This document has the following main sections:

- 1 A list of assumptions – knowledge required to perform the tasks outlined in this document.
- 2 An explanation of the problem
- 3 An explanation of the solution
- 4 An overview of the Dynalab solution to the problem, including an example Sequence.

Assumptions

To successfully use this document, the following knowledge is required:

- basic knowledge of how to enter harness data using PASS® 6.0
- knowledge of how to use the Sequence table to create a Sequence

For assistance on how to use features of PASS® 6.0, see the PASS® 6.0 Help file.

Problem

Pre-printed labels are used for product identification. Sometimes, wrong pre-printed labels are applied to the harnesses.

Solution

During the test, the Operator is prompted to scan the bar-coded product information on the harness label. The Analyzer then determines if the correct pre-printed labels are being used.

Solution Overview

The Dynalab Bar Code Laser Scanner is used to input bar code data directly into XL Series Analyzers using one of the RS-232 serial ports. The Analyzer can be programmed to compare the data input from the scanner with a predefined string of characters. The bar code on a product identification label can be validated by comparing it with the correct character string. In this way, usage of the correct pre-printed labels can be assured.

Serial Port Connection

Each scanner is equipped with a serial cable. The serial cable connector should be connected to the Analyzer's Serial Port 1, since this is the default port for serial data input to the Analyzer. If this is not suitable, the scanner's serial cable connector may be connected to Serial Port 2. This would require that the SERIAL2 Sequence item be inserted at the beginning of the program Sequence to instruct the Analyzer to monitor Serial Port 2 for input.

Programming the Analyzer to read and validate bar code data

The STRING Sequence item is used in the PASS Sequence to instruct the Analyzer to read a string of characters from the serial port. When STRING is executed, the Analyzer monitors the serial port until a valid character string is received.

Once a character string is input from the bar code scanner, the Analyzer can be programmed to compare the data input by the bar code scanner to a pre-defined entry in the Messages table. This is implemented with the STRCMP Sequence item. STRCMP requires a Parameter that specifies the message number of the entry in the Messages table to which the input is compared.

STRCMP is a Sequence item that compares the string stored in the string buffer to a message. STRCMP requires a parameter that specifies the message number that the string will be compared to. A character-by-character search of the message is performed and the position in the message where the string is encountered is stored in the holding register. If the string is not found in the message, the holding register will have a value of 0.

The use of the wildcard character "?" is permitted in the message (same usage as the wildcard character used in DOS). Any character in the string in the same position as the wildcard character in the message will be considered a match. Note: if wildcards are used, the number of characters in the string must equal the number of characters in the message. The wildcard character "?" can be thought of as a placeholder. The following examples show the holding register as a result of various STRCMPs

	String	Message	Holding Register	Description
1	ABCD	ABCD	1	String "ABCD" found at column 1 of message
2	AB	ABCD	1	String "AB" found at column 1 of message
3	BC	ABCD	2	String "BC" found at column 2 of message
4	AA	??	1	String "AA" found at column 1 of message, the wildcard characters "??" are considered a match of "AA" in the string
5	ABC123	A??1?3	1	String "ABC123" found at column 1 of message, the three wildcard characters in the message are considered a match of "B", "C", and "2" in the string
6	ABCDE	ABCD	0	String "ABCDE" not found in message, the string contains more characters than the message
7	123	?123	0	String "123" not found in message, since a wildcard is present in the message, the number of characters in the message must equal the number of characters in the string

Note: It is not possible to determine an exact match. If the scanned data is an exact match, STRCMP will place a 1 in the holding register, as in example 1 above. However, if the scanned data is a substring of the message table entry starting in position 1, then STRCMP will also place a 1 in the holding register as shown in example 2 above. Thus, it is not possible to distinguish between an exact match or a substring in position 1.

To validate that the data input by the bar code scanner either matches or is a sub-string of an entry in the Messages table, an entry in the Messages table must be created which contains a string of characters that is to be compared to the scanned input. For instance, if the scanned bar code characters are: H-2672-R77-1, then a message must be constructed as follows:

Messages		
	Number	Message
	2	H-2677-R77-1

Wild Card Match

To validate that the scanned data is a partial match, an entry in the Messages table must be created that uses the wildcard character: '?'. Wherever the character '?' appears in the message, it means that any character is valid in that position. So, for the entry in the Messages table shown below, the two character positions represented by the wildcard may contain any character. If the wildcard character is used, it is important to note that the length of the scanned character string must be exactly the same as the length of the pre-defined message.

Messages		
	Number	Message
	2	H-267?-R77-?

? means that any character will match in this position

Making the comparison

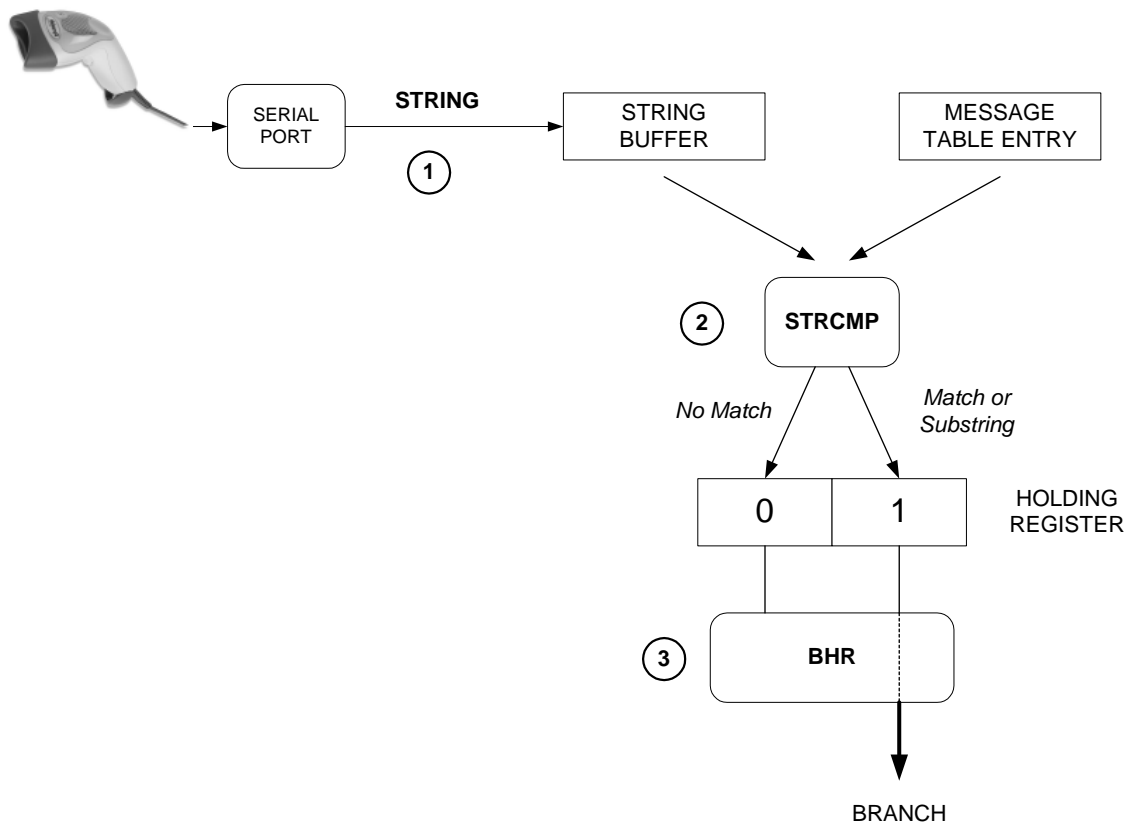
Once the message is defined, the Sequence must contain the following 3 items:

Sequence		
	Command	Value
1	MESSAGE	1
2	STRING	
3	STRCMP	2
4	BHR	8
5	SCOMP	12

1. **STRING** instructs the Analyzer to monitor Serial Port 1 and wait until a character string has been scanned.
2. **STRCMP** instructs the Analyzer to compare the scanned character string to the specified message – in this example it is Message 2. If **STRCMP** determines that the scanned character string either matches or is a substring of Message 2, the value of the holding register is set to 1. If **STRCMP** determines that the scanned character string does not match Message 2 nor is a substring of Message 2, the value of the holding register is set to 0.

3. **BHR** branches to the specified line in the Sequence if the value of the holding register is greater than 0. In this case, BHR branches to line 8. If the value of the holding register is 0, then execution of the Sequence continues at the next line.

These three steps for making the comparison are illustrated below. Note that BHR causes a branch only if the string read by the scanner matches the message table entry or is a substring of the message table entry.



Example Sequence

The following example Sequence illustrates how to program the Analyzer to receive data from a scanner and validate the data against a pre-defined character string.

This Sequence prompts the operator to scan the bar code label on the harness, waits for input on the serial port, and compares the input to a pre-defined character string. The Sequence flow continues if the harness label is valid, or branches if the harness label is not valid.

Example Sequence for using a Bar Code Scanner to validate Product Identification

Line	Command	Parameter	Application Effect
1	MESSAGE	1	Displays message 1: "SCAN Bar Code Label"
2	STRING		Reads string variable from serial port
3	STRCMP	2	Compares input string to message 2: H-267?-R77-?
4	BHR	8	Branch to line 8 if the holding register value is greater than zero
5	SOUND	12	Play a sound indicating that the harness label is not valid
6	KMESSAGE	3	Display message 3: "Invalid Label: x Press Start to Continue"
7	GOTO	1	Go to Line 1
8	SOUND	3	Play a sound indicating that the harness label is valid
9	TEST	MAIN	Perform a complete scan of the MAIN Netlist
10	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
11	KWAIT		Pause until START Button is depressed
12	REPEAT		Repeat the Sequence, starting at line 1

- Line 1** MESSAGE 1: Displays the message defined as message number 1 in the Messages table. This message is defined as "SCAN Bar Code Label".
- Line 2** STRING: reads a string of characters from the Analyzer's serial port. In this example, STRING is reading characters from the Bar Code Scanner. After STRING is invoked, the Analyzer will wait until the bar code is scanned. After the bar code is scanned, the string of characters is stored in the Analyzer's string buffer and program execution continues with the next Sequence item. (By default, the Analyzer monitors Serial Port 1 for input from a scanner. If this is not suitable, the SERIAL2 Sequence item may be inserted at the beginning of the sequence to instruct the Analyzer to monitor Serial Port 2 for input.)
- Line 3** STRCMP: compares the contents of the Analyzer's string buffer to the contents of the entry in the Messages table referenced by the parameter. In this case, the parameter is 2 – so STRCMP will compare the scanned bar code characters with the contents of message 2 in the Messages table. In this example, message 2 is defined as "H-267?-R77.?". STRCMP will declare a successful match if the scanned bar code string is of exactly the same length as message 2, and if all the characters match exactly except the fifth and eleventh characters. If STRCMP finds a match, the value of the Analyzer's holding register is set to 1. If STRCMP

determines that the two character strings do not match, the value of the holding register is set to zero.

- Line 4** BHR: Examines the value of the Analyzer's holding register and branches to the specified line number if the value of the holding register is greater than zero. In this case, execution branches to line 8 if the value of the holding register is greater than 0 or continues with the next line if the value of the holding register is equal to 0. In this example, this means that execution branches to line 8 if the scanned bar code label is valid (successful match found by STRCMP), or execution continues with the next line if the scanned bar code label is not valid.
- Line 5** SOUND 12: Plays a sound to indicate that the bar code label is NOT valid.
- Line 6** KMESSAGE 3: Displays message 3: "Invalid Label |x – Press Start Continue". The Analyzer will continue to display this message until the operator presses the START button (the "|x" format symbol will cause the scanned characters to be displayed as part of the message).
- Line 7** GOTO 1: instructs the Analyzer to go to line 1 which repeats execution of the Sequence
- Line 8** SOUND 3: Plays a sound to indicate that the bar code label is valid.
- Line 9** TEST: Performs a complete test scan of the entire harness.
- Line 10** REPORT: Displays a summary report
- Line 11** KWAIT: Waits for the operator to press the START button
- Line 12** REPEAT: instructs the Analyzer to go to line 1 which repeats execution of the Sequence.