Application Note

HOW TO PRINT A HARNESS LABEL AND A BOX LABEL

Dynalab
TEST SYSTEMS

How to Print a Harness Label and a Box Label

Introduction

A common label printing application requires that the PASS program be designed to print two distinct labels: one for each good harness, and another for the box in which the good harnesses are packaged. This document explains how to do this.

This document contains the following main sections:

1. a list of assumptions – knowledge required to perform the tasks outlined in this document
2. An example Sequence for printing a harness label and a box label using the same printer
3. An example Sequence for printing a harness label and a box label using two printers to support different label sizes
4. An example Sequence for printing a harness label and a box label using two printers to support different label sizes, with a prompt to reset the counters.

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.

Copyright 2002

Example 1

The following assumptions apply to this example:

- A harness label is printed for each good harness
- A shipping box contains 50 good harnesses. Therefore, a box label must be printed for every 50 good harnesses.
- The harness label contains a serial number
- The harness label and the box label are the same size, allowing for use of a single printer.

The PLABEL Sequence item is used to print a label. If a second distinct label is to be printed, the PLABEL2 Sequence item is used. In this example, PLABEL is used to print the harness label, and PLABEL2 is used to print the box label.

RCOUNT is a counter available in PASS. It will be used to count the number of harnesses tested. When RCOUNT reaches 50, the box label will be printed.

SCOUNT is a counter available in PASS. It will be used to generate a serial number. SCOUNT will automatically roll over to zero after it reaches its maximum value of 65535

Example Sequence for printing a harness label and a box label: labels are same size

Line	Sequence Item	Parameter	Description
1	ADVOFF		Prevents advancing by Start button on error
2	TEST	MAIN	Performs a complete Netlist scan of MAIN Netlist
3	SCOUNT++		Increments SCOUNT by 1
4	RCOUNT++		Increments RCOUNT by 1
5	PLABEL	4X1-HARN	Prints a label, using template file 4X1-HARN
6	HOLDING=	50	Sets the holding register to 50
7	MESSAGE	1	HARNESS IS GOOD! 3 of 50
8	RCOUNT==		Compares the value of RCOUNT to the holding register
9	BNHR	12	Branches to line 12 if holding register value is zero
10	PLABEL2	4X1-BOX	Prints a label, using template file 4X1-BOX
11	RCOUNT=0		Set RCOUNT to zero
12	AUTO	MAIN	Performs continuous continuity scan until removal of harness
13	REPEAT		Repeats sequence from first line

Line 1 ADVOFF is a flag that tells the Analyzer to prevent any attempt to advance by pressing the Start button when stopped on an error.

Line 2 TEST performs a complete Netlist scan and tests for continuity and shorts. When an error is found, TEST stops and displays the error. Since ADVOFF is set in line 1, the operator must either fix the error or remove the harness and restart the program. The operator cannot advance by pressing the Start button.

Line 3 SCOUNT++ increments SCOUNT by one. This counter is used to generate a serial number. The serial number is one of the

Copyright 2002

variables printed on each harness label. The serial number will automatically roll over to zero after reaching its maximum value of 65535.

Line 4 RCOUNT++ increments RCOUNT by one. This counter is used to keep track of the number of harnesses tested.

Line 5 PLABEL prints the harness label, using the label template specified by the Parameter – in this case, 4X1-HARN. This line will only be executed if the harness passed all tests, since ADVOFF was set in Line 1.

Line 6 HOLDING= sets the value of the holding register equal to the Parameter. The Parameter should be equal to the number of harnesses in a box – in this case, 50.

Line 7 MESSAGE displays a message indicating that the harness is good, and displays the current value of RCOUNT and the value of the holding register. The message is defined in PASS as follows:

Line 8 RCOUNT== compares the value of RCOUNT to the holding register. If RCOUNT is equal to the value of the holding register, the holding register is set to 1. If RCOUNT is NOT equal to the value of the holding register, the holding register is set to zero.

Line 9 BNHR branches to the line number specified by the Parameter if the value of the holding register is zero. In this example, since the holding register is set to zero in Line 8 only if RCOUNT is NOT equal to 50, then execution will branch to line 12 if there are not yet enough harnesses to fill the box. Branching to line 12 skips line 10, which prints the box label. On the other hand, if RCOUNT is equal to 50, then the value of the holding register was set to 1 in Line 8, and execution does NOT branch to line 12, but continues with the next line.

Line 10 PLABEL2 prints the box label using the label template specified by the Parameter – in this case, 4X1-BOX. This line is executed only if the value of RCOUNT is equal to the number of harnesses per box – in this case, 50.

Line 11 RCOUNT=0 resets the value of RCOUNT to zero, to start counting for the next box.

Line 12 AUTO performs a continuous scan for continuity for all items in the specified Netlist (MAIN), looking for all harness connections to be open. Once all connections are open, the Analyzer continues execution with the next Sequence item. AUTO is used to determine that a harness has been completely removed from the fixture

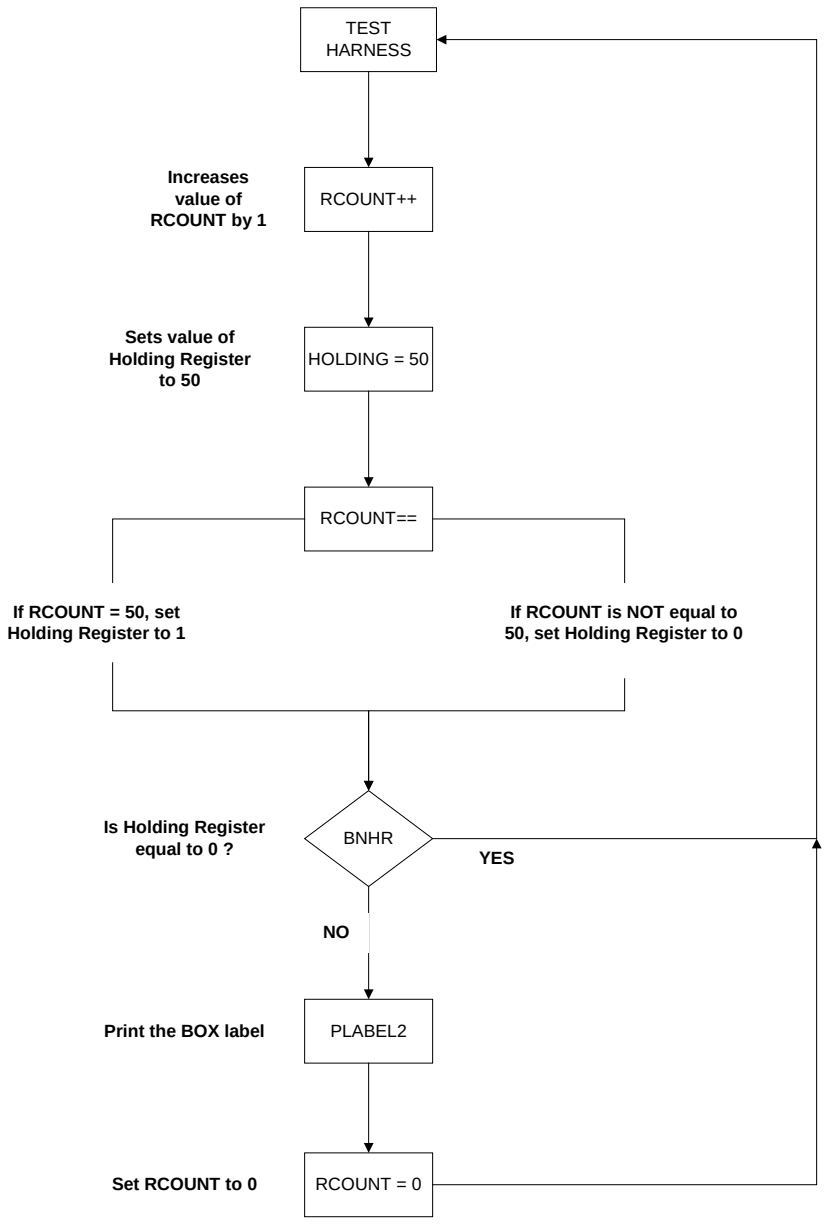
Line 13 REPEAT repeats the sequence from the first line.

Copyright 2002

It is important to understand that RCOUNT is actually the Report Counter, and is incremented by the Analyzer whenever the REPORT Sequence item is executed. Therefore, when using RCOUNT for a different purpose, as in this example, it is necessary to avoid use of the REPORT Sequence item. The example above avoids the REPORT Sequence item by setting ADVOFF, resulting in only good harnesses getting past the TEST item in line 2. This allows for the use of a message indicating the harness is good in line 6.

Line	Sequence Item	Parameter
1	ADVOFF	
2	TEST	MAIN
3	SCOUNT++	
4	RCOUNT++	
5	PLABEL	4X1-HARN
7	HOLDING=	50
6	MESSAGE	1
8	RCOUNT==	
9	BNHR	12
10	PLABEL2	4X1-BOX
11	RCOUNT=0	
12	AUTO	MAIN
13	REPEAT	

The following flowchart illustrates the algorithm used in the Sequence to keep track of the number of harnesses tested in order to determine when to print a box label. In this example, a box label is printed for every 50 harnesses. RCOUNT is used to count the number of harnesses.



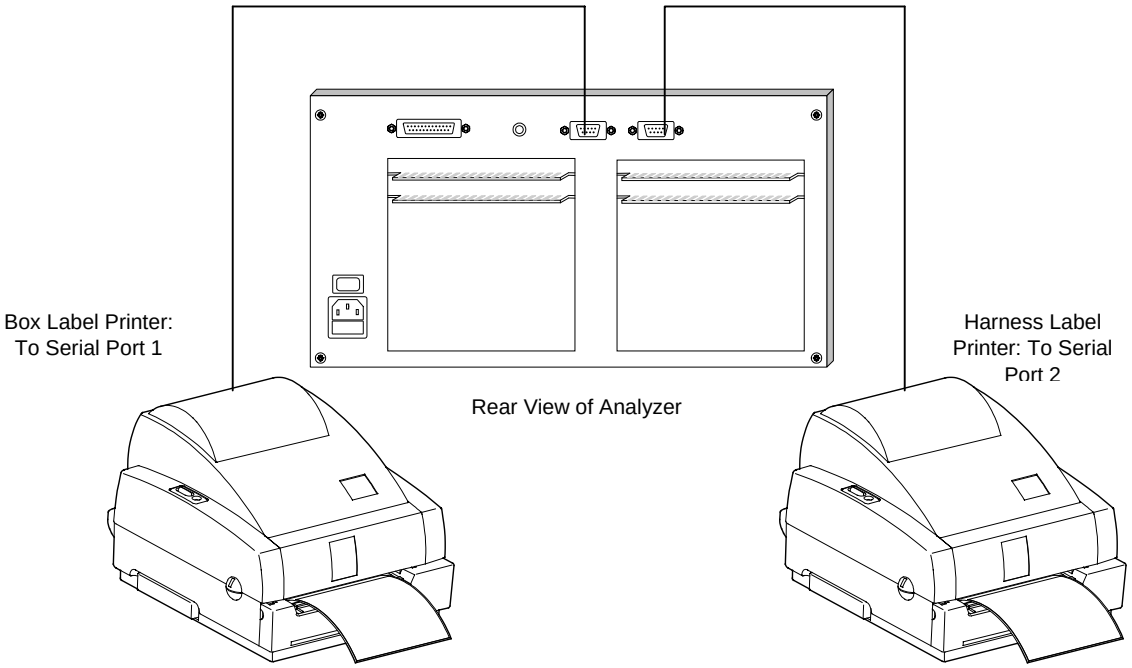
Copyright 2002

Example 2

In the previous example, the assumption was made that the harness label and box label are the same size. This allows for printing each label using the same printer.

There may be situations where the box label is of a different size than the harness label. In order to accommodate two different size labels, it is necessary to use two different printers. Each printer will have a different size label – one for the harness and one for the box.

Each printer is connected to one of the Analyzer's serial ports as shown in the diagram below:



In the PASS 6.0 program, it will be necessary to specify which serial port is being used when printing a label. The box label will be printed using serial port 1. The harness label will be printed using serial port 2.

The SERIAL1 and SERIAL2 Sequence items will be used to select the appropriate serial port when printing a label. This is illustrated in the example sequence shown below.

Copyright 2002

Example Sequence for printing a harness label and a box label: labels are different sizes

Line	Sequence Item	Parameter	Description
1	ADVOFF		Prevents advancing by Start button on error
2	TEST	MAIN	Performs a complete Netlist scan of MAIN Netlist
3	SCOUNT++		Increments SCOUNT by 1
4	RCOUNT++		Increments RCOUNT by 1
5	**SERIAL2**		**Sets serial port 2 as the active serial port**
6	PLABEL	4X1-HARN	Prints a label, using template file 4X1-HARN
7	HOLDING=	50	Sets the holding register to 50
8	MESSAGE	1	HARNESS IS GOOD! 3 of 50
9	RCOUNT==		Compares the value of RCOUNT to the holding register
10	BNHR	14	Branches to line 14 if holding register value is zero
11	**SERIAL1**		**Sets serial port 1 as the active serial port**
12	PLABEL2	3X5-BOX	Prints a label, using template file 3X5-BOX
13	RCOUNT=0		Set RCOUNT to zero
14	AUTO	MAIN	Performs continuous continuity scan until removal of harness
15	REPEAT		Repeats sequence from first line

Line 1 ADVOFF is a flag that tells the Analyzer to prevent any attempt to advance by pressing the Start button when stopped on an error.

Line 2 TEST performs a complete Netlist scan and tests for continuity and shorts. When an error is found, TEST stops and displays the error. Since ADVOFF is set in line 1, the operator must either fix the error or remove the harness and restart the program. The operator cannot advance by pressing the Start button.

Line 3 SCOUNT++ increments SCOUNT by one. This counter is used to generate a serial number. The serial number is one of the variables printed on each harness label. The serial number will automatically roll over to zero after reaching its maximum value of 65535.

Line 4 RCOUNT++ increments RCOUNT by one. This counter is used to keep track of the number of harnesses tested.

Line 5 SERIAL2 sets serial port 2 as the active serial port.

Line 6 PLABEL prints the harness label, using the label template specified by the Parameter – in this case, 4X1-HARN. This line will only be executed if the harness passed all tests, since ADVOFF was set in Line 1. The label is printed using serial port 2 because SERIAL2 was set in Line 5. The harness label printer must be attached to the Analyzer's serial port 2.

Line 7 HOLDING= sets the value of the holding register equal to the Parameter. The Parameter should be equal to the number of harnesses in a box – in this case, 50.

Line 8 MESSAGE displays a message indicating that the harness is good, and displays the current value of RCOUNT and the value of the holding register

Copyright 2002

Line 9 RCOUNT== compares the value of RCOUNT to the holding register. If RCOUNT is equal to the value of the holding register, the holding register is set to 1. If RCOUNT is NOT equal to the value of the holding register, the holding register is set to zero.

Line 10 BNHR branches to the line number specified by the Parameter if the value of the holding register is zero. In this example, since the holding register is set to zero in Line 9 only if RCOUNT is NOT equal to 50, then execution will branch to line 14 if there are not yet enough harnesses to fill the box. Branching to line 14 skips line 12, which prints the box label. On the other hand, if RCOUNT is equal to 50, then the value of the holding register was set to 1 in Line 8, and execution does NOT branch to line 14, but continues with the next line.

Line 11 SERIAL1 sets serial port 1 as the active serial port

Line 12 PLABEL2 prints the box label using the label template specified by the Parameter – in this case, 4X1-BOX. This line is executed only if the value of RCOUNT is equal to the number of harnesses per box – in this case, 50. The label is printed using serial port 1 because SERIAL1 was set in Line 11. The box label printer must be attached to the Analyzer's serial port 1

Line 13 RCOUNT=0 resets the value of RCOUNT to zero, to start counting for the next box.

Line 14 AUTO performs a continuous scan for continuity for all items in the specified Netlist (MAIN), looking for all harness connections to be open. Once all connections are open, the Analyzer continues execution with the next Sequence item. AUTO is used to determine that a harness has been completely removed from the fixture

Line 15 REPEAT repeats the sequence from the first line.

Copyright 2002

Example 3

The values of RCOUNT and SCOUNT are persistent. They do not reset when the Analyzer is powered down or when a new program is loaded. The program automatically resets RCOUNT when it reaches 50. SCOUNT is not reset in the program, but automatically rolls over to zero after reaching its maximum value of 65535. There may be times when it is desirable to manually initiate a reset one or both of these counters.

One way to reset the counters is to modify the PASS program such that when the program is initially executed, the operator is prompted to indicate if the counters should be reset.

Another way is to manually reset the counters through the Analyzer's OPTIONS MENU.

The following is an example sequence illustrating how to prompt the operator with the option to reset the serial number counter and box counter when the program is initially executed.

Example Sequence: Prompt the operator to reset Serial Number and/or Box Count

Line	Sequence Item	Parameter	Description
1	**MENU**	**10**	**Reset Serial Num? N Y**
2	**SWITCH**		**Go to CASE label matching holding value**
3	**CASE**	**2**	**Label for holding value = 2 (selection is YES)**
4	**SCOUNT=0**		**Set SCOUNT equal to zero**
5	**CASE**	**1**	**Label for holding value = 1 (selection is NO)**
6	**MENU**	**15**	**Reset Box Count? N Y**
7	**SWITCH**		**Go to CASE label matching holding value**
8	**CASE**	**2**	**Label for holding value = 2 (selection is YES)**
9	**RCOUNT=0**		**Set RCOUNT equal to zero**
10	**CASE**	**1**	**Label for holding value = 1 (selection is NO)**
11	ADVOFF		Prevents advancing by Start button on error
12	TEST	MAIN	Performs a complete Netlist scan of MAIN Netlist
13	SCOUNT++		Increments SCOUNT by 1
14	RCOUNT++		Increments RCOUNT by 1
15	SERIAL2		Sets serial port 2 as the active serial port
16	PLABEL	4X1-HARN	Prints a label, using template file 4X1-HARN
17	HOLDING=	50	Sets the holding register to 50
18	MESSAGE	1	HARNESS IS GOOD!
19	RCOUNT==		Compares the value of RCOUNT to the holding register
20	BNHR	24	Branches to line 24 if holding register value is zero
21	SERIAL1		Sets serial port 1 as the active serial port
22	PLABEL2	4X1-BOX	Prints a label, using template file 4X1-BOX
23	RCOUNT=0		Set RCOUNT to zero
24	AUTO	MAIN	Performs continuous continuity scan until removal of harness
25	GOTO	12	Go to line number 12

Copyright 2002

Line 1 The MENU Sequence item displays a menu using the message text associated with the message number specified as the parameter – in this case, 10. Message 10's text is defined as:

```
Reset Ser Num?: N Y
```

This causes the menu and selections to be displayed by the Analyzer as follows:

```
  Reset Serial Num ?
>N
 Y
```

The operator uses the DOWN and UP buttons to place the cursor next to the desired option. Once the cursor is next to the desired option, the operator presses the START button to make the selection.

If the operator selects the first option (N), the Analyzer's holding register is set to 1. If the operator selects the second option (Y), the holding register is set to 2.

Line 2 SWITCH causes execution to branch to the CASE label corresponding to the value of the holding register. In this situation, execution branches to Line 5 if the operator selected N (holding register = 1), or to Line 3 if the operator selected Y (holding register = 2).

Line 3 CASE – since the parameter is 2, this is the CASE label to which execution branches when the holding register = 2. This is where execution resumes when the operator selects Y.

Line 4 SCOUNT=0 causes the value of SCOUNT to be equal to zero.

Line 5 CASE – since the parameter is 1, this is the CASE label to which execution branches when the holding register = 1. This is where execution resumes when the operator selects N

Line 6 The MENU Sequence item displays a menu using the message text associated with the message number specified as the parameter – in this case, 15. Message 15's text is defined as:

```
Reset Box Count?: N Y
```

This causes the menu and selections to be displayed by the Analyzer as follows:

```
  Reset Box Count ?
>N
 Y
```

The operator uses the DOWN and UP buttons to place the cursor next to the desired option. Once the cursor is next to the desired option, the operator presses the START button to make the selection.

Copyright 2002

If the operator selects the first option (N), the Analyzer's holding register is set to 1. If the operator selects the second option (Y), the holding register is set to 2.

Line 7 SWITCH causes execution to branch to the CASE label corresponding to the value of the holding register. In this situation, execution branches to Line 10 if the operator selected N (holding register = 1), or to Line 8 if the operator selected Y (holding register = 2).

Line 8 CASE – since the parameter is 2, this is the CASE label to which execution branches when the holding register = 2. This is where execution resumes when the operator selects Y.

Line 9 RCOUNT=0 causes the value of RCOUNT to be equal to zero.

Line 10 CASE – since the parameter is 1, this is the CASE label to which execution branches when the holding register = 1. This is where execution resumes when the operator selects N

Line 11 ADVOFF is a flag that tells the Analyzer to prevent any attempt to advance by pressing the Start button when stopped on an error.

Line 12 TEST performs a complete Netlist scan and tests for continuity and shorts. When an error is found, TEST stops and displays the error. Since ADVOFF is set in line 1, the operator must either fix the error or remove the harness and restart the program. The operator cannot advance by pressing the Start button.

Line 13 SCOUNT++ increments SCOUNT by one. This counter is used to generate a serial number. The serial number is one of the variables printed on each harness label. The serial number will automatically roll over to zero after reaching its maximum value of 65535.

Line 14 RCOUNT++ increments RCOUNT by one. This counter is used to keep track of the number of harnesses tested.

Line 15 SERIAL2 sets serial port 2 as the active serial port

Line 16 PLABEL prints the harness label, using the label template specified by the Parameter – in this case, 4X1-HARN. This line will only be executed if the harness passed all tests, since ADVOFF was set in Line 1.

Line 17 HOLDING= sets the value of the holding register equal to the Parameter. The Parameter should be equal to the number of harnesses in a box – in this case, 50.

Line 18 MESSAGE displays a message indicating that the harness is good.

Line 19 RCOUNT== compares the value of RCOUNT to the holding register. If RCOUNT is equal to the value of the holding register, the holding register is set to 1. If RCOUNT is NOT equal to the value of the holding register, the holding register is set to zero.

Line 20 BNHR branches to the line number specified by the Parameter if the value of the holding register is zero. In this example, since the holding register is set to zero in Line 19 only if RCOUNT is

Copyright 2002

NOT equal to 50, then execution will branch to line 24 if there are not yet enough harnesses to fill the box. Branching to line 24 skips line 22, which prints the box label. On the other hand, if RCOUNT is equal to 50, then the value of the holding register was set to 1 in Line 19, and execution does NOT branch to line 24, but continues with the next line.

Line 21 SERIAL1 sets serial port 1 as the active serial port

Line 22 PLABEL2 prints the box label using the label template specified by the Parameter – in this case, 4X1-BOX. This line is executed only if the value of RCOUNT is equal to the number of harnesses per box – in this case, 50.

Line 23 RCOUNT=0 resets the value of RCOUNT to zero, to start counting for the next box.

Line 24 AUTO performs a continuous scan for continuity for all items in the specified Netlist (MAIN), looking for all harness connections to be open. Once all connections are open, the Analyzer continues execution with the next Sequence item. AUTO is used to determine that a harness has been completely removed from the fixture

Line 25 GOTO 12 causes execution to resume at Line 12.

.

Copyright 2002