معـــهـــد التكنولوجـيا التطبيقيـــة

INSTITUTE OF APPLIED TECHNOLOGY

Micro-Controllers

Module 4: Digital Display

PREPARED BY

IAT Curriculum Unit

August 2008

Module 4: Digital Display

Module Objectives

1. Realize digital display concept
2. Realize 7-Segment display LED
3. Learn how to program 7-Segment display

Module Contents

Digital Display

In this module you will learn about digital display of information, and how to program the Micro-Controller to display numbers and letters on the digital display.

Fig 4.1: Analog girl's watch (left) and digital boy's watch (right).

When you need to know the time you simply look at your watch. Some times you need to follow the minutes and hours pointers to figure the time like the girly watch shown in figure 4.1 to the left, and other times the watch displays the time in digits simple to read like the Micky mouse watch shown to the right. Both watches will display time information, but we call the type of display in the girly pink watch as analog display. The time display in the boy's watch is referred to as digital display.
Which display is easier to read?
Which one is more accurate?
Where else can you find digital display?

In digital display, each digit is called 7-Segment display. A 7-Segment display is rectangular block of 7 lines of equal length that can be lit selectively to display digits and some letters. Every line corresponds to one LED that can be controlled individually. Figure 4.2 shows a part drawing of the 7-Segment LED display you will use in this module's activities. It has one additional LED, a dot that can be used as a decimal point. The letters shown corresponds to lines. The numbers refer to the pins numbers. Table 4.1 lists the pin mapping. Pins' maps help you connect the 7-Segment display in the electric circuit. It informs the pin number, pin name and pin reference.

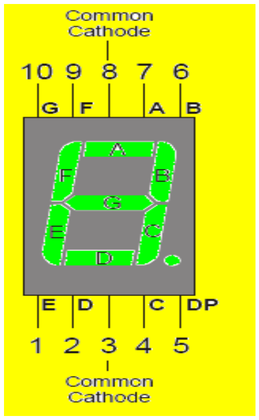


Fig 4.2: 7-Segment LED Display Drawing and Pin Map

Pin No.	Mapping
Pin 1	Controls Segment E
Pin 2	Controls Segment D
Pin 3, 8	Common Cathode
Pin 4	Controls Segment C
Pin 5	Controls decimal point
Pin 6	Controls Segment B
Pin 7	Controls Segment A
Pin 9	Controls Segment F
Pin 10	Controls Segment G

Table 4.1: 7 Segment Display pin mapping

Figure 4.3 illustrates the 7-segment display schematic diagram. Each LED anode is connected to a separate pin (pins 1-2, 4, 6-7, 9-10). The cathodes are internally connected and can be connected to the circuit ground Vss by using either pin 3 or 8.

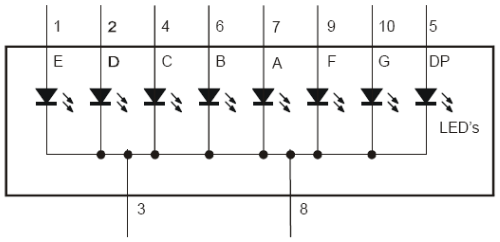


Fig 4.3: 7-Segment Display schematic diagram

Connect and Test the 7-Segment display Experiment

In this experiment you will connect the 7-Segment display and manually test each segment.
You will need 1x 7-Segment display, 5x 1kΩ resistors, 5x jumper wires.

- Set the 3-position switch on the BOE to position-0 to turn off the Power.

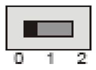


Fig 4.4: Power switch is off on 0 position

- Build the circuit shown in figures 4.5 and 4.6. “nc” means wire is not connected.

- Reconnect power and see what happens? Segment A is supposed to emit light.

____

If Segment A did not emit light, revise your connections and note what was wrong

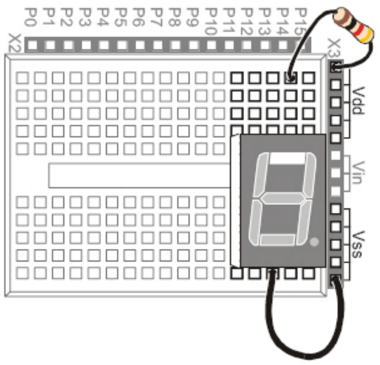


Fig 4.5: 7-Segment display test circuit (A Segment)

____
____
____
____
____
____
____
____
____
____
____

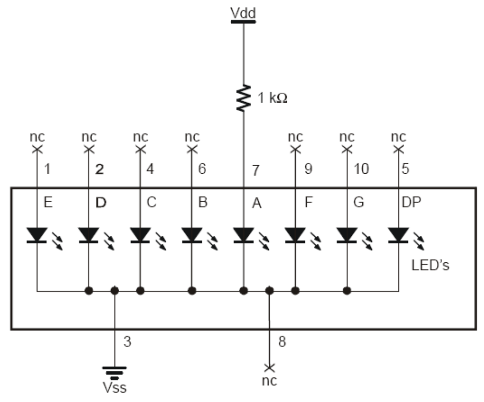


Fig 4.6: 7-Segment display test circuit schematic diagram

- Disconnect power, modify the circuit by connecting the resistor to the B Segment LED, Which pin number is that? _______ . if you need connection help check figures 4.7 and 4.8.

- Reconnect power; verify that Segment B emits light.

If Segment B did not emit light, revise your connections and note what was wrong.

____
____
____
____
____
____
____
____
____
____
____
____
____

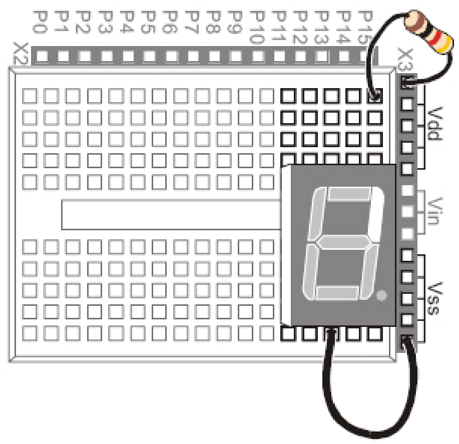


Fig 4.7: 7-Segment display test circuit (B Segment)

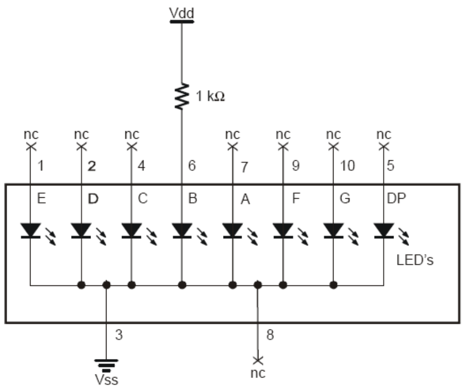


Fig 4.8: 7-Segment display test circuit schematic diagram

- Disconnect the power and repeat the same steps for the remaining segments. Fill in table 4.2

Segment	Pin No.	Done
C		
D		
E		
F		
G		
DP		

Table 4.2: Segments' test results

- Disconnect the power, and build the circuit shown in figures 4.9 and 4.10 to display number 3.

- Reconnect power, verify that it is number 3

If number 3 was not on, revise your connections and note what was wrong

____
____
____
____
____
____
____
____
____
____

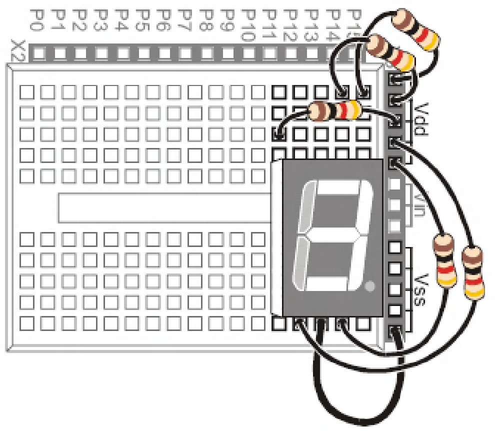


Fig 4.9: 7-Segment display test circuit (Number 3)

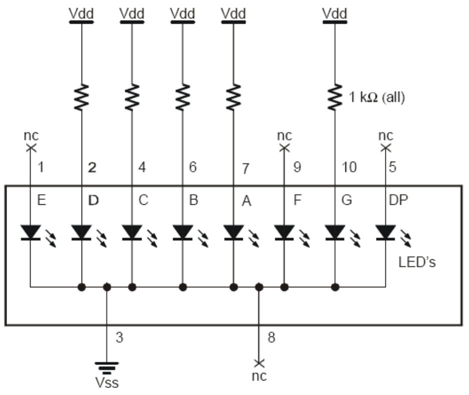


Fig 4.10: 7-Segment display test circuit schematic diagram (Number 3)

- Can you display Number 2?

- Draw the schematic diagram in the box below

- Connect the circuit and test your connection.

- Once you are done, disconnect the power.

Programming 7-Segment Display

If you include the decimal point there you need to connect eight different Basic Stamp I/O pins that send high/low signals to the 7-Segment LED display. That's eight different **HIGH** or **LOW** commands just to display one number. If you want to count from zero to nine, that would be a huge amount of programming. It is simple but lengthy.

There are other shorter ways to program the 7-Segment display and also as simple such as: DIRH and OUTH.
DIRH and OUTH are not commands, they are system defined byte variables that you can read and modify, and you don't need to declare them.

DIRH

DIRH is a variable that controls the direction (input or output) of the pins P8-P15.
Example:
DIRH= % 11110000
DIR is short for Direction,
    H is short for high P8-P15,
        it could be L short for low P0-P7
% means following number is a binary number
11110000: each bit corresponds to a pin in the following order:

```
%            1  1  1  1  0  0  0  0
Pin number: 15 14 13 12 11 10  9  8
```

1 means Output
0 means input

OUTH

OUTH is a variable that controls the status of each output pin (high or low) for the pins P8-P15.

Example:
OUTH= % 11110000
H is short for high (P8-P15),
it could be L short for Low (P0-P7)
11110000: each bit corresponds to a pin in the following order:

```
%              1  1  1  1  0  0  0  0
Pin number:   15 14 13 12 11 10  9  8
```

1 means high
0 means low

Basic Stamp Controlled 7-Segment Display via OUTH command Experiment

In this experiment, you will connect the 7-Segment LED display to the BASIC Stamp, and then run a simple program to test and make sure each LED is properly connected.

You will need 1x 7-Segment display, 8x 1kΩ resistors and jumper wires

- Disconnect the power
- Build the circuit shown in figures 4.11 and 4.12
- Write the following program, download and test:

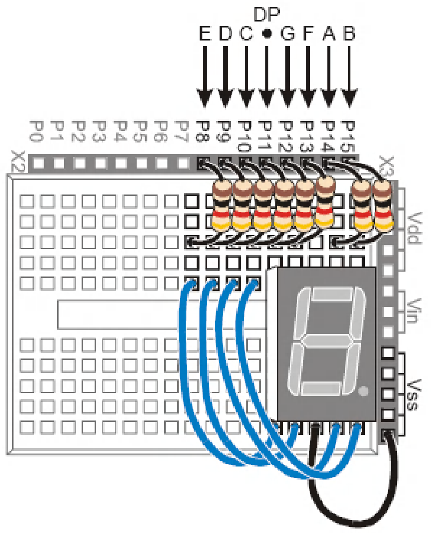


Fig 4.11: Basic Stamp controlled 7-Segment display circuit

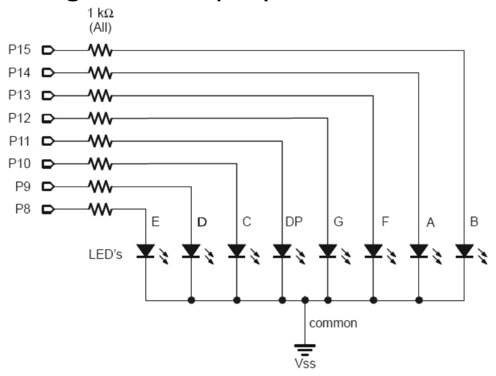


Fig 4.12: 7-Segment display schematic diagram

```
'{$STAMP BS2}
'{$PBASIC 2.5}

pinCounter     VAR     Nib

DEBUG "I/O Pin", CR,
      "-------", CR

FOR pinCounter = 8 TO 15

  DEBUG DEC2 pinCounter, CR
  HIGH pinCounter
  PAUSE 1000
  LOW pinCounter

NEXT
```

- Verify that every segment in the 7-Segment display lights briefly **on** then **off**. ________________________

- Fill in the Micro-Controller pin to segment connection table

Micro-Controller Pin	**Segment**
P8	
P9	
P10	
P11	
P12	
P13	
P14	
P15	

Table 4.3: Micro-Controller's pins connection segments

- Remove the command **"LOW** pincounter", and rerun the program. What is the difference?

____

____

____

- Write the following program, download, and analyze what every command will do.

Program | **Analysis**

```
DEBUG "Program Running!"
OUTH = %00000000
DIRH = %11111111
                ' BAFG.CDE
OUTH = %11100111
PAUSE 1000
OUTH = %10000100
PAUSE 1000
OUTH = %11010011
PAUSE 1000
OUTH = %11010110
PAUSE 1000
OUTH = %10110100
PAUSE 1000
OUTH = %01110110
PAUSE 1000
OUTH = %01110111
PAUSE 1000
OUTH = %11000100
PAUSE 1000
OUTH = %11110111
PAUSE 1000
OUTH = %11110110
PAUSE 1000
DIRH = %00000000
END
```

- Modify the program to display your name and test it

- Modify the program to display the letters A, b, C, d, E, and F and fill in table 4.4.

Letter	OUTH value
A	
b	
C	
d	
E	
F	

Table 4.4: A-F letter value.

LOOKUP and LOOKDOWN Commands

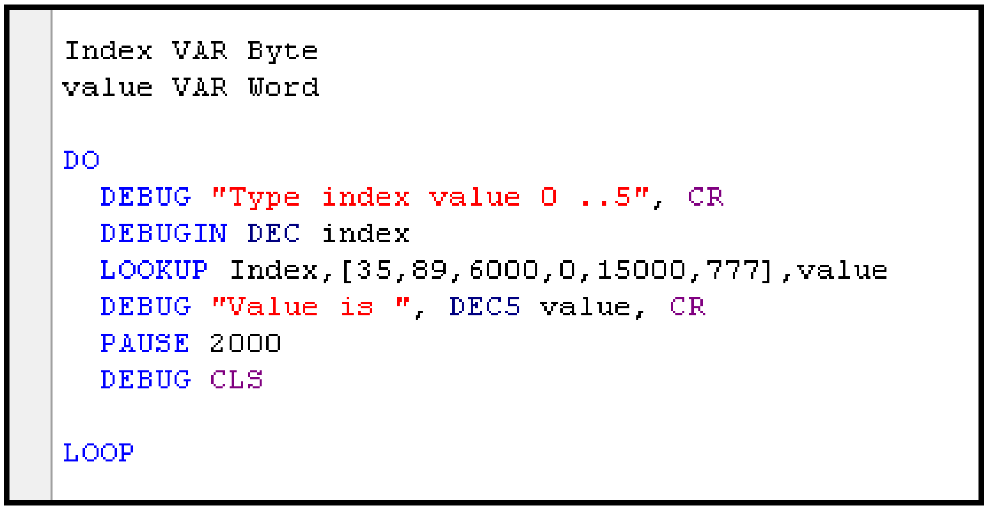
LOOKUP and LOOKDOWN commands also can be used to Program the 7-Segment display. In LOOK commands you can select an item from a list, or compare an item to list of items.

LOOKUP

Using this command you can retrieve an item from a pre-defined list based on that item's position (index) in the list.

Command Syntax:
LOOKUP Index, [item1, item2, …], Value
Index: it is a variable of maximum type byte to indicate the position of the required item in the list.
[item1, item2 …] is the list of items that could be of any variable type. The maximum number of items in a list =256 (0-255).
Value: it is a variable to store the required item from the list.

Example:



```
Index VAR Byte
value VAR Word

DO
  DEBUG "Type index value 0 ..5", CR
  DEBUGIN DEC index
  LOOKUP Index,[35,89,6000,0,15000,777],value
  DEBUG "Value is ", DEC5 value, CR
  PAUSE 2000
  DEBUG CLS

LOOP
```

Fig 4.13: LOOKUP command example

Once you type in the index value, the related item list will be stored in Value as follows:
Index = 0, value =35
Index =1, value = 89
Index =2, value = 0
…

Remarks:
- Pay attention that index starting number is **0**.
- If you give an index number exceeds the item numbers in a list, the last stored value will be displayed.

LOOKDOWN
LOOKDOWN command compares a value with items of a list and informs the position (index) of the first item that fits the comparison criteria.

Command Syntax:
LOOKDOWN value, Comparison [item1,I tem2 …], Index

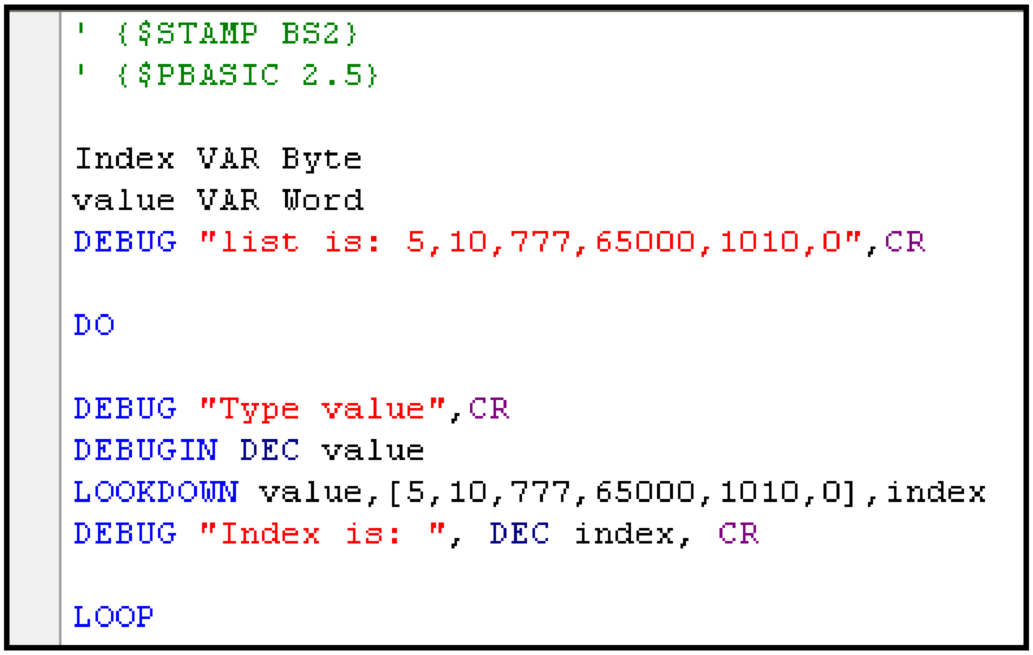
Value: is the variable to be compared to items in a list.

Comparison: is the comparison operation required, Let is be equal to (=) values, greater than (>), or less than (<).
When comparison left empty, it will be specified equal to (=) by default.

[item1, item2 …] is the list of items.

Index: it is the variable that will store the item’s position in the list. Every time the result will be one index value only, which is the index of the first item that matches the criteria, even if there is more than one match.

Example:



```
' {$STAMP BS2}
' {$PBASIC 2.5}

Index VAR Byte
value VAR Word
DEBUG "list is: 5,10,777,65000,1010,0",CR

DO

DEBUG "Type value",CR
DEBUGIN DEC value
LOOKDOWN value,[5,10,777,65000,1010,0],index
DEBUG "Index is: ", DEC index, CR

LOOP
```

Fig 4.14: LOOKDOWN command example

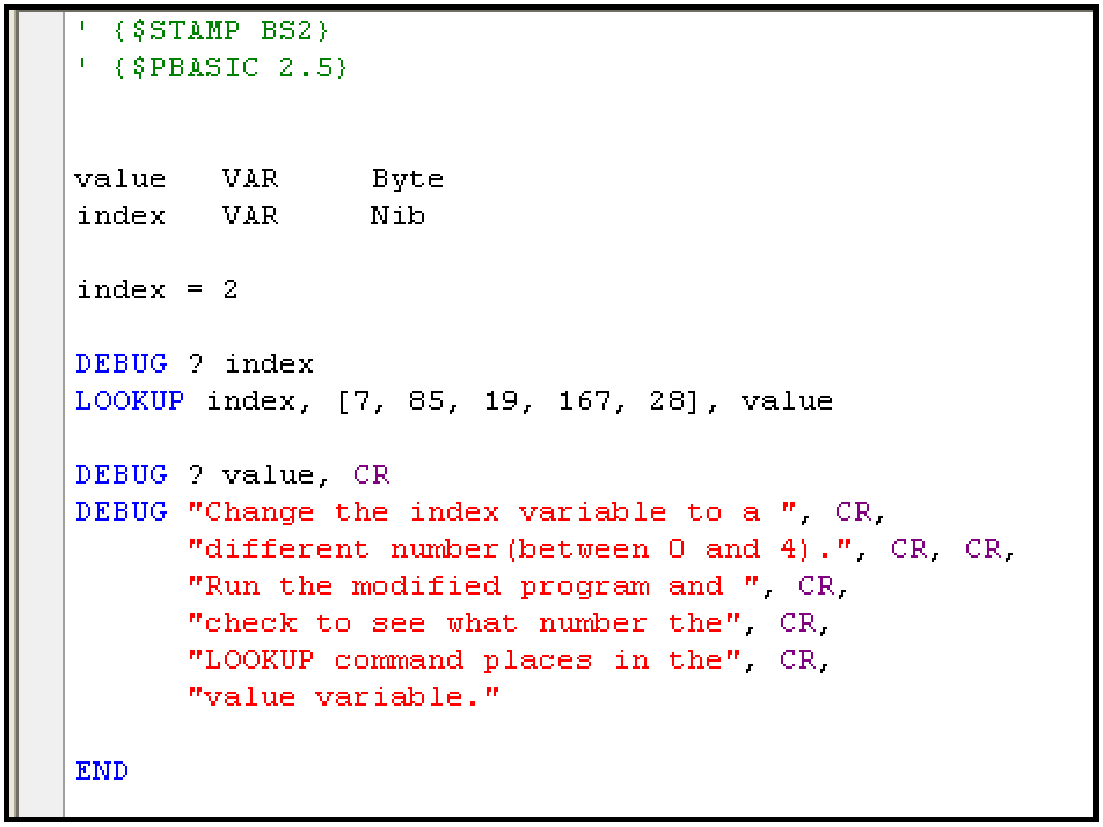
Once you type in the value, LOOKDOWN command will store the position of this value in the variable index. For Example:
value =5, Index= 0
value = 10, Index =1
value = 777, Index= 2
…
Since no comparison operator was mentioned it will be equal to (=) by default.

LOOK commands Test Experiment
In this part of the experiment you will simply test the functionality of LOOKUP and LOOKDOWN commands. All what you need is BOE only.

LOOKUP command experiment:

- Power on BOE

- Write and run the following program as is.



```
' {$STAMP BS2}
' {$PBASIC 2.5}


value   VAR     Byte
index   VAR     Nib

index = 2

DEBUG ? index
LOOKUP index, [7, 85, 19, 167, 28], value

DEBUG ? value, CR
DEBUG "Change the index variable to a ", CR,
      "different number(between 0 and 4).", CR, CR,
      "Run the modified program and ", CR,
      "check to see what number the", CR,
      "LOOKUP command places in the", CR,
      "value variable."

END
```

Fig 4.15: LOOKUP command experiment program

- When Index is 2, the value displayed on Debug Terminal Window is___________.

- Change the index number to any value between 0-5 and fill in the table.

Index	**Value**
0	
1	
2	
3	
4	
5	

Table 4.5: LOOKUP index- value table

- Do results match your expectations? _______

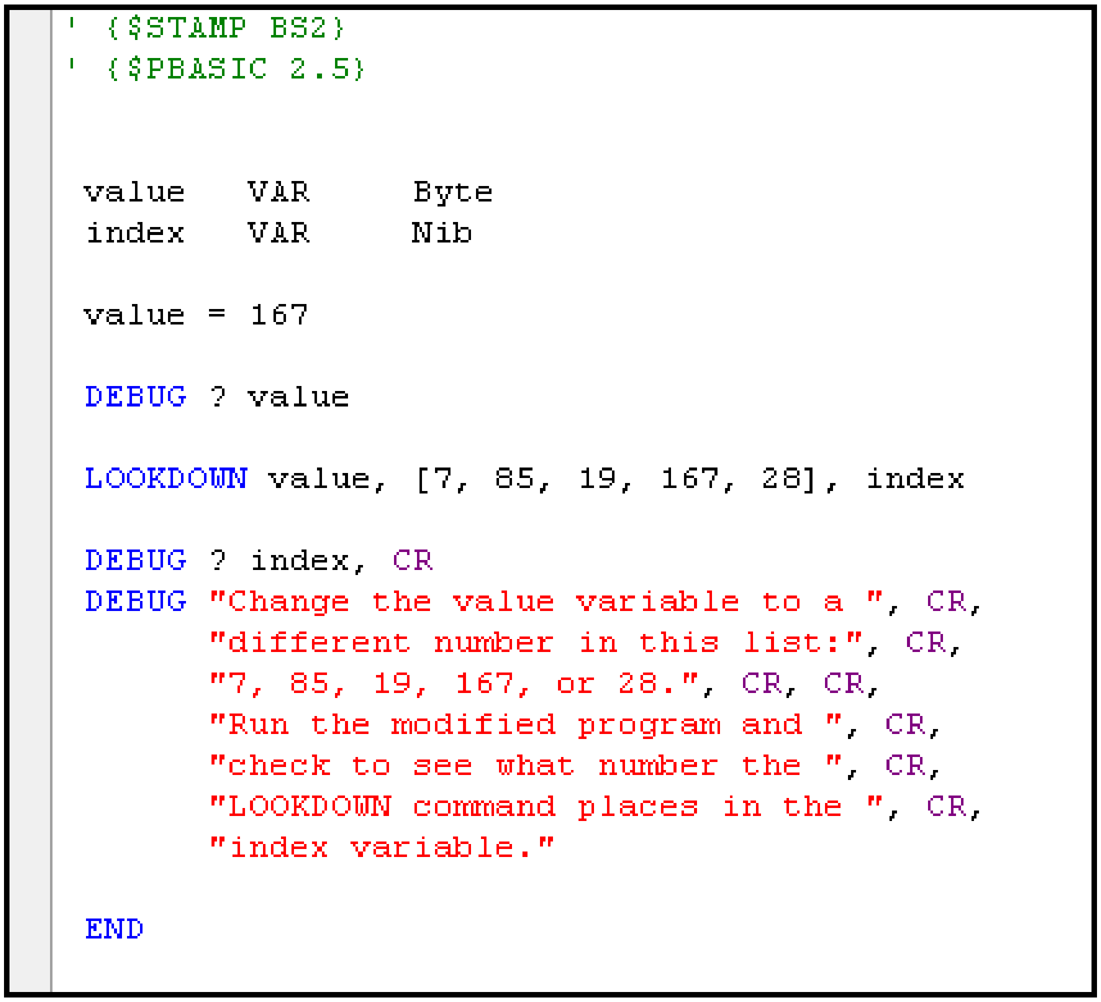
- Justify the result you got when index =5:

  ____

  ____

LOOKDOWN command experiment:

- Write and run the following program as is.



```
' {$STAMP BS2}
' {$PBASIC 2.5}


value   VAR     Byte
index   VAR     Nib

value = 167

DEBUG ? value

LOOKDOWN value, [7, 85, 19, 167, 28], index

DEBUG ? index, CR
DEBUG "Change the value variable to a ", CR,
      "different number in this list:", CR,
      "7, 85, 19, 167, or 28.", CR, CR,
      "Run the modified program and ", CR,
      "check to see what number the ", CR,
      "LOOKDOWN command places in the ", CR,
      "index variable."

END
```

Fig 4.16: LOOKDOWN command experiment program

- When value = 167, Index = ______.

- Change the value to 7, 85, 19, 28 and 168. Run the program after every change, and fill in table 4.6.

- Justify the result you got when Value was 168:

  ____

  ____

Value	Index
7	
85	
19	
28	
168	

Table 4.6: LOOKDOWN value-index table

- Modify the LOOKDOWN command as follows:
  LOOKDOWN value, <= [7, 19, 28, 85, 167], index, where value =35

  The index is _____ .

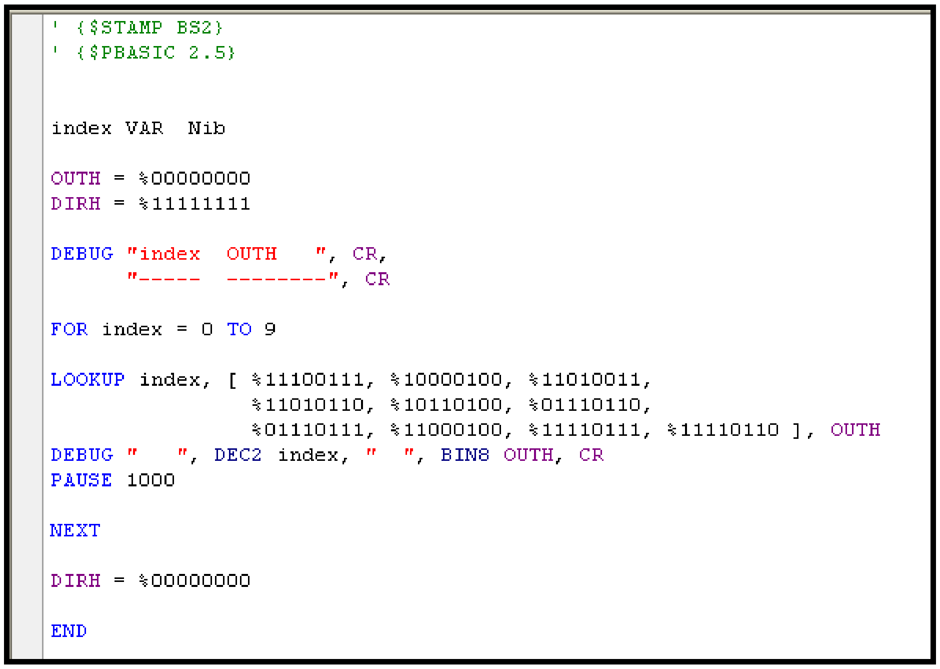
- Justify the result:

____
____
____
____
____
____
____

Programming 7-Segment Display with LOOKUP command experiment

In this experiment you will program 7-Segment display by using LOOKUP command instead of OUTH command, and realize the difference. You will need the same circuit you connected in the Basic Stamp Controlled 7-Segment Display via OUTH command Experiment in page 9.

- Write and run the following program:



```
' {$STAMP BS2}
' {$PBASIC 2.5}


index VAR  Nib

OUTH = %00000000
DIRH = %11111111

DEBUG "index  OUTH   ", CR,
      "-----  --------", CR

FOR index = 0 TO 9

LOOKUP index, [ %11100111, %10000100, %11010011,
                %11010110, %10110100, %01110110,
                %01110111, %11000100, %11110111, %11110110 ], OUTH
DEBUG "   ", DEC2 index, "   ", BIN8 OUTH, CR
PAUSE 1000

NEXT

DIRH = %00000000

END
```

Fig 4.17: 7-Segment Programming via LOOKUP command

- Verify that it displays digits 0 to 9 (with much less work).

- Take a look at the Debug Terminal while the program runs. It shows how the value of **index** is used by the **LOOKUP** command to load the correct binary value from the list into **OUTH**.

- Modify the program to display A, b, C, d, E and F letters.

- Once You are done, do the following:

1- Save your program.
2- Switch off the BOE power
3- Disassemble your circuit, and return the components to the safe.
4- Clean your working area and report any damaged components.