

# Beginners Guide to Programming

## “Lesson 3”

By Michael Simpson

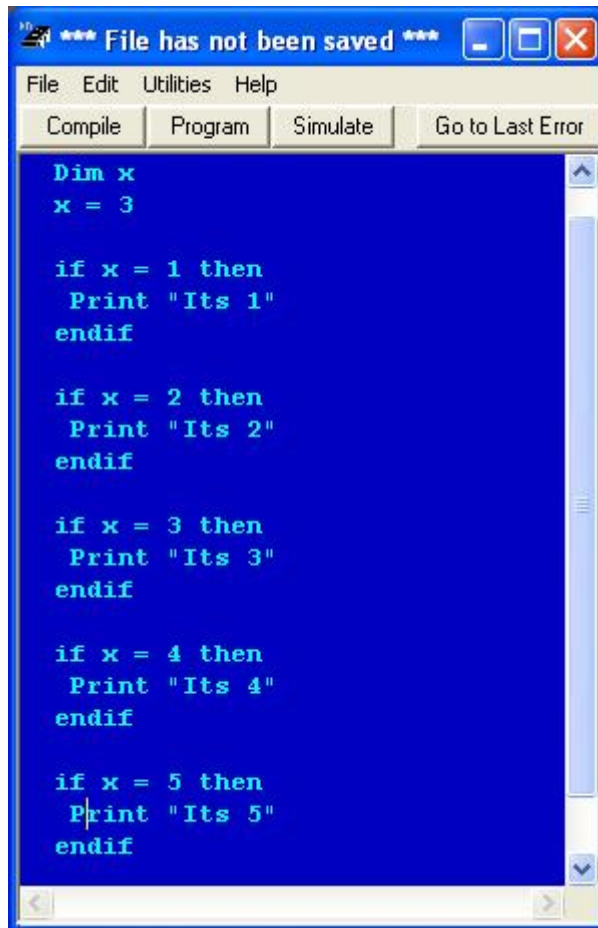
This is the final lesson in the “Beginners Guide to Programming”. It is important that you read and understand last month’s Lesson 2 before you proceed with this month’s lesson. We are going to cover the Branch, LookUp, LookDown and LCD commands.

Once you complete this lesson, you should have enough of an understanding of the Basic programming language to venture forth and explore the world of microcontrollers.

### Branch Command

Let’s say that you want to perform a set of actions based on a number between 1 and 5. One way to do this is with an If statement as shown in Figure 1. If you single step through the code you will notice that each condition is tested until a match is made. Even after a match is made, we continue to test until the end of the program is reached. This is not a very efficient way to perform this kind of action.

The Branch command will allow you to branch to a particular location in the code based on an index value. Take a look at the code in Figure 2. The Branch command uses the first argument as an index to point to one of the labels that you have provided. You supply a list of labels that align with the index. In this case, we used the labels do0, do1, do2, do3, do4, and do5. You might ask yourself why we had to use the do0 label. The Branch command is zero-based and the first label that you provide is the label that is jumped to if the index is 0. The next is 1, and then 2, and so on.



```
Dim x
x = 3

if x = 1 then
  Print "Its 1"
endif

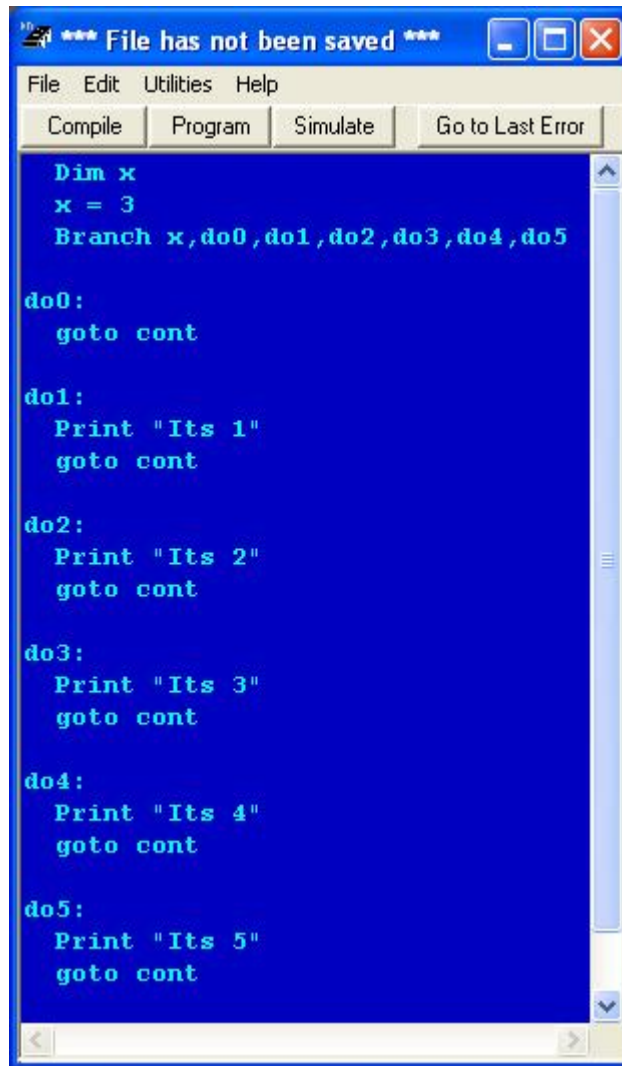
if x = 2 then
  Print "Its 2"
endif

if x = 3 then
  Print "Its 3"
endif

if x = 4 then
  Print "Its 4"
endif

if x = 5 then
  Print "Its 5"
endif
```

Figure 1



```
*** File has not been saved ***
File Edit Utilities Help
Compile Program Simulate Go to Last Error
Dim x
x = 3
Branch x, do0, do1, do2, do3, do4, do5

do0:
goto cont

do1:
Print "Its 1"
goto cont

do2:
Print "Its 2"
goto cont

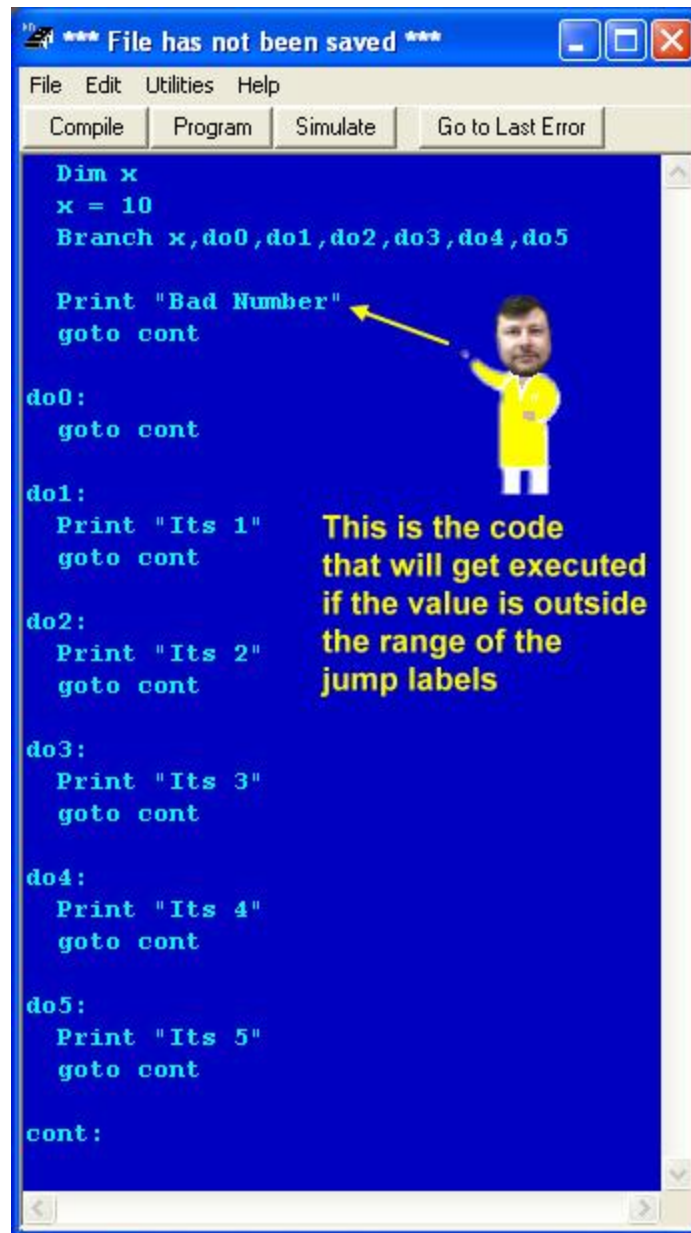
do3:
Print "Its 3"
goto cont

do4:
Print "Its 4"
goto cont

do5:
Print "Its 5"
goto cont
```

**Figure 2**

Now you might wonder what happens if the index falls outside the range of the provided jump labels. For instance, what would happen if the index value was 10? The Branch command is designed to fall through to the very next command if no label has been provided for the given index. The code in Figure 3 has been modified to handle numbers outside the range of the provided jump labels.



```
*** File has not been saved ***
File Edit Utilities Help
Compile Program Simulate Go to Last Error

Dim x
x = 10
Branch x, do0, do1, do2, do3, do4, do5

Print "Bad Number"
goto cont

do0:
goto cont

do1:
Print "Its 1"
goto cont

do2:
Print "Its 2"
goto cont

do3:
Print "Its 3"
goto cont

do4:
Print "Its 4"
goto cont

do5:
Print "Its 5"
goto cont

cont:
```

This is the code that will get executed if the value is outside the range of the jump labels

**Figure 3**

The Branch command is a very valuable command, and when programming microcontrollers you will find it very efficient. Only a single condition test is made no matter how many jump labels are provided. Using a little math you can manipulate the index so that larger values may be used with only a few jump locations. Later I will show you how to do some fancy jumping using the LookDown and Branch commands together.

In older versions of the Basic language, the Branch command was called OnGoto.

## Setting Simulator to Intermediate

From this point on you need to have the Athena Simulator Mode set to Intermediate. You do this by selecting Intermediate from the Mode menu as shown in Figure 4.

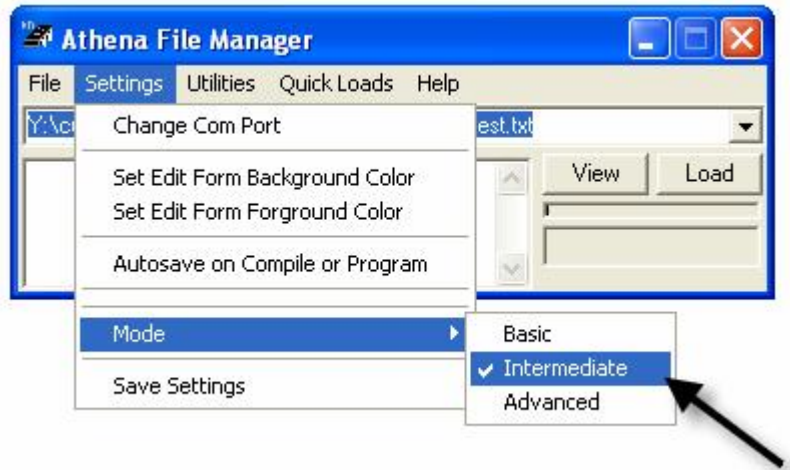


Figure 4

Once the mode is set, make sure you save your settings as shown in Figure 5.

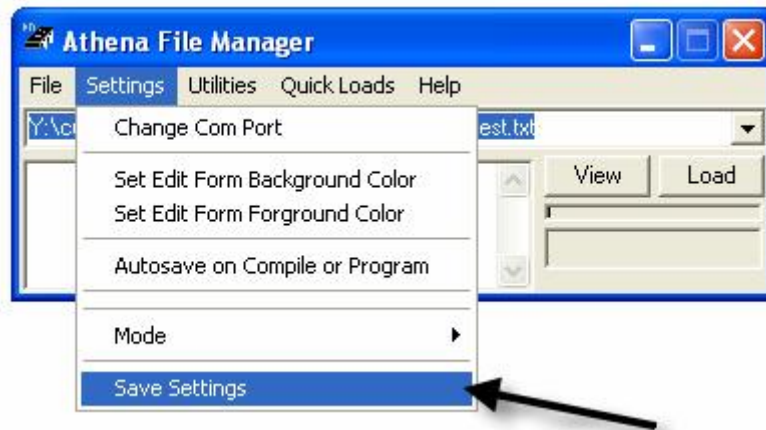


Figure 5

## LookUp Command

The LookUp command will allow you to create a lookup table. For instance, let's say you need to return a particular value based on a number between 0 and 10. Looking at Figure 6, you can see that the first argument is the result variable in which you wish to place the looked up value. The next argument is the actual index in the range of 0-255. You then supply a list of look up values. Each of these values maps to the index. For instance, the first lookup value maps to 0 and the next to 1 and so on.

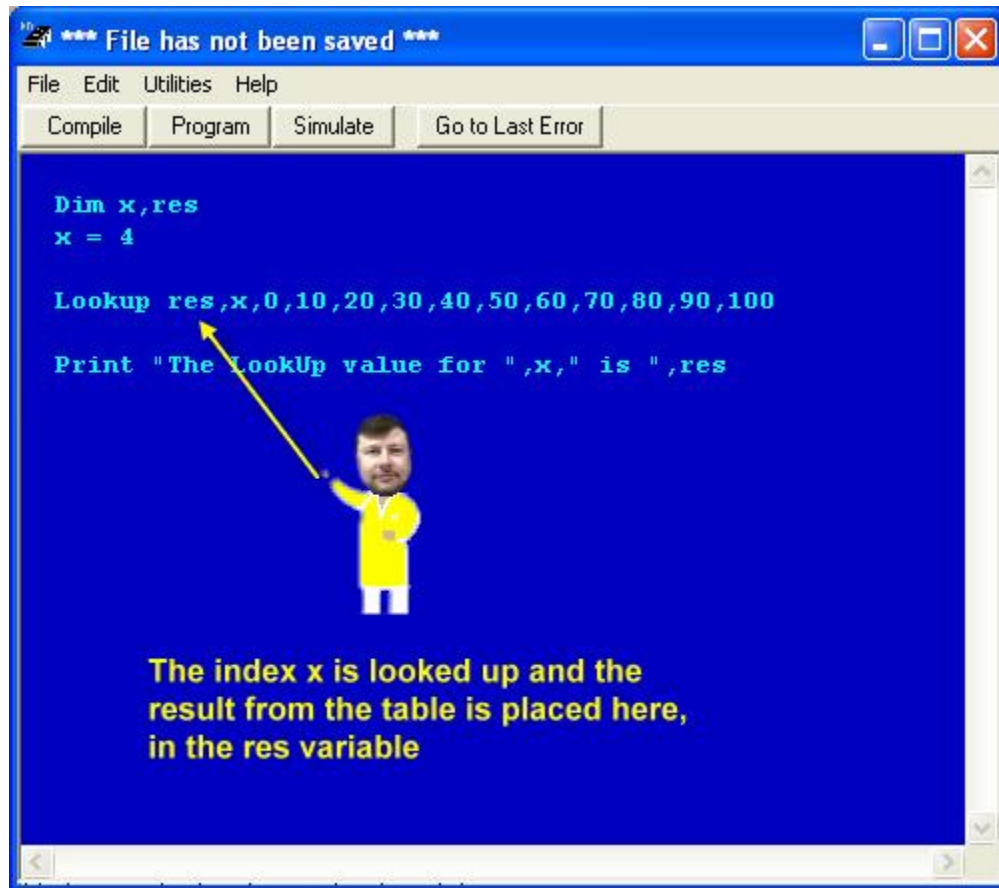


Figure 6

If a lookup value is not found for the index supplied, the original value will not be changed.

## Command Syntax Help

You can get full syntax help for any of the Athena commands by hitting F8 on any Edit Form. All the available commands for the current difficulty mode will be displayed on the left, as shown in Figure 7. When you select one of the commands the syntax, description, and a code example will be displayed on the bottom. The Paste button Will

paste the actual command shown in the Usage field into your code at the current cursor position.

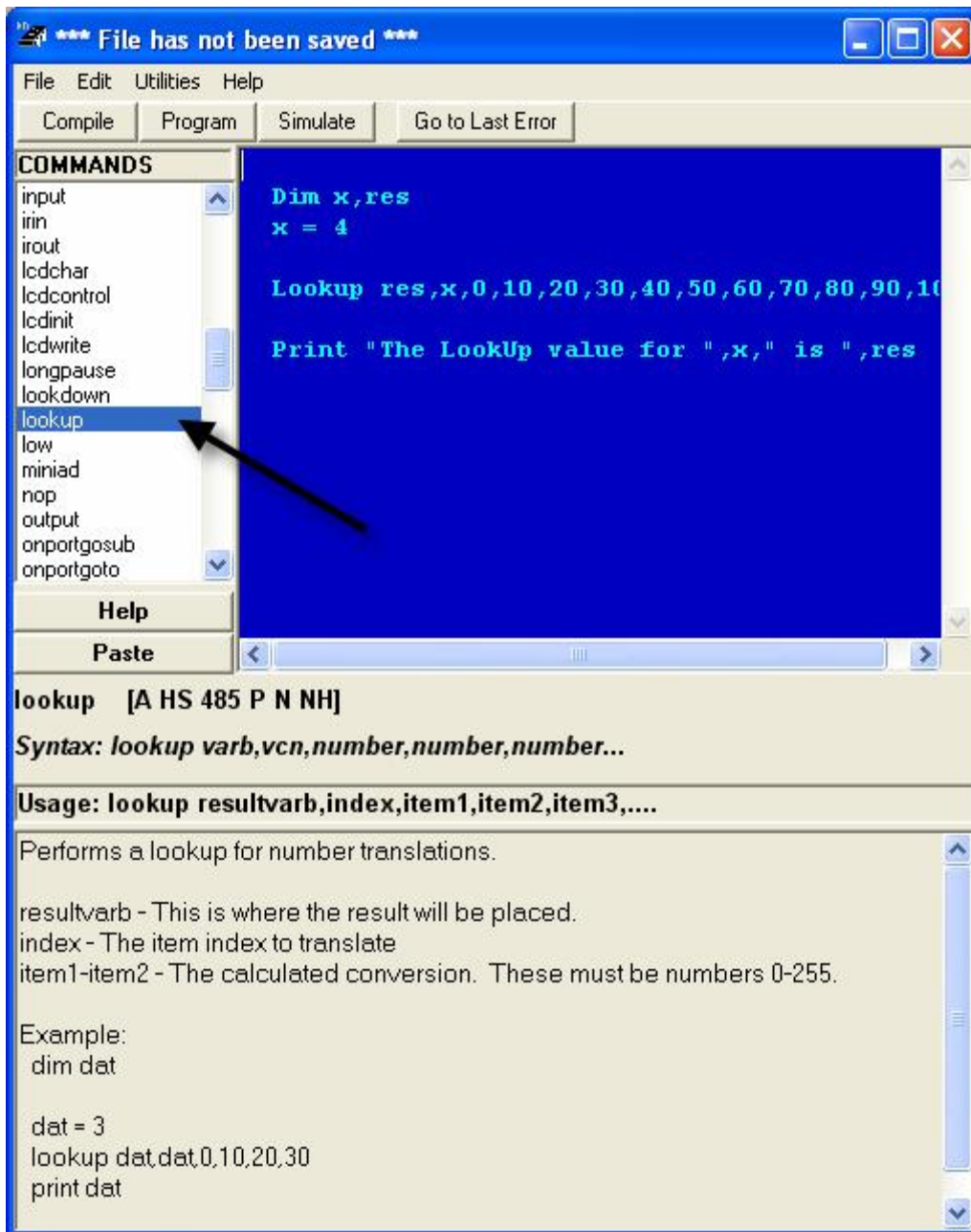
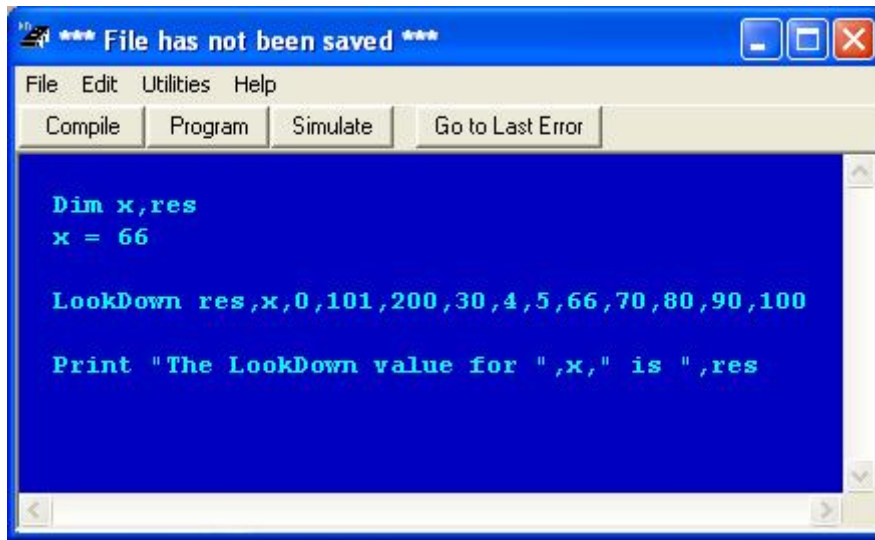


Figure 7

## LookDown Command

The LookDown command works much the same way as the LookUp command in the way that you supply the arguments; however it takes a second argument and performs a search against all the table entries. When it finds a match it returns the index of the item found. In the example shown in Figure 8 the 7th element is matched so a value of 6 is returned. Remember that the command is zero-based, so the first element will return a zero and the second will return one and so on.

A screenshot of a software window with a blue title bar that says "File has not been saved". The window has a menu bar with "File", "Edit", "Utilities", and "Help". Below the menu bar are four buttons: "Compile", "Program", "Simulate", and "Go to Last Error". The main area of the window is a blue text editor containing the following code:

```
Dim x,res
x = 66

LookDown res,x,0,101,200,30,4,5,66,70,80,90,100

Print "The LookDown value for ",x," is ",res
```

**Figure 8**

The real power of the LookDown command is when it is used with other commands like the branch command as shown in Figure 9. By using the LookDown command first, we can create an index off of any range of numbers. The index can then be used to perform a jump using the branch command.

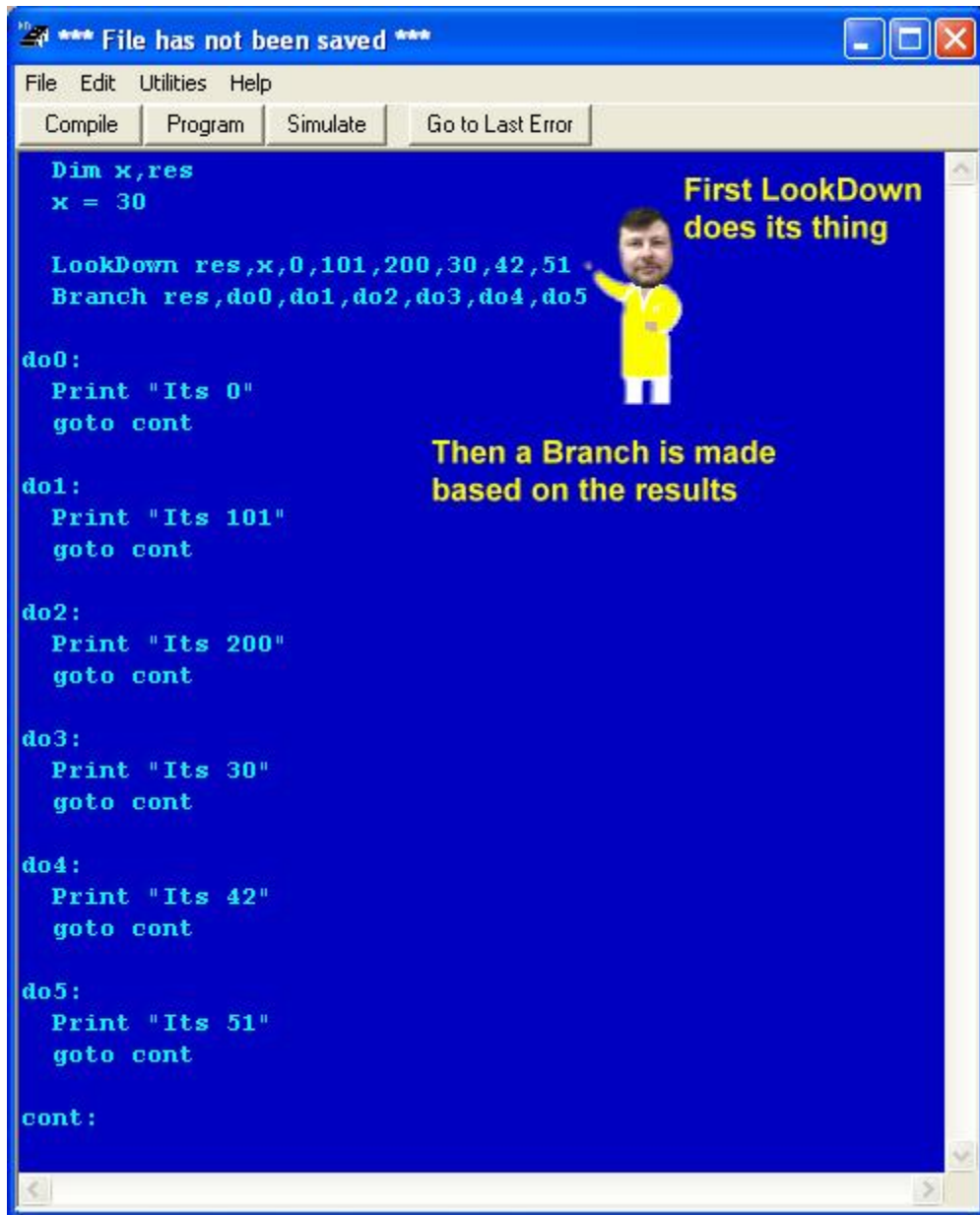


Figure 9

## LCD Commands

Many microcontrollers and compilers have support for Character LCDs and the Athena is no exception. First, in order to see the results of your LCD commands you need to activate the LCD in the simulator. This is done by selecting the Show LCD option in the Watch menu on the chip simulation form as shown in Figure 10.

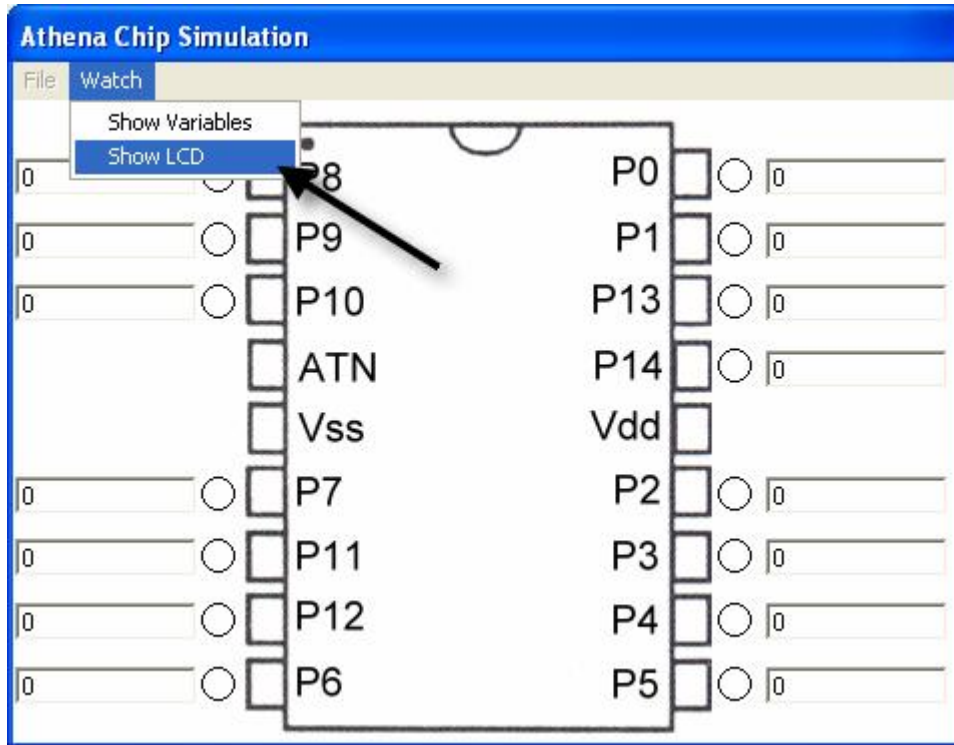


Figure 10

Once selected, a 16 x 2 character LCD will pop up as shown in Figure 12. This LCD is a very close simulation of a 16 x 2 LCD using a Hitachi 44780 controller. The memory mapping and control characters are all supported.

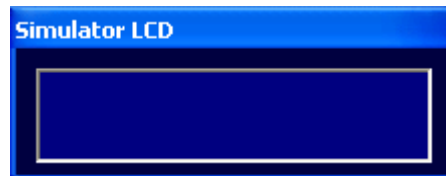


Figure 11

To use the LCD commands, you must first use the **LCDinit** command. This properly initializes the LCD. Once initialized, you have three commands that can be used to send data to the LCD.

### LCDchar

Sends a single character to the LCD

### LCDcontrol

Sends a special control code the LCD

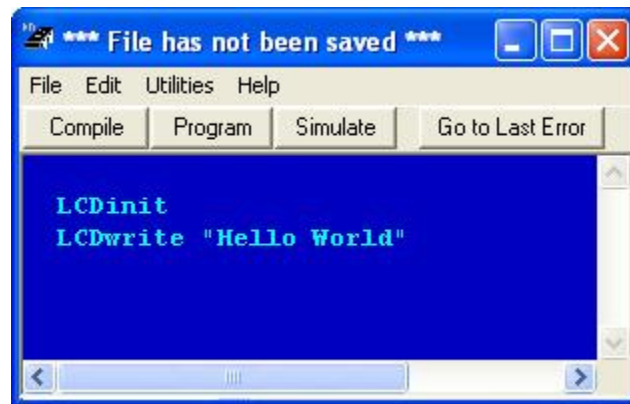
- 1 – Clear the Display
- 2 – Home the Display

- 12 – Cursor Off
- 14 – Cursor On, No Blink
- 15 – Cursor On, Blink
- 16 – Shift Cursor Left
- 20 – Shift Cursor Right
- 24 – Shift Display Left
- 28 – Shift Display Right

### **LCDwrite**

Sends a string of characters to the LCD

To send a simple message to the LCD you use the LCDwrite command as shown in Figure 12. This will display the message shown in Figure 13.

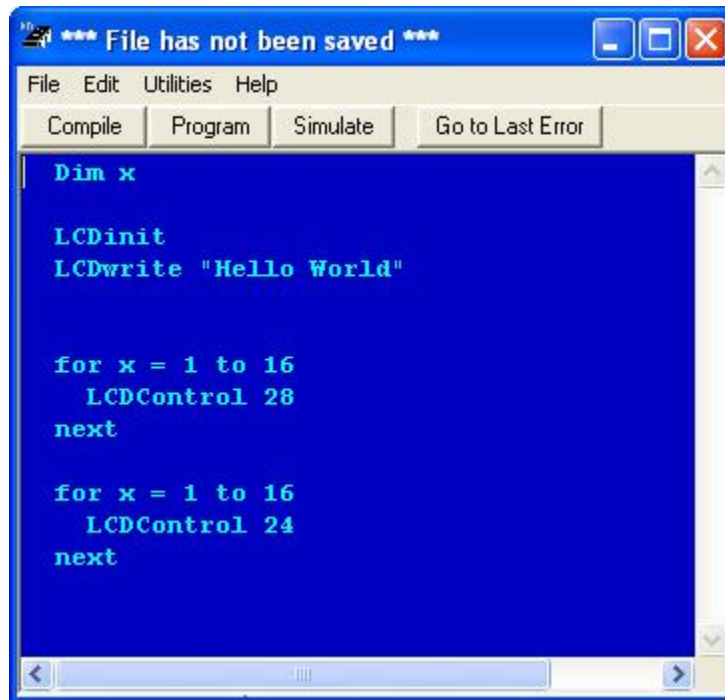


**Figure 12**



**Figure 13**

We can scroll the LCD display to the right then back to the left by utilizing a couple control codes. Load and run the code in Figure 14 to see the effect.



```
Dim x

LCDinit
LCDwrite "Hello World"

for x = 1 to 16
  LCDControl 28
next

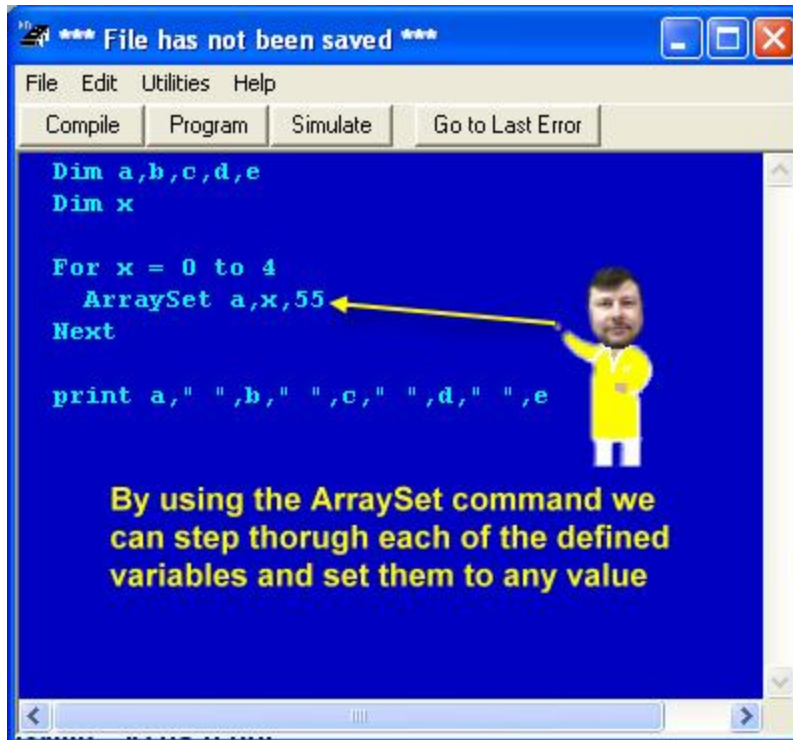
for x = 1 to 16
  LCDControl 24
next
```

**Figure 14**

### **ArrayGet and ArraySet Commands**

In more advanced dialects of the Basic language, you have access to variable arrays. While the Athena Simulator does not have access to such variables, it does have a couple of commands that will let you access declared variables as a group.

When you create variables in your code, they are stored in contiguous locations in memory. When you use the Dim command to create the variables a, b, c, d, and e, they are each stored in memory, one after another. The ArrayGet and ArraySet commands take advantage of this and let you access the variables by using only the first variable as a reference.



**Figure 15**

The code in Figure 15 is using the ArraySet Command to set the variables a, b, c, d, and e to a value of 55. The first argument is the reference variable. This is the first variable in the list that you wish to manipulate. The second argument is the index. If the value of this index is 0, then the variable a will be accessed. If the value is 1, then the variable b is accessed, and so on.

The ArrayGet works much the same way as the ArraySet command. Instead of passing a value to set, we pass a variable that will hold the result of the indexed variable.

```
Dim a,b,c,d,e
Dim x,res

For x = 0 to 4
  ArraySet a,x,55
Next

For x = 0 to 4
  ArrayGet a,x,res
  print res
Next
```

Figure 16

## Programming a Real Athena

This month we completed the final lesson in the “Beginners Guide to Programming”. The next logical step is to actually program a microcontroller. The Athena line of microcontrollers was created for the sole purpose of getting you started programming a microcontroller. The Athena microcontroller shown in Figure 17 is a perfect device with which to start. KronosRobotics sells them for only \$6.95 each. The Perseus microcontroller is the smallest but it packs quite a bit of power in that small package. It features AtoD and runs at twice the speed of the Athena. The Nemesis is the Flagship of the Athena line and allows you to perform inline assembly language, so it’s a perfect way to take the next step in programming. All three chips use the Basic language featured in Lessons 1, 2, and 3.

In order to program the chips, you need an RS232 driver in order to interface the chip to your PC. There are application notes on the KronosRobotics web site that show you how to build one from scratch. You can also purchase an EZRS232 driver shown in Figure 18 for under \$10.

You have the choice of using the chips on a breadboard or prototype board or you can use one of the many carrier boards available from KronosRobotics. The smaller carriers like the ones shown in Figure 19 are great for small dedicated robot projects. The flagship carrier board, called the Athena WorkBoard Deluxe shown in Figure 20, can be used with any of these chips and includes a breadboard, program interface and power supply.



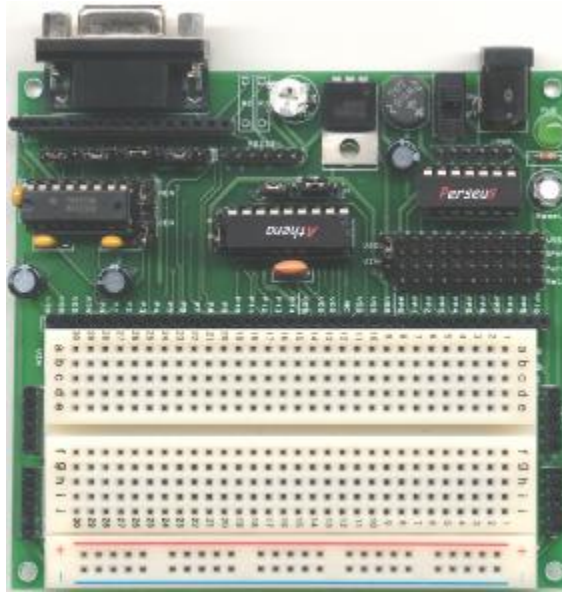
**Figure 17**



**Figure 18**



**Figure 19**



**Figure 20**

If you do decide to program an actual microcontroller you can start by running through the lessons I presented, or you can follow the tutorial included with the compiler download. Make sure you visit the kronosrobotics website at [www.kronosrobotics.com](http://www.kronosrobotics.com) for various tutorials and application notes. You will find that we have not even scratched the surface of all the available commands, many of which are designed to interface with other devices or sensors.

## **Summary**

### **Final Lesson Questions**

1. Create a program using the Branch command that will take a value from 0 to 3 and convert that to 0, 21, 22, and 23, respectively, into a variable called res.
2. Create a program using the Lookup command that will take a value from 0 to 3 and convert that to 0, 21, 22, and 23, respectively, into a variable called res.
3. What shortcut key is used to activate command lists built into the edit form?
4. If you want to use the Branch command to execute various code segments based on non contiguous values, what other command can be used to help?
5. Create a program using both the LookDown and Branch commands that executes a segment of code based on the values 100,200,50,75, and 90.

6. What command must be issued before you can start using an LCD?
7. What command is used to clear the LCD display?
8. Create a program that flashes your name on an LCD.

## **Answers**



Remember, you can pick up a free copy of the Athena compiler/simulator at:

[www.kronosrobotics.com](http://www.kronosrobotics.com).

The program installs a complete manual and several sample programs.