

Technical Capabilities of the MicroLogix 1100 Embedded LCD



Today's low-end controller marketplace is more competitive than ever. Thriving in such an environment means using the best tools and technologies the world has to offer. All over the globe, companies requiring compact controllers look to the Allen-Bradley MicroLogix™ family from Rockwell Automation.

The MicroLogix family offers a breadth of controllers that satisfies the widest variety of applications. This ensures you'll find a world-class micro-controller that fits your application as easily as one that fits your budget.

The newest member to the family, the MicroLogix 1100, provides many of the same features you've come to expect from the MicroLogix family. With outstanding flexibility and enhanced networking capability, the MicroLogix 1100 is meant to complement the coverage area of MicroLogix controllers at an unbeatable price.

Each of the MicroLogix 1100 programmable controller supports sixteen digital I/O (10 inputs and 6 outputs), and two analog inputs (0 to 10V, 10 bit resolution); and then may be expanded using the same 1762 I/O platform as the MicroLogix 1200. Up to four 1762 I/O modules, can be used with a single MicroLogix 1100 controller.

Standing apart from the rest of the MicroLogix family, the MicroLogix 1100 is the first to support online editing utilizing the latest version of the RSLogix 500 programming software. Increase your production efficiency and speed troubleshooting by editing applications while they are running, without having to shut down the entire system.

In addition, an embedded 10/100 Mbps EtherNet/IP port for peer-to-peer messaging offers you high speed connectivity between controllers and the ability to access, monitor and program from the factory floor to anywhere an Ethernet connection is available. An embedded Web server allows for system parameters to be adjusted from afar and displayed as a Web page.

Finally, a Liquid Crystal Display (LCD) on every MicroLogix 1100 controller allows for limited user interaction and control. The remainder of this document will provide an in-depth look at the features and functionality of this embedded LCD.

Features

- 4x12 multipurpose text display
- User defined custom start-up screen
- Observe communication channel activity
 - Channel 0 Combo RS-232 / RS-485 port
 - Channel 1 EtherNet/IP port
 - Monitor the controller's IP address
- Toggle between default (factory) communication settings and user defined settings for communication trouble shooting or to provide an easy means to monitor or modify your program when configured for ASCII or Modbus RTU Master/Slave
- Battery health status
- Monitor controller operating mode and embedded digital I/O status
 - 10 digital inputs
 - 6 digital outputs
- Monitor and change controller mode, with local override capability at start-up (Run or Program)
- Monitor controller error / fault codes
- Monitor controller
- Supervise and modify controller data
 - Two digital trim pots
 - User configured minimum and maximum values
 - Unlike analog trim pots, values do not change over time or with temperature
 - 48 bits, each with individual protection
 - Monitor and modify bits that can be used by your control program
 - 48 integers, each with individual protection
 - Observe and change timer and counter values, setpoints, etc.
- New LCD Instruction
 - RSLogix 500 Version 6.4 and above
 - Controller program can send combinations of string, integer and bit data to LCD to communicate with an operator
 - Request data from operator to provide feedback to the control program

MicroLogix 1100 LCD Functionality

Through the embedded LCD, your MicroLogix 1100 lets you monitor bit and integer data within the controller, and optionally modify that data, to interact with your control program. Similarly to the optional 1764-DAT for the MicroLogix 1500 controllers, the embedded MicroLogix 1100 LCD allows users access to 48 bits and 48 integers, each of which can be individually protected. Need to know the speed of a conveyor, the status of a remote sensor, or how close your process is running relative to its optimal temperature? Just monitor your LCD. Need to manually start an operation, change a timing sequence, or make adjustments to a counter? Why use your laptop and programming software when the LCD allows you to simulate pushbuttons or numeric entry devices? By simply moving or copying data in and out of the bit and integer files, you now can monitor and modify the parameters that your controller uses.



Making use of the new MicroLogix 1100 "LCD Instruction", your controller can directly interface with a local operator using your ladder logic. The LCD Instruction executes under two modes of operation, the first mode being ladder logic output to the display only. In this mode, up to three lines of data, with up to 12 characters per line, can be sent to the display from the ladder logic running in the controller. Think of this as simply messaging to the LCD. These lines can consist of combinations of Bits, Integers, and String characters. So now the control program can send alert / alarm messages, I/O data values, simple text messages, or combinations of these messages to the operator. These messages can be triggered by events (input sensors, timer "done bits", message from another controller, etc.), or based on a schedule (using the embedded real time clock, or free running timers).




The second mode of operation again allows for output from the ladder logic to the display, but adds input from the operator back to the controller. Up to two lines of up to 12 characters each can still be sent to the LCD for display, but the third line, in this mode, is used to obtain numeric input from the user. Bit, Integer, or Long Integer file types can be used to provide this input.



The user can select 'User Display' from the LCD menu. The User Display screen will show the specified output data when the LCD Instruction is energized. If "DISPLAY WITH INPUT" is set to "YES", the user can enter input using the LCD keypad to enter Bit, Integer, or Long Integer data. Examples at the end of this whitepaper illustrate how the data exchange can be used.

The LCD Instruction is organized as a "Function File". Function Files organize similar controller functions into simple structures for ease of use. The Function File for the LCD appears below:

HSC	PTO	PWM	STI	EII	RTC	LCD	MMI	BHI	CS0	CS1	IOS	
-----	-----	-----	-----	-----	-----	------------	-----	-----	-----	-----	-----	---

Address	Value	Description
LCD:0	{...}	
CBS - Customized Boot Message String File	0	
SCD - Start with Customized Display	0	
TO - Data Input Timeout of LCD Instruction (x Sec)	0	
DN - LCD Instruction Job Done	1	
ERR - LCD Display Operation Error Bit	0	
ERN - LCD Module Operation Error Number	0	
TBF - Target Bit File Number	0	
TIF - Target Integer File Number	0	
JOG - Jog data update Mode Set	0	
TMIN - Trimpot Low Value	0	
TMAX - Trimpot High Value	250	
POT0 - Trimpot 0 Data (TMIN - TMAX)	0	
POT1 - Trimpot 1 Data (TMIN - TMAX)	0	
WND - Instruction Display Window	0	
OK - OK key in Customized Display Mode	0	
ESC - ESC key in Customized Display Mode	0	

LCD Function File Setup Information:

Read/Write items: JOG, SCD, CBS, TO, TBF, TIF, TMIN, TMAX, OK, ESC

Read-Only items: DN, ERR, ERN, POT0, POT1, WND

Boolean value items: JOG, SCD, DN, ERR, OK, ESC, WND

Integer value items: CBS, TO, ERN, TBF, TIF, TMIN, TMAX, POT0, POT1

Element	Bit		Description
3	-	CBS	Customized Boot Message String File Address Offset
0	5	SCD	Start with Customized Display
4	-	TO	Data Input Timeout of LCD instruction (xSec)
1	0	DN	LCD Instruction Job Done
1	1	ERR	LCD Display Operation Error Bit
5	-	ERN	LCD Module Operation Error Number
6	-	TBF	Target Bit File Number
7	-	TIF	Target Integer File Number
0	3	JOG	Jog data update Mode set
8	-	TMIN	Trim pot low value
9	-	TMAX	Trim pot high value
10	-	POT0	Trim pot 0 Data (TMIN – TMAX)
11	-	POT1	Trim pot 1 Data (TMIN – TMAX)
12	-	OK	OK key in Customized Display
13	-	ESC	ESC key in Customized Display
14	-	WND	Instruction Display Window

These elements allow you to monitor and change the functionality of the LCD using programming software. But these elements are also available to the ladder logic running in your controller, so changes in your control environment can be reflected in the operation of the LCD. There are a number of examples at the end of this whitepaper that illustrate a few of the many ways that you can make use of this tool.

Applications

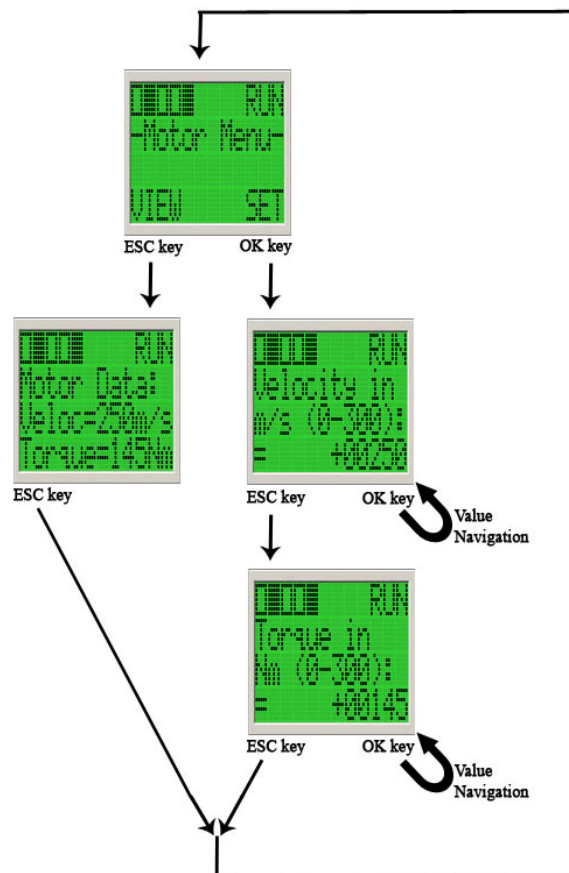
Start/Stop inputs – Use the LCD Bits as simulated real world inputs

- Control machines, drives, conveyors, processes, etc.
- Numeric entry and display – Use the integers to modify controller data
 - Manage speed, distance, position, pressure, etc.
- Perform simple calculations
 - Implement password protection for critical ladder logic sections
 - Date/Time entry and display



Multi-screen alarm messages using multiple LCD Instructions and timers

- Control program navigation
 - Via integer/bit input and the Jump To Subroutine Instruction
- Custom user created menus controlled by ESC/OK keys
 - Ladder logic determines when ESC/OK keys are pressed and Jump Instructions are used to navigate to corresponding ladder logic



The LCD also provides support for troubleshooting field issues. The MicroLogix 1100 LCD supports the ability to toggle from the default factory communications settings, to user configured communication settings, and back through the LCD keypad. So if your controller is configured for Modbus RTU Master/Slave or ASCII, and you need to make a program change, you can simply and easily: reconfigure communications back to factory defaults; make any changes necessary; and return to your configured communication settings.

The LCD can be used to force the controller into Run or Program mode, so the controller remains in the selected mode (unaffected by remote communications), or set to Remote operating mode, so that switching in and out of Run mode can be facilitated through remote communications.

The controller can display error codes on the LCD when a fault is encountered, allowing you to quickly determine the cause.

The LCD can also be used to display battery health status. So when the long life of the replaceable battery is coming to an end, the LCD can alert the operator well in advance.

The MicroLogix 1100's digital trim pots, that can be monitored and modified using LCD, never change with aging or temperature changes.

The MicroLogix 1100 embedded LCD was designed to add functionality and value to the controller. While it may provide the interface required for many applications, we recognize that it is very limited relative to a true HMI (human machine interface) device. But when you combine the benefits of the LCD with the functionality and dependability of this MicroLogix controller, your limits may only be bounded by your imagination. Please review the examples below for help with understanding the capabilities, and the implementation of LCD based features.

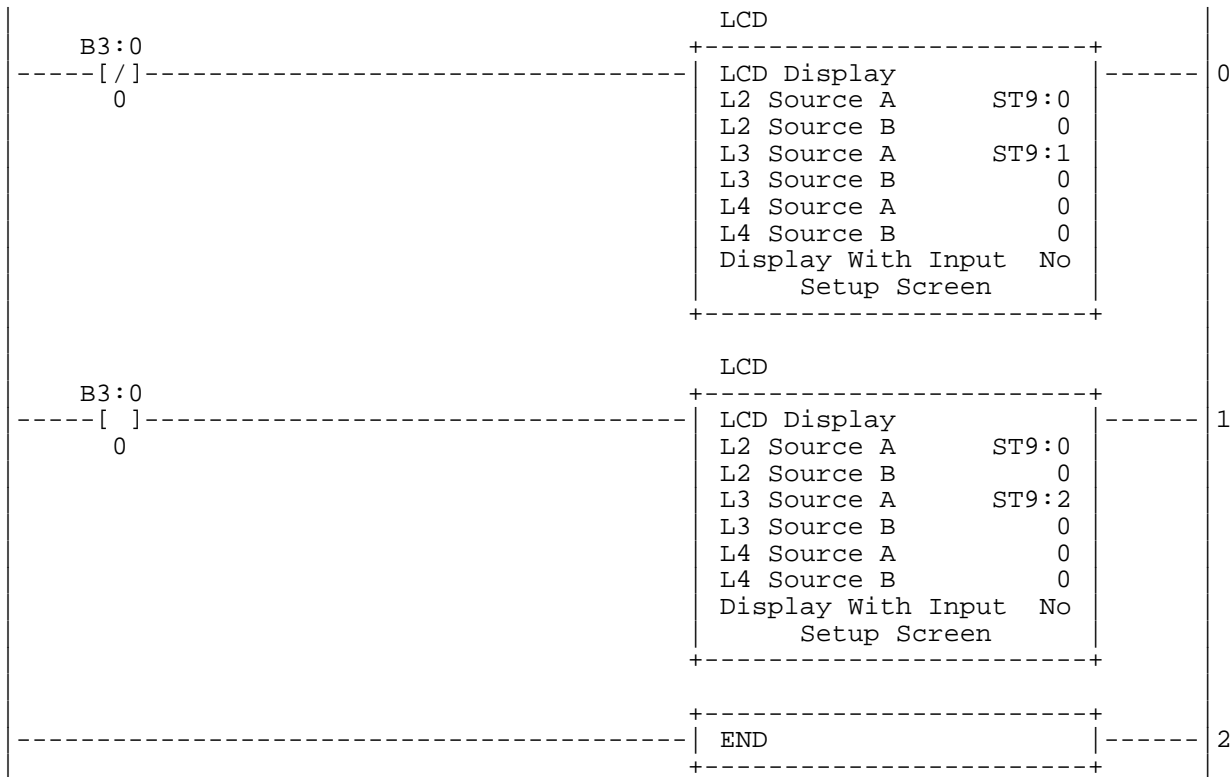
Example Code LCD 1

This is an example of string output on the LCD window based on the **value of a bit element**.

String Data File Values

Data File	Initial Value	Description
ST9:0	"B3:0/0 is"	Line 2 string definition
ST9:1	"De-energized"	Line 3 string definition 1
ST9:2	"Energized"	Line 3 string definition 2

Ladder Diagram



LCD Messages

If B3:0/0 is ON:

**B3:0/0 is
Energized**

If B3:0/0 is OFF:

**B3:0/0 is
De-energized**

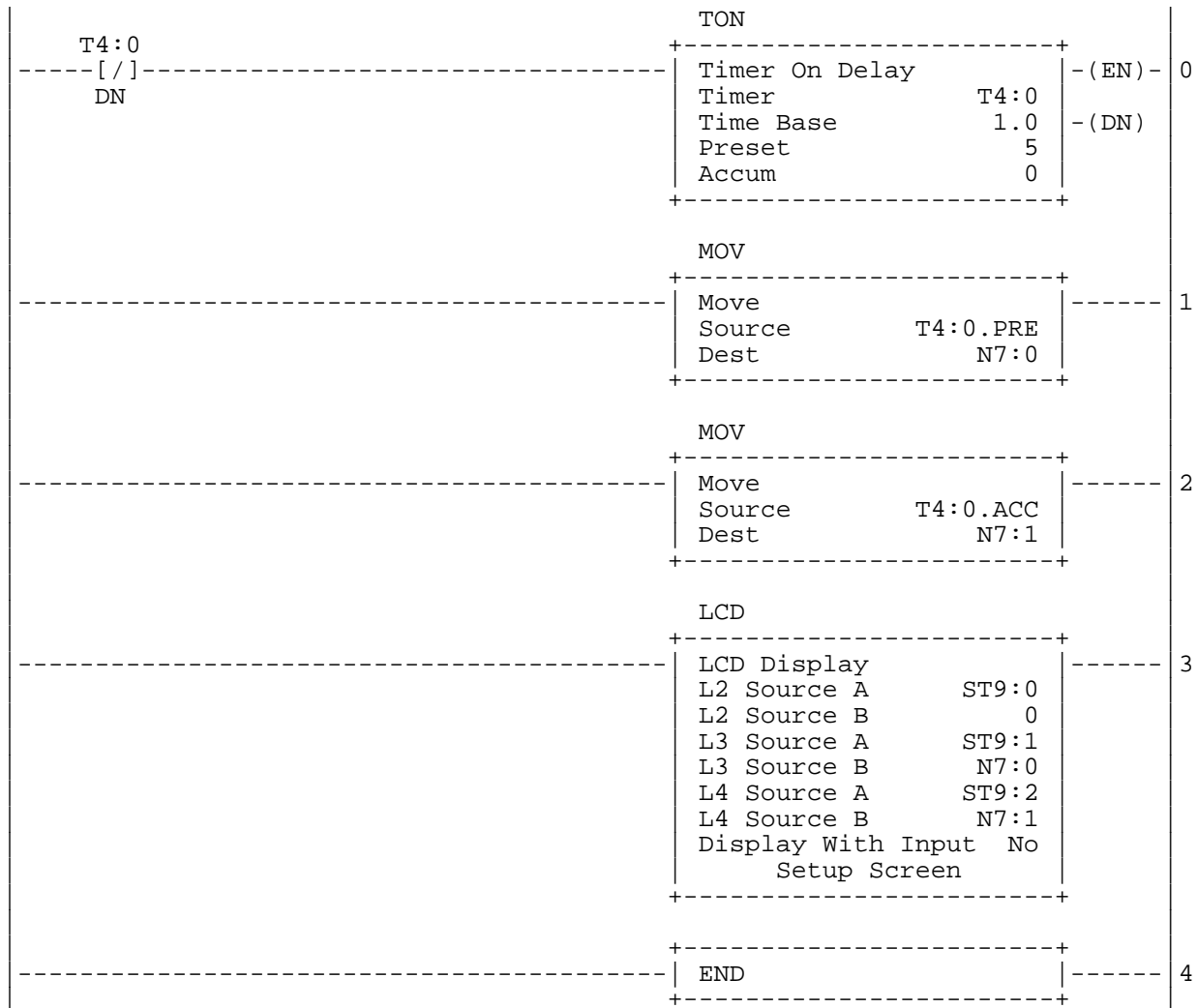
Example Code LCD 2

This is an example of string and data output on the LCD window based on Timer Data.

String Data File Values

Data File	Initial Value	Description
ST9:0	"T4:0 Info"	Line 2 string definition
ST9:1	"Preset="	Line 3 left string definition
ST9:2	"Accum="	Line 4 left string definition

Ladder Diagram



LCD Operation

Accum value is changing every 1second. Accum=0,1,2,3,4,0,1,2,3...

T4:0 Info
Preset=5
Accum=0

T4:0 Info
Preset=5
Accum=1

...

T4:0 Info
Preset=5
Accum=4

...

Example Code LCD 3

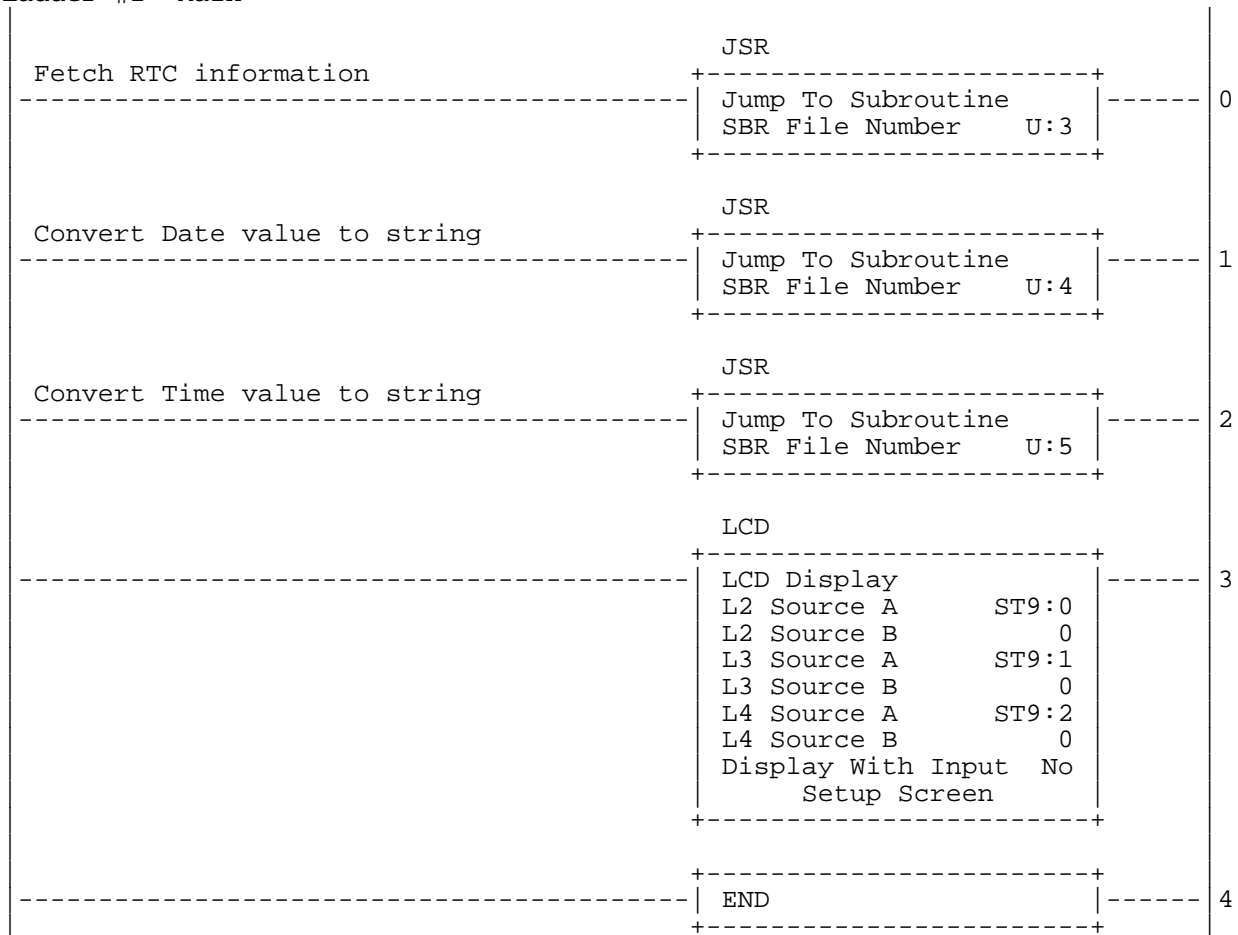
This is an example of string and data output on the LCD window to display Real Time Clock time and date.

String Data File Values

Data File	Initial Value	Description
ST9:0	"Date & Time"	Line 2 string definition
ST9:1	""	Line 3 string
ST9:2	""	Line 4 string
ST9:3	"-"	separator character
ST9:4	""	temporary register for Line 3 working
ST9:5	""	temporary register for Line 4 working

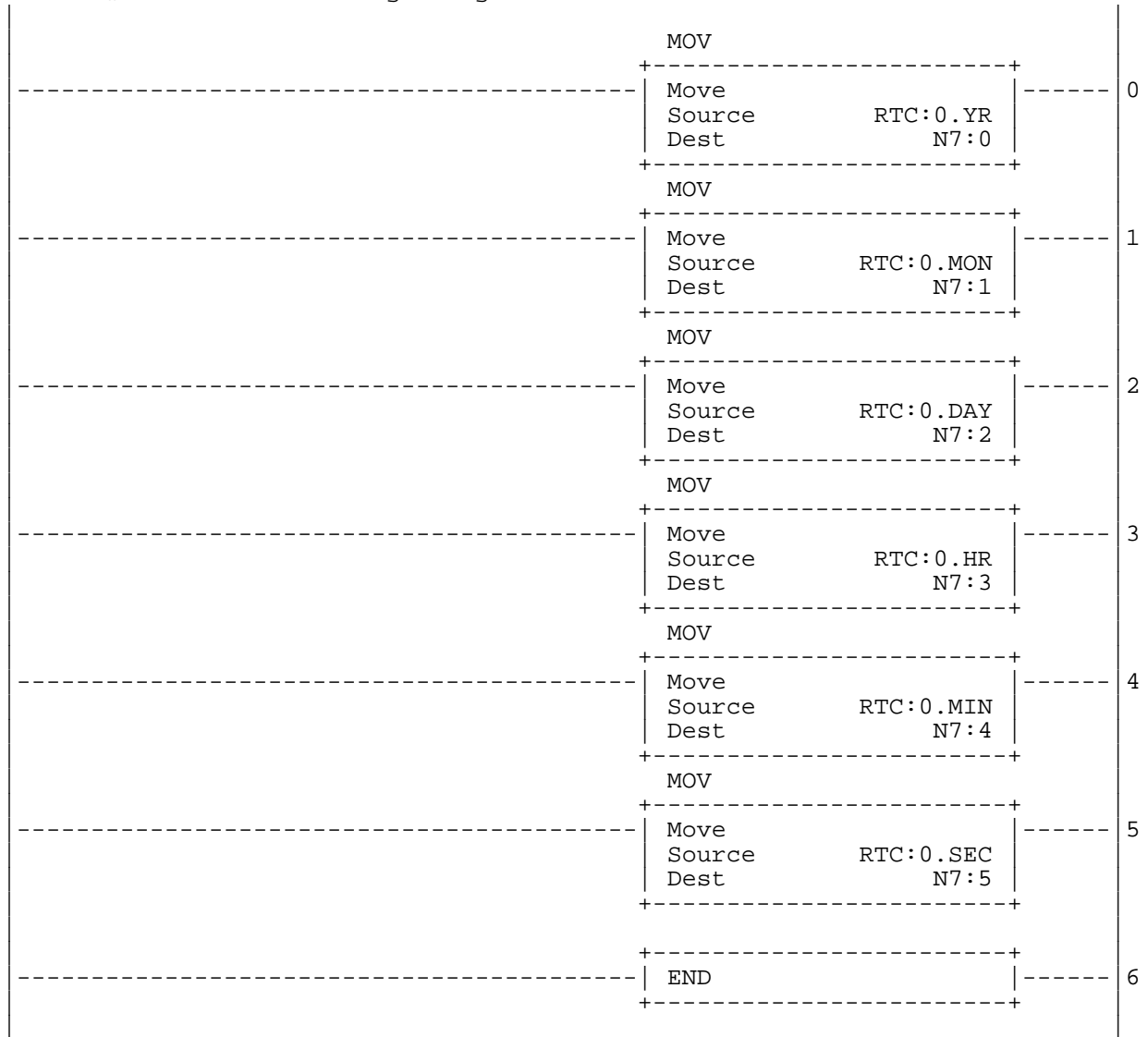
Ladder Diagram

Ladder #2: Main



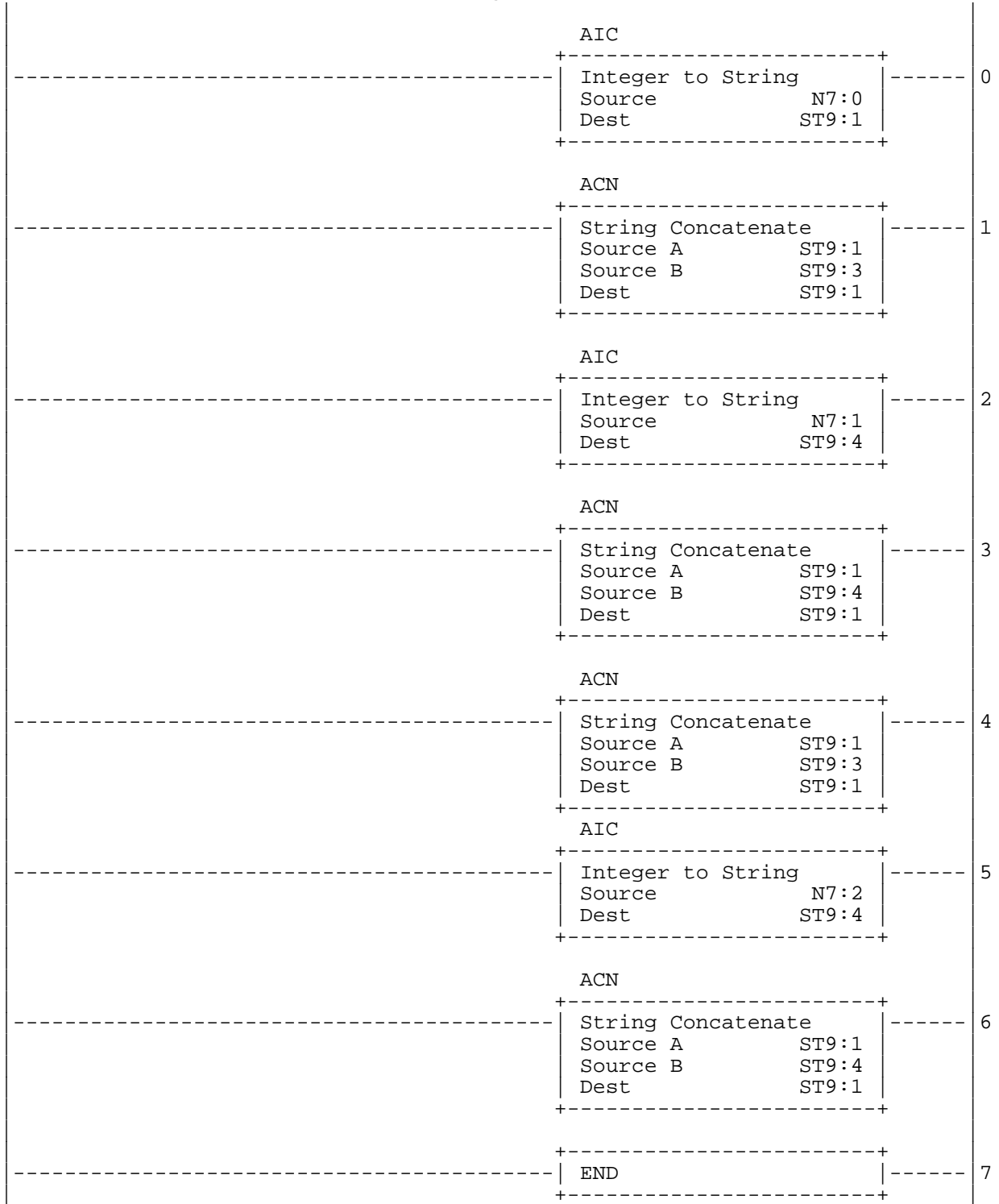
continued...

Ladder #3: Subroutine for getting RTC information



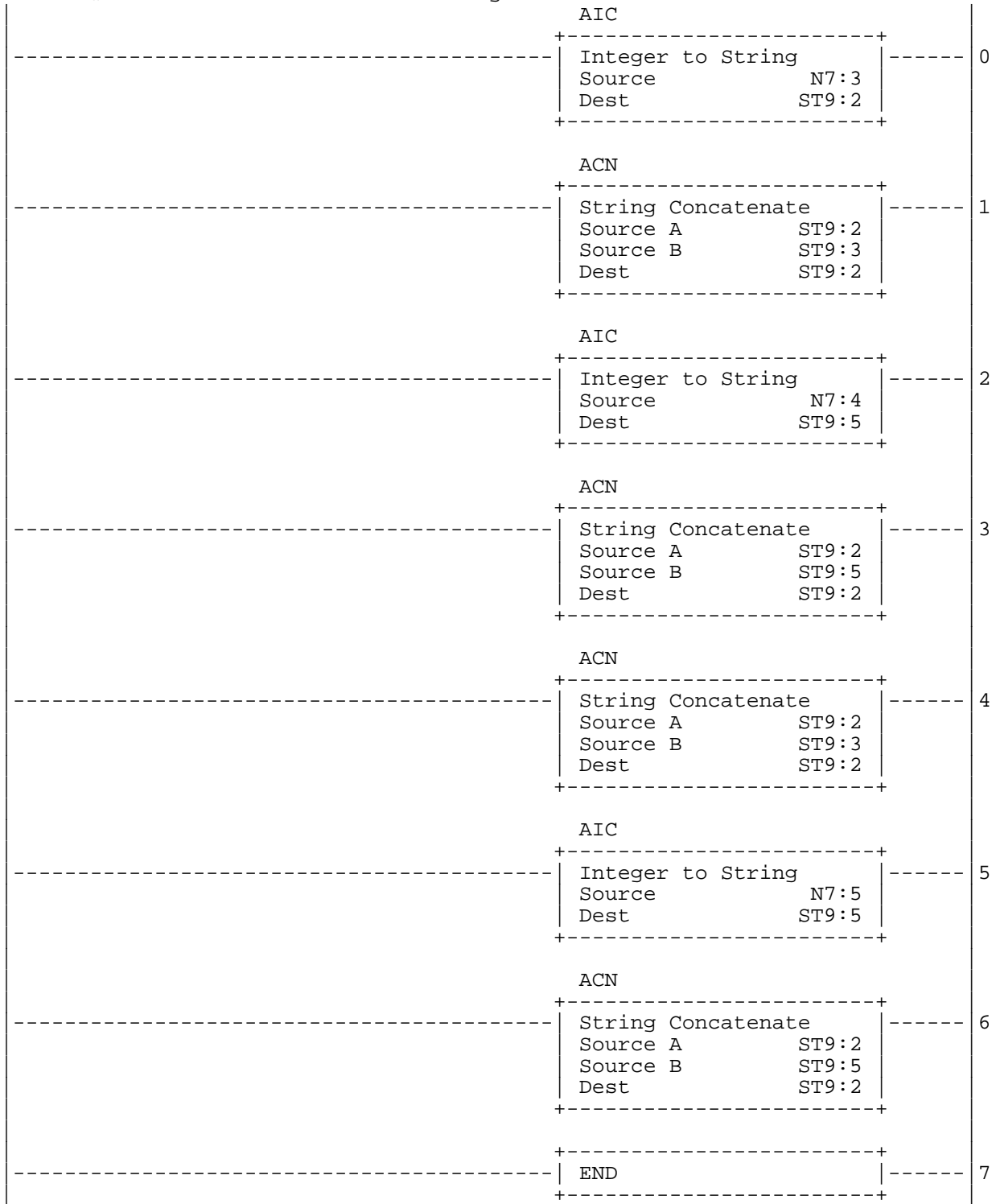
continued...

Ladder #4: Convert Date value to string



continued...

Ladder #5: Convert Time value to string



LCD Operation

Date and time value print on the LCD window:

Date & Time
2005-9-15
12-37-44

Date & Time
2005-9-15
12-37-46

Date & Time
2005-9-15
12-37-48

...

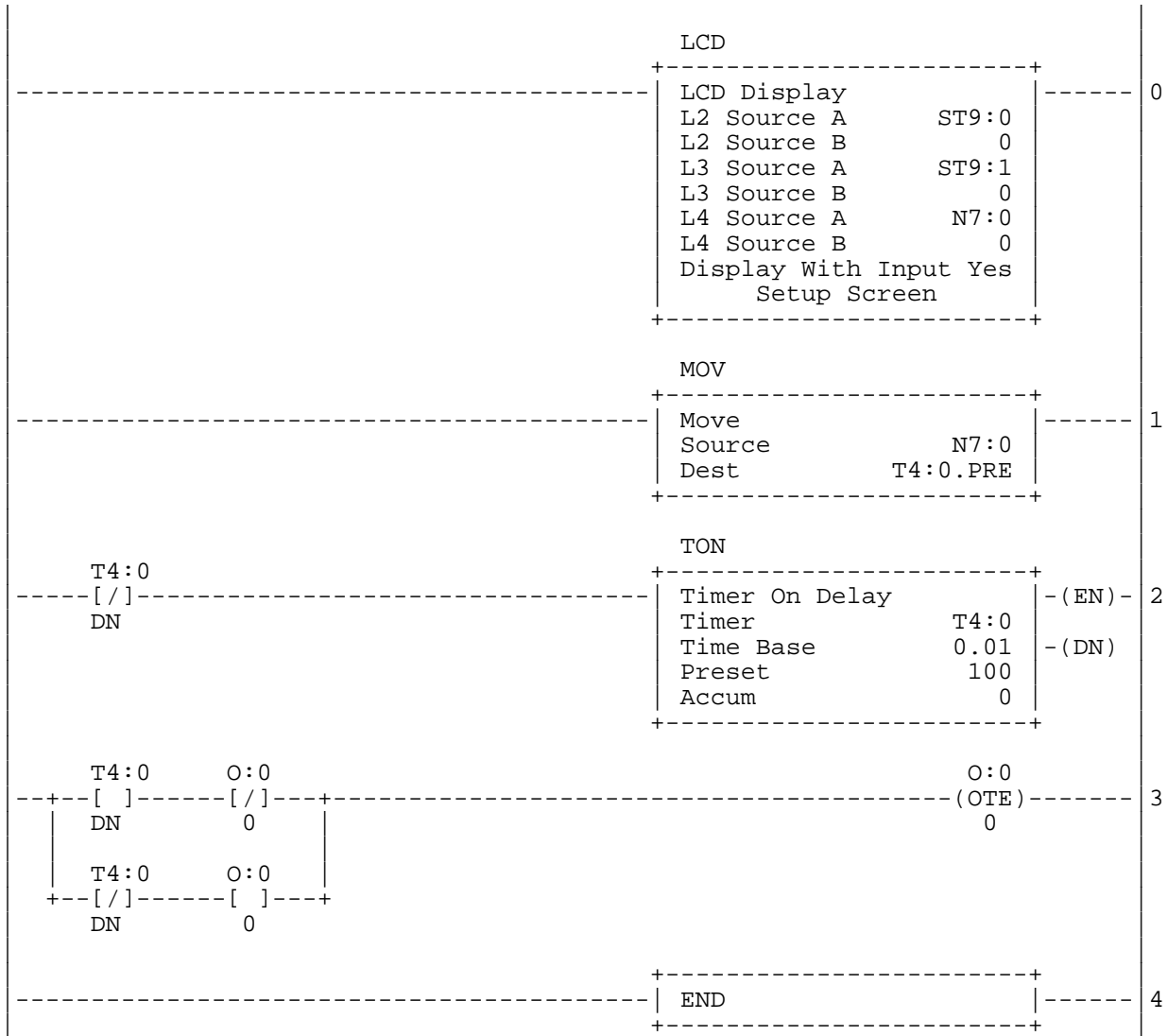
Example Code LCD 4

This is an example of string output to request integer data input from the operator using the LCD.

String Data File Values

Data File	Initial Value	Description
ST9:0	"Set Timer #4"	Line 2 string definition
ST9:1	"Preset?"	Line 3 query string definition

Ladder Diagram



LCD Operation

Line four of the LCD is blinking, waiting for operator input. You can change the value with LCD keys, Left/Right/Up/Down. The displayed value is updated when you press the OK key. If ESC key is pressed, the changed data is ignored, and the value in the integer file remains as the last data entered.

```

Set Timer #4
Preset?
= +00100
  
```

```

Set Timer #4
Preset?
= +00222
  
```

```

Set Timer #4
Preset?
= +00333
  
```

```

Set Timer #4
Preset?
= xxxxxx
  
```

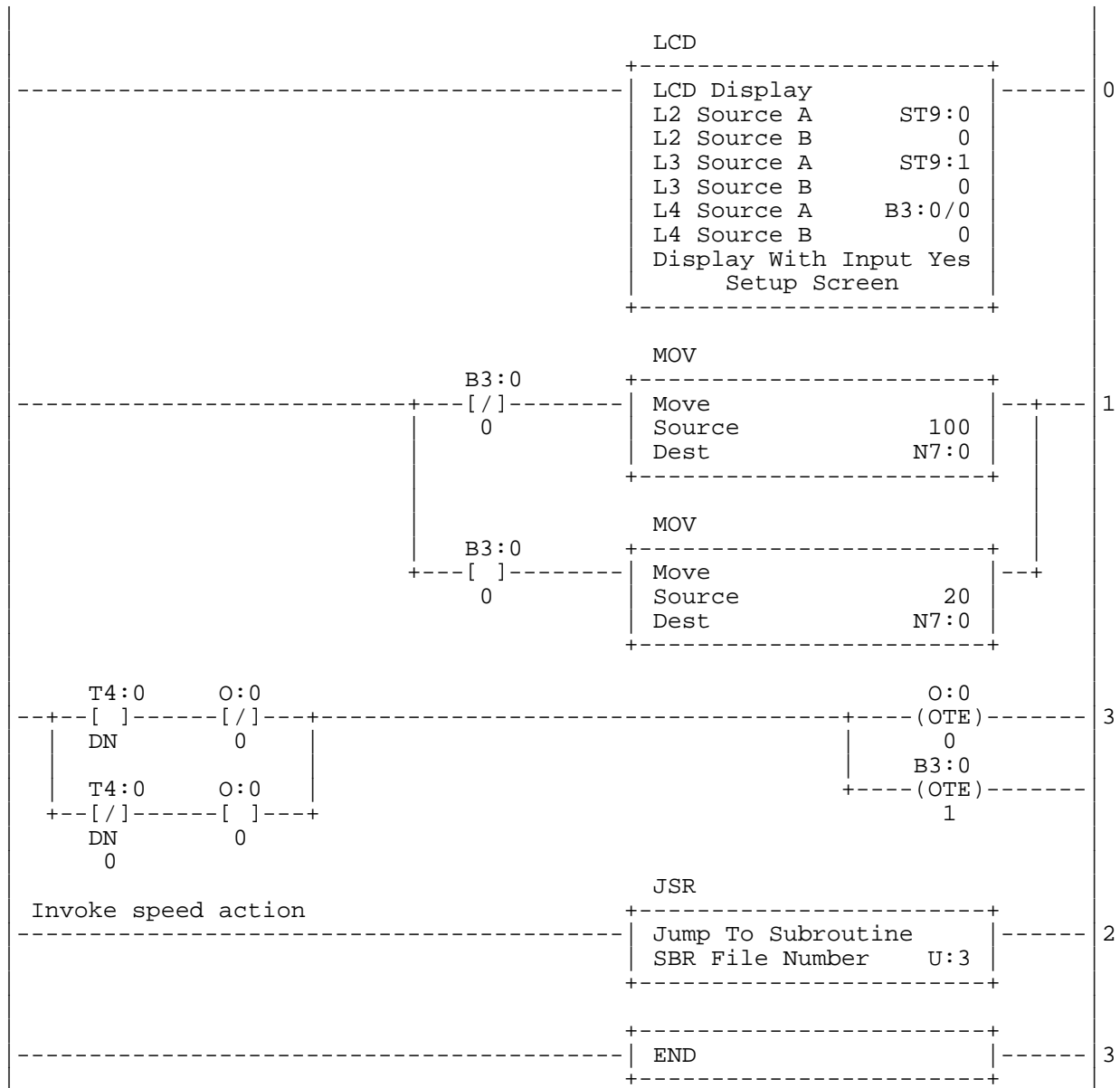
Example Code LCD 5

This is an example of string output to request bit data input from the operator using the LCD for "low / high" speed control

String Data File Values

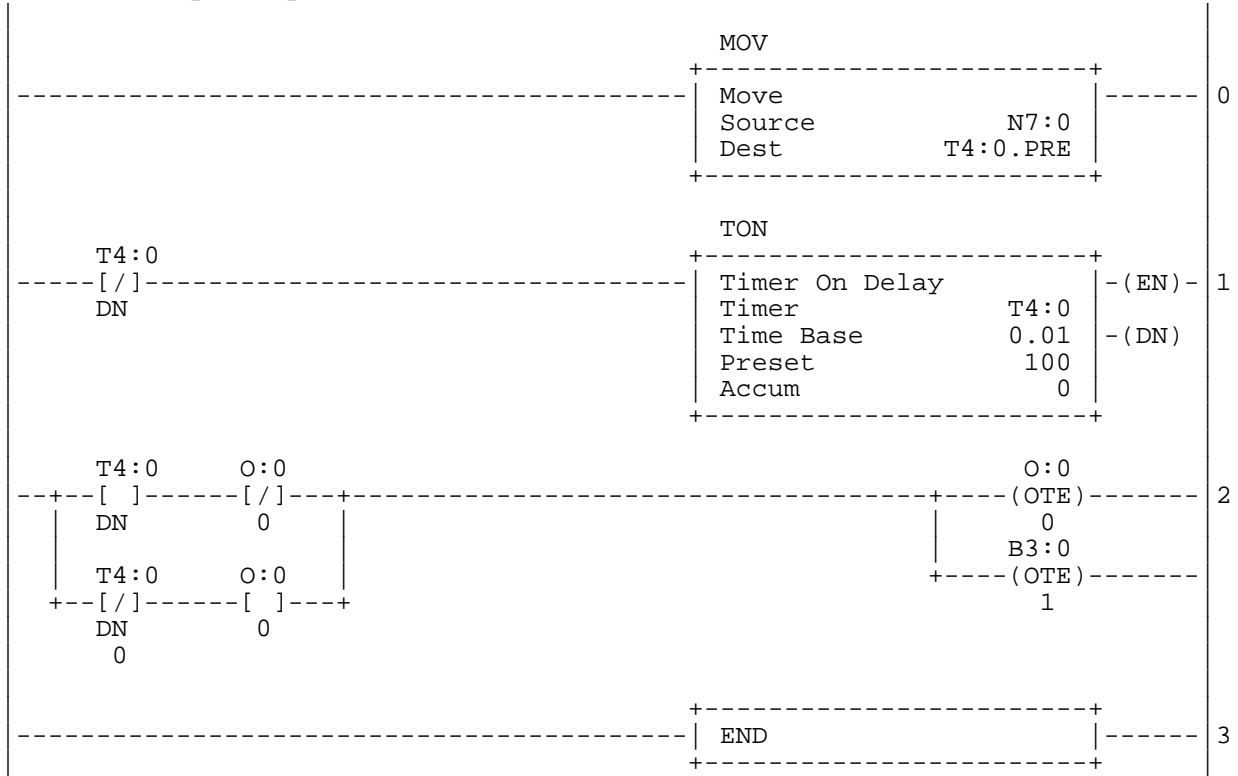
Data File	Initial Value	Description
ST9:0	"Speed?"	Line 2 string definition
ST9:1	"0.Low 1.High"	Line 3 string definition
N7:0	0	Speed value

Ladder Diagram



continued...

Ladder #3: Speed Operation Sub Routine



LCD Operation

Line four of the LCD is blinking, waiting for operator input. You can change the value with LCD keys, Left/Right/Up/Down. The displayed value in the bit data file, B3:0/0, is updated when you press the OK key. If ESC key is pressed, the changed data is ignored, and the value in the bit file remains as the last data entered.

```
Speed?
0.Low 1.High
=          +0
```

```
Speed?
0.Low 1.High
=          +1
```

```
Speed?
0.Low 1.High
=          +x
```

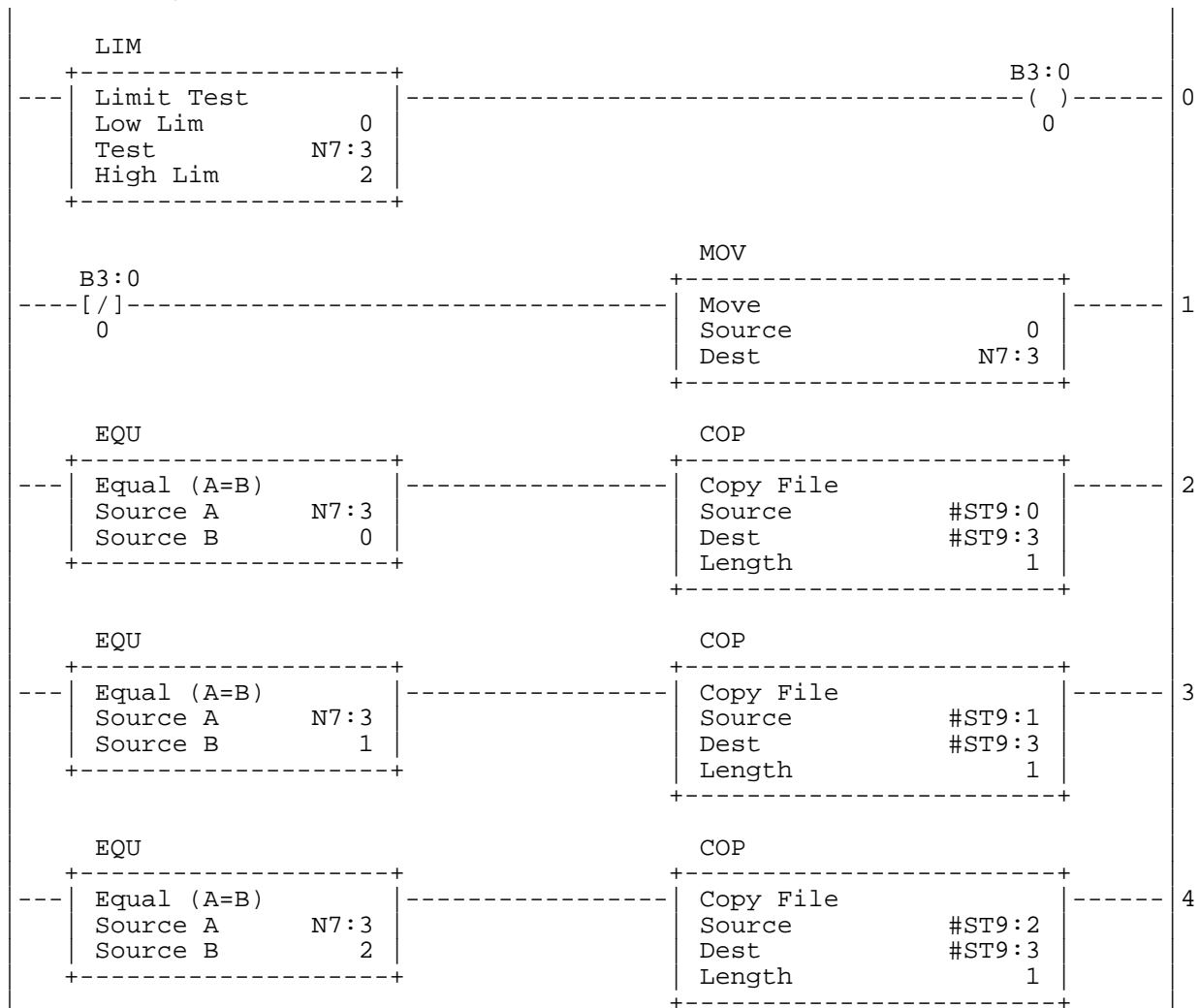
Example Code LCD 6

This is an example of using operator input, and indirect addressing, to request inventory information using multiple screens from the LCD.

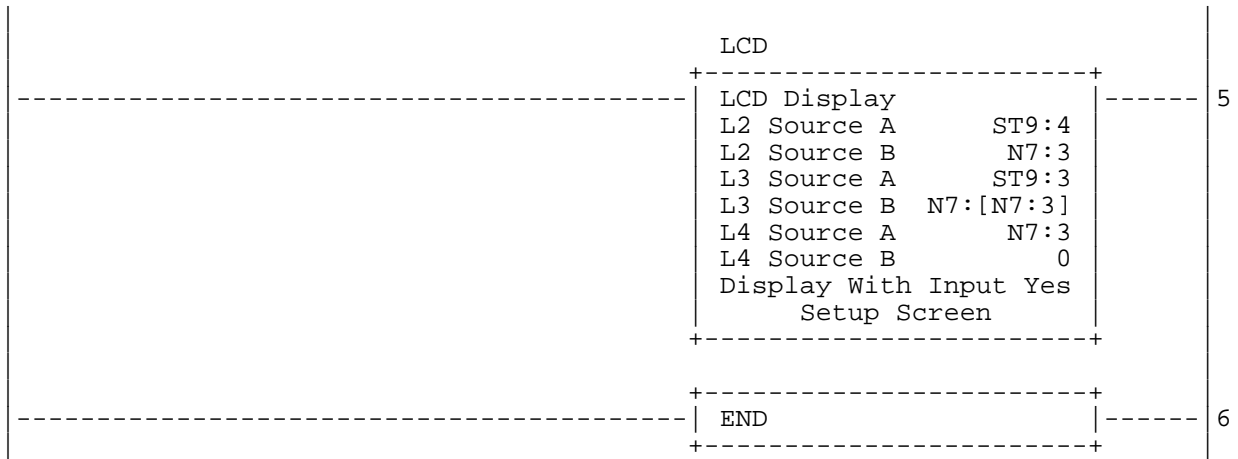
String and Integer Data File Values

Data File	Initial Value	Description
ST9:0	"Raw= "	Line 3 string pre-define
ST9:1	"Assy= "	Line 3 string pre-define
ST9:2	"Final= "	Line 3 string pre-define
ST9:3	""	Temporary address for string operation
ST9:4	"Stock Qty "	Line 2 string definition used as header in displayed message
N7:0	123	Quantity of Raw materials in inventory
N7:1	456	Quantity of Assembled components in inventory
N7:2	789	Quantity of Final finished product in inventory
N7:3	0	Inventory selection number

Ladder Diagram



continued...



LCD Operation

Line four of the LCD is blinking, waiting for operator input. You can change the inventory selection with LCD keys, Left/Right/Up/Down. The displayed value in the integer data file, N7:3, is updated when you press the OK key. If ESC key is pressed, the changed data is ignored, and the value in the bit file remains as the last data entered.

```
Stock Qty 0
Raw= 123
=           +0
```

```
Stock Qty 1
Assy= 456
=           +1
```

```
Stock Qty 2
Final= 789
=           +2
```

```
Stock Qty x
Final= x
=           +x
```

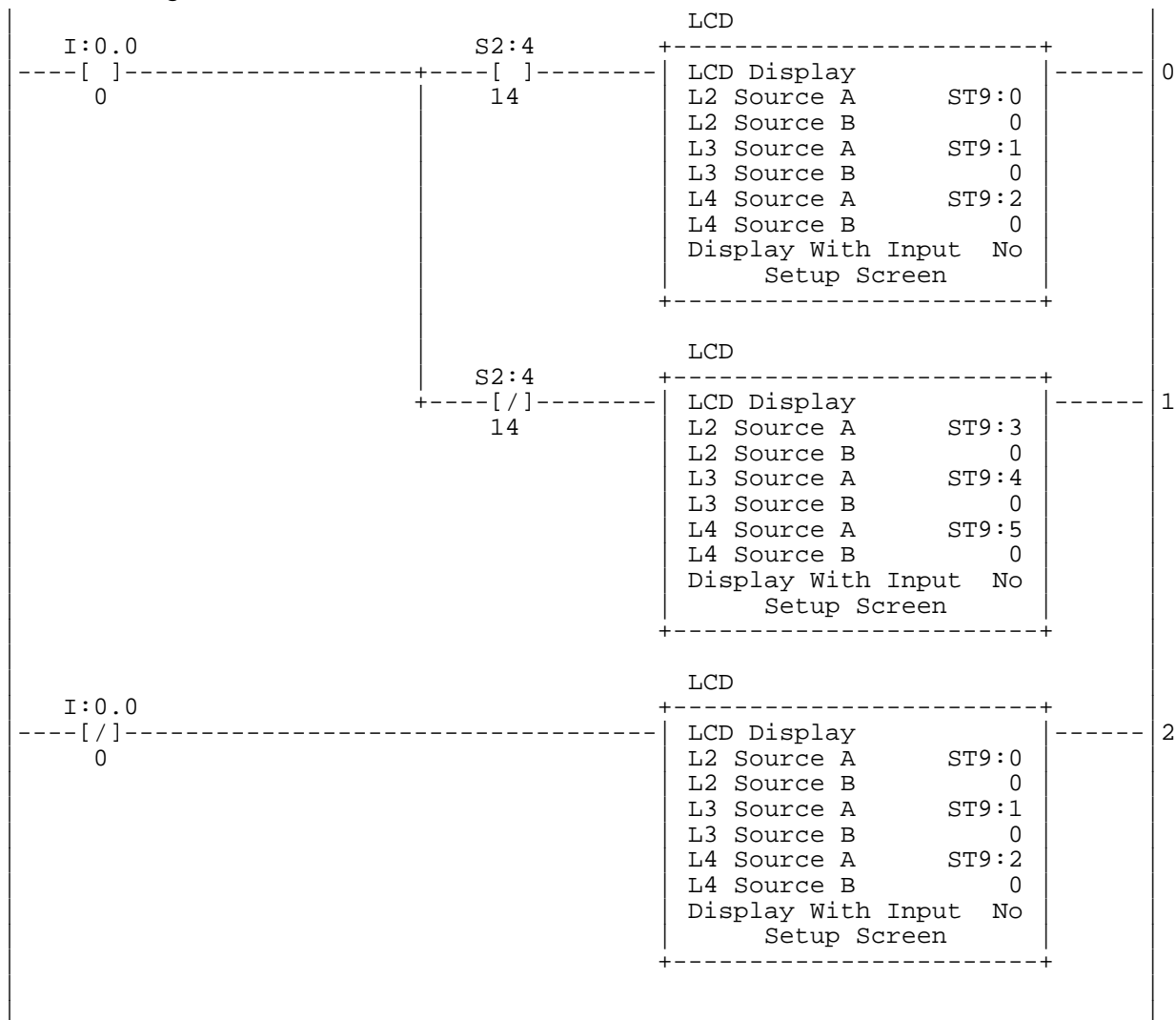
Example LCD Code 7 Alarm Windows

This is an example of using error bits to generate alarm windows for the operator.

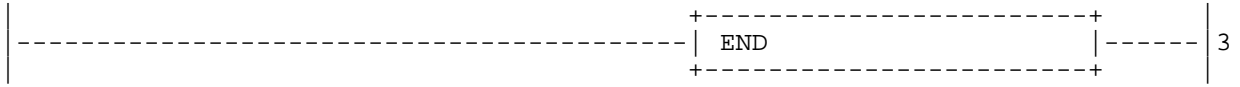
String and Integer Data File Values

Data File	Initial Value	Description
ST9:0	"Your system"	Line 2 string of fault window 1 and normal situation
ST9:1	"has an error"	Line 3 string of fault window 1
ST9:2	"Please clear"	Line 4 string of fault window 1
ST9:3	"the abnormal"	Line 2 string of fault window 2
ST9:4	" condition!"	Line 3 string of fault window 2
ST9:5	"ErrCode="	Line 4 left string of fault window 2
ST9:6	"is operating"	Line 3 string of normal situation
ST9:7	" normally."	Line 4 string of normal situation
I:0.0/0	0	System fault notification bit
N7:0	911	User defined error code.

Ladder Diagram

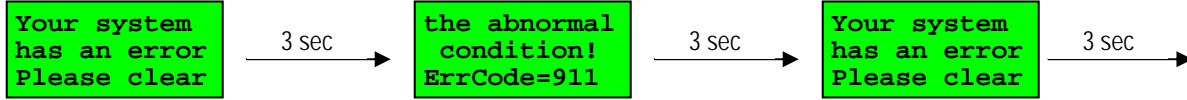


continued...

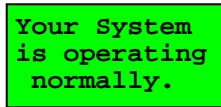


LCD Operation

If I:0.0/0 is ON:



If I:0.0/0 is OFF:



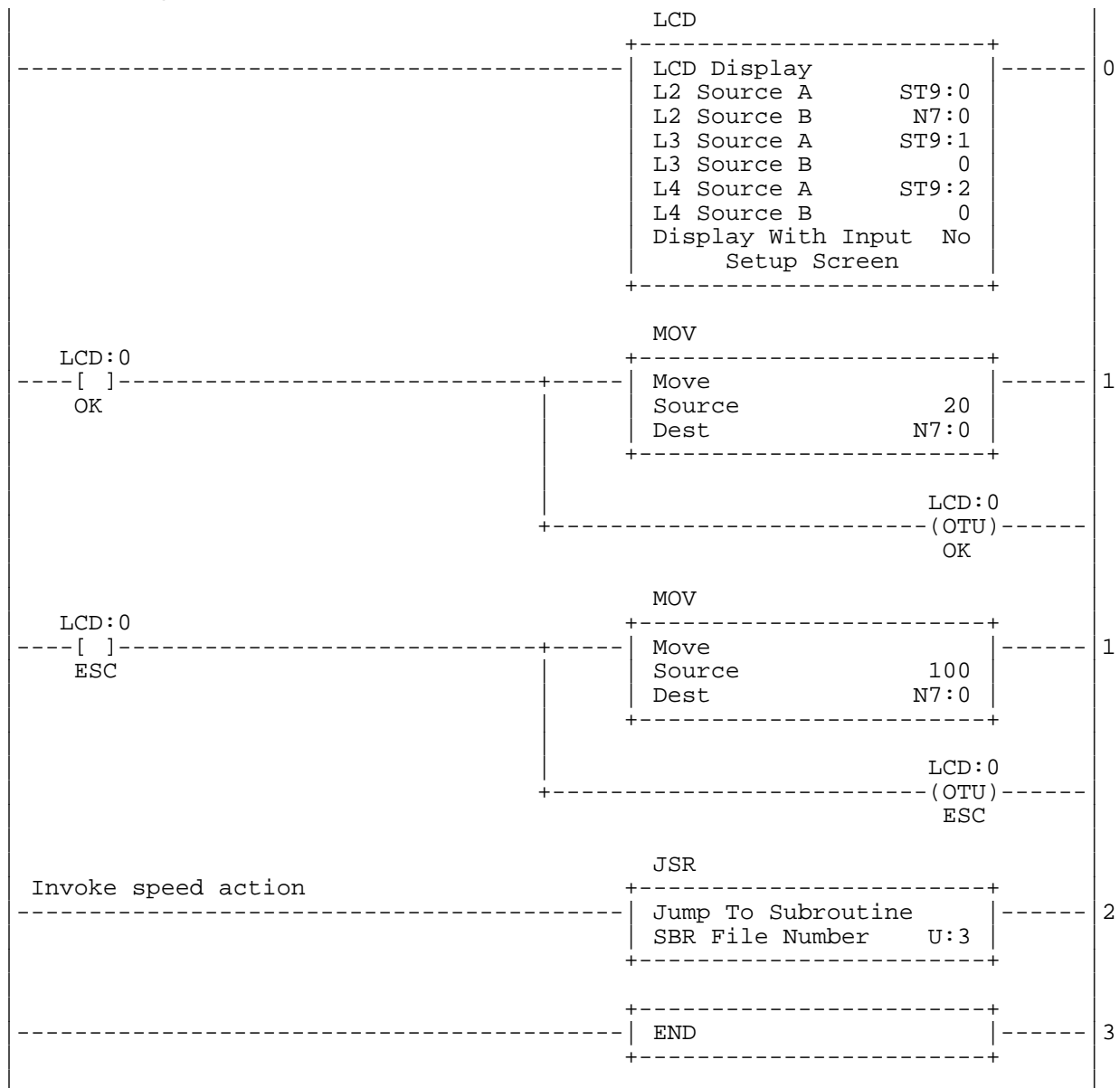
Example LCD Code 8 Alarm Windows

This is an example of simple "one touch" control using the keypad of the LCD.

String and Integer Data File Values

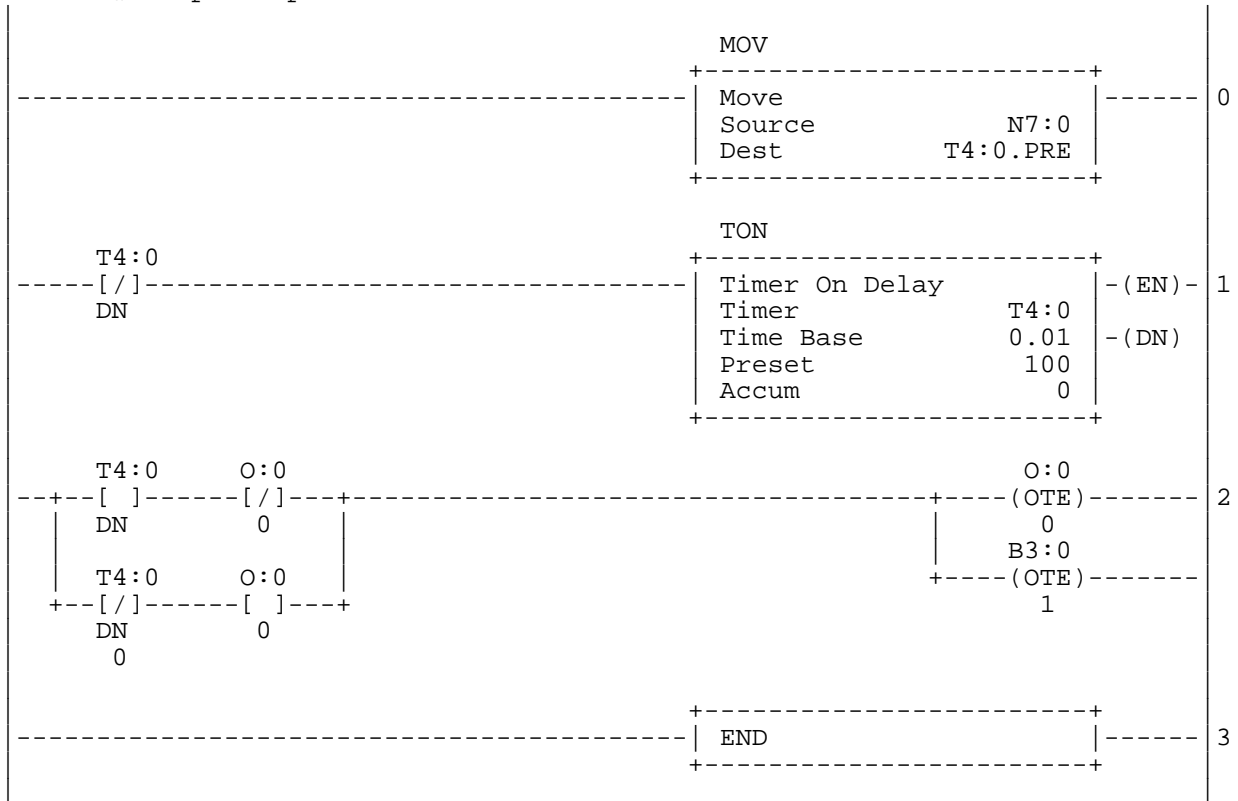
Data File	Initial Value	Description
ST9:0	"Speed="	Line 2 left string
ST9:1	"Select what?"	Line 3 string
ST9:2	"SLOW FAST"	Line 4 string
N7:0	100	used for set the timer preset value

Ladder Diagram



continued...

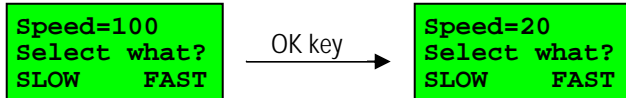
Ladder #3: Speed Operation Sub Routine



LCD Operation

If OK key pressed:

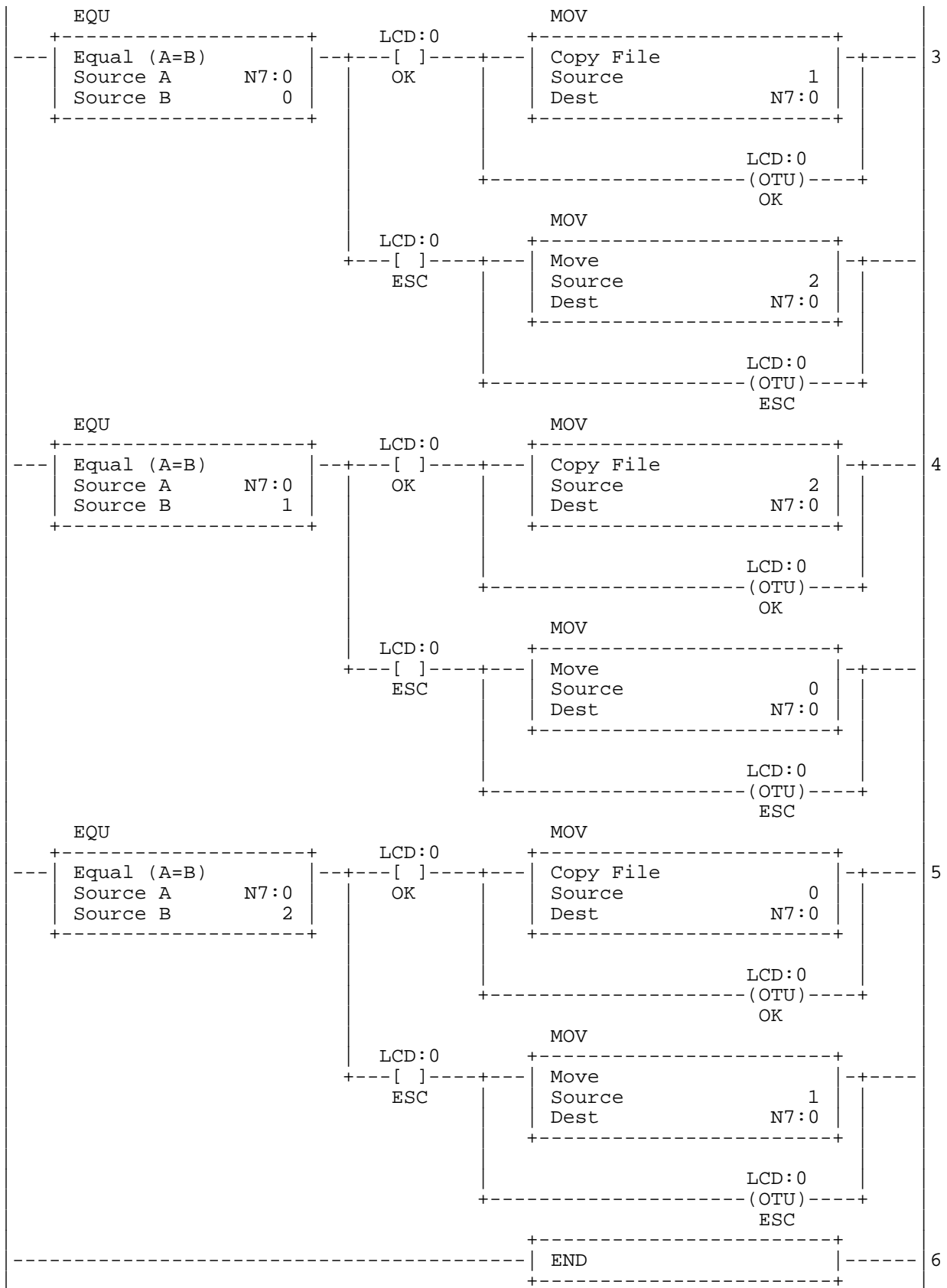
Timer preset value is set to 20, so the output is toggled faster.



If ESC key pressed:

Timer preset value is set to 100, so the output is toggled slower.



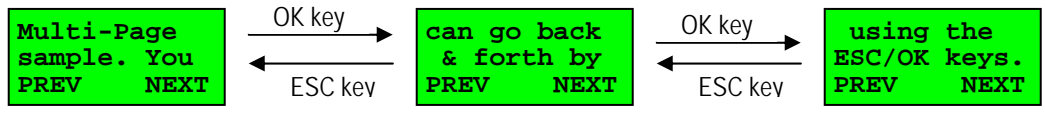


continued...

LCD Operation

If OK and ESC keys are pressed:

Three predefined message windows can be transacted by pressing the OK or ESC key of the LCD Module.



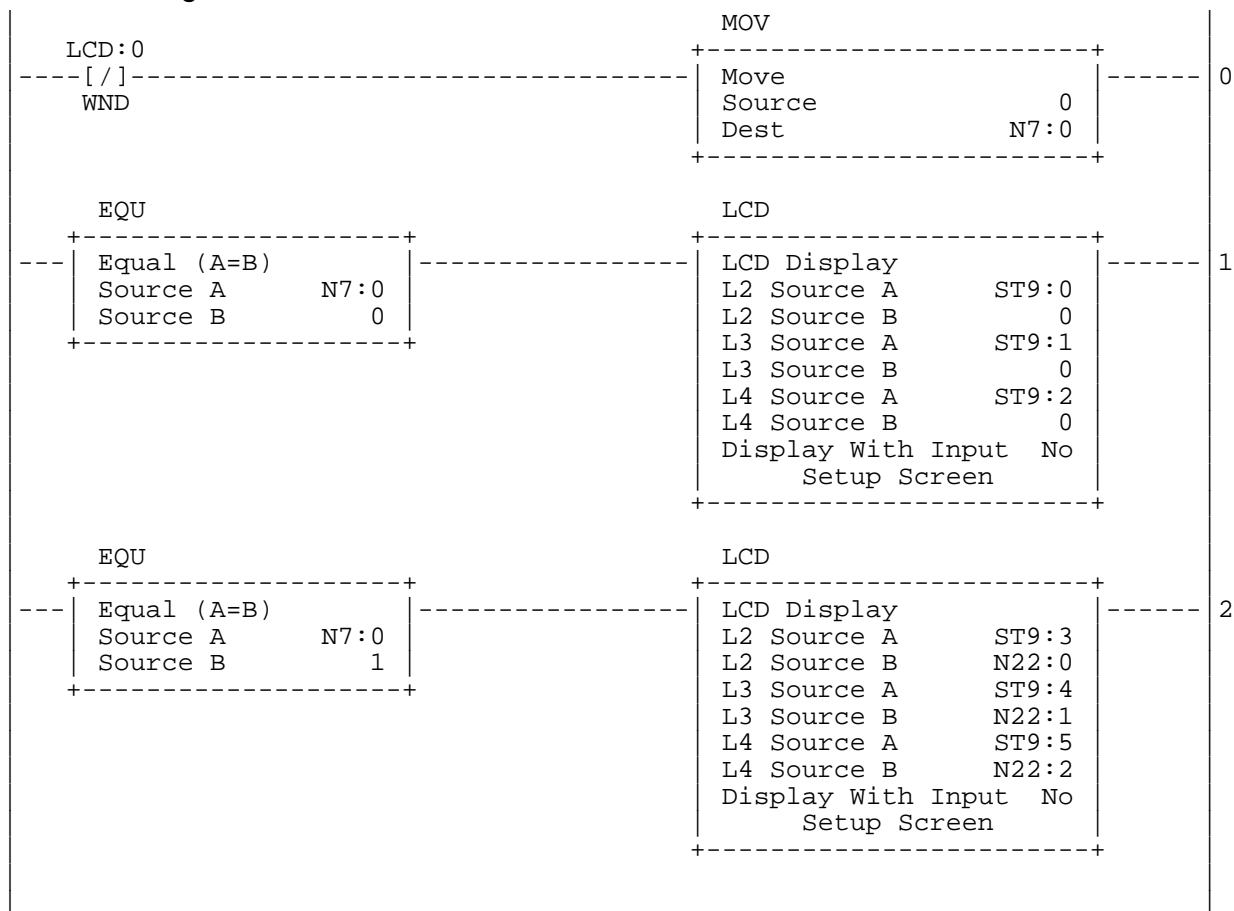
Example LCD Code 10 Monitor or Manipulate your data, decide using multiple display windows

This is an example of setting up a display message that uses the keypad of the LCD to decide if the user should just monitor controller parameters, or change into a modify mode if an adjustment is required.

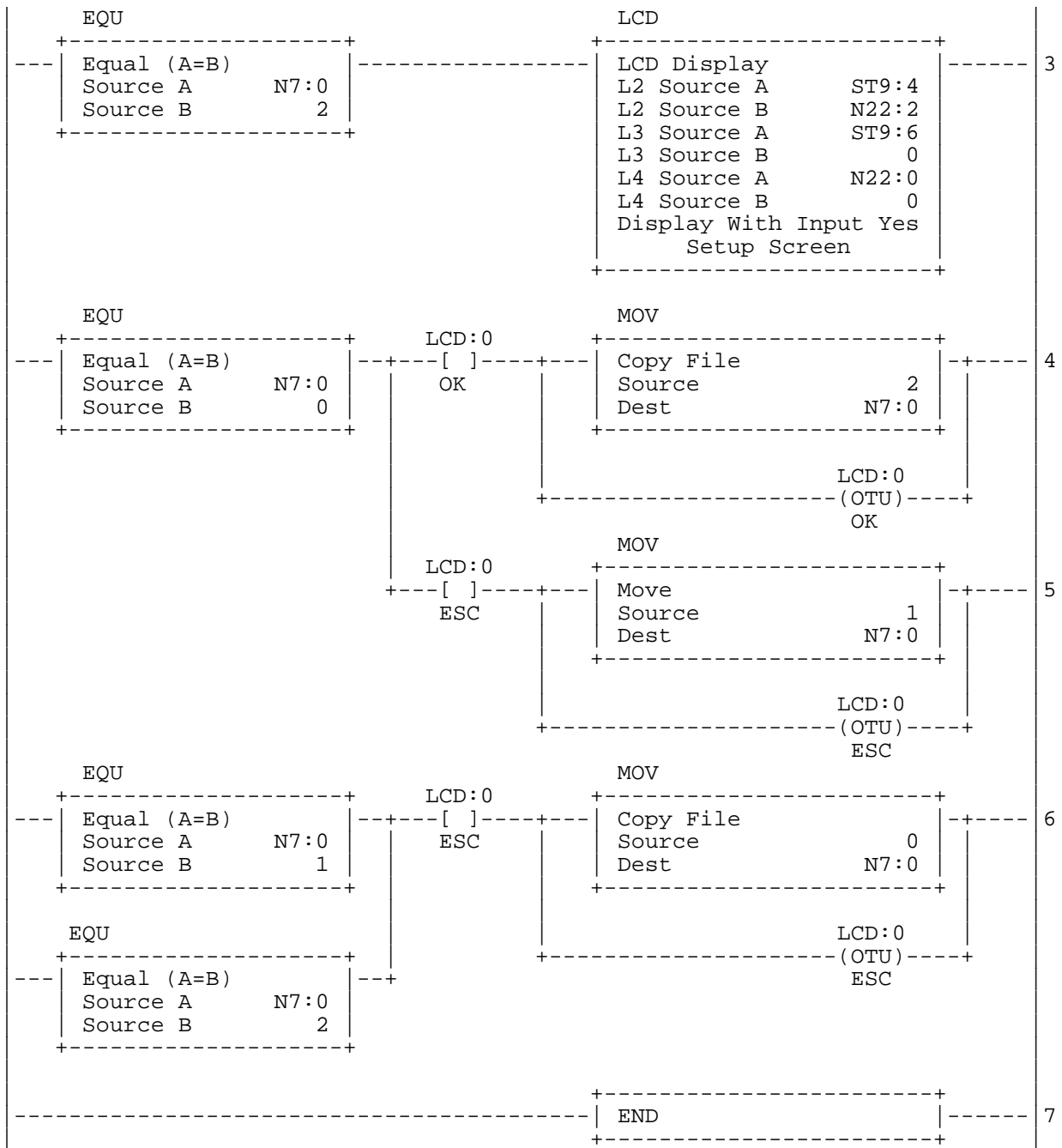
String and Integer Data File Values

Data File	Initial Value	Description
ST9:0	"VIEW STATUS?"	Line 2 string of window 1
ST9:1	"MODIFY DATA?"	Line 3 string of window 1
ST9:2	"VIEW MODIFY"	Line 3 string of window 1
ST9:3	"FREQ ="	Line 2 string of window 2
ST9:4	"TOTAL="	Line 3 string of window 2
ST9:5	"OUTPUT="	Line 4 string of window 2
ST9:6	"FREQ(10-30)?"	Line 3 string of window 3
N7:0	0	used for window Index
N22:0	10	Value. PTO Output Frequency (=PTO:0.FR)
N22:1	100	Value. PTO Total Output Pulses To Be Generated (PTO:0.TOP)
N22:2	0	Value. PTO Output Pulses Produced (PTO:0.OPP)
PTO:0.OUT	3	PTO Output
PTO:0.ADP	25	PTO Accel/Decel Pulses or File:Elem, if ADI=1

Ladder Diagram



continued...



continued...

LCD Operation

Select the functionally mode of the LCD (View Status of the operation, or Modify operating parameters):

If the ESC key pressed:

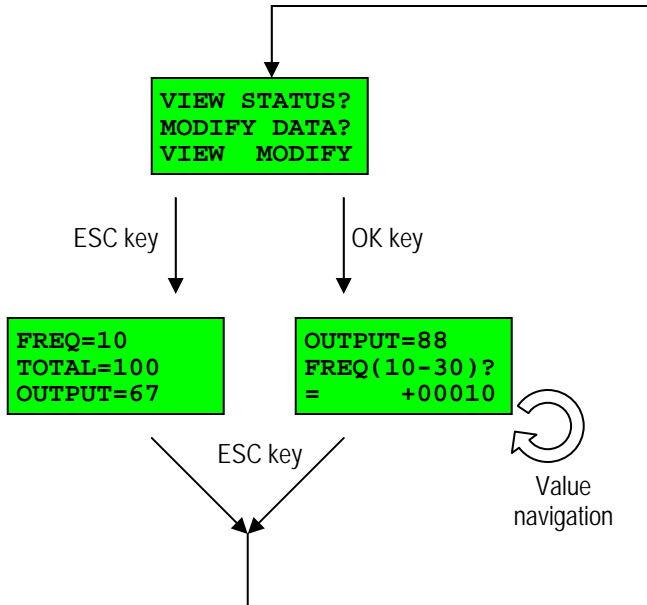
Current values for three operating parameters are displayed.

If ESC key pressed again, from the monitor page, the display returns to the selection window.

If the OK key pressed:

An input page is invoked. You can modify the current value with the Left/Right/Up/Down keys.

If ESC key pressed, from the modify page, the display returns to the selection window.



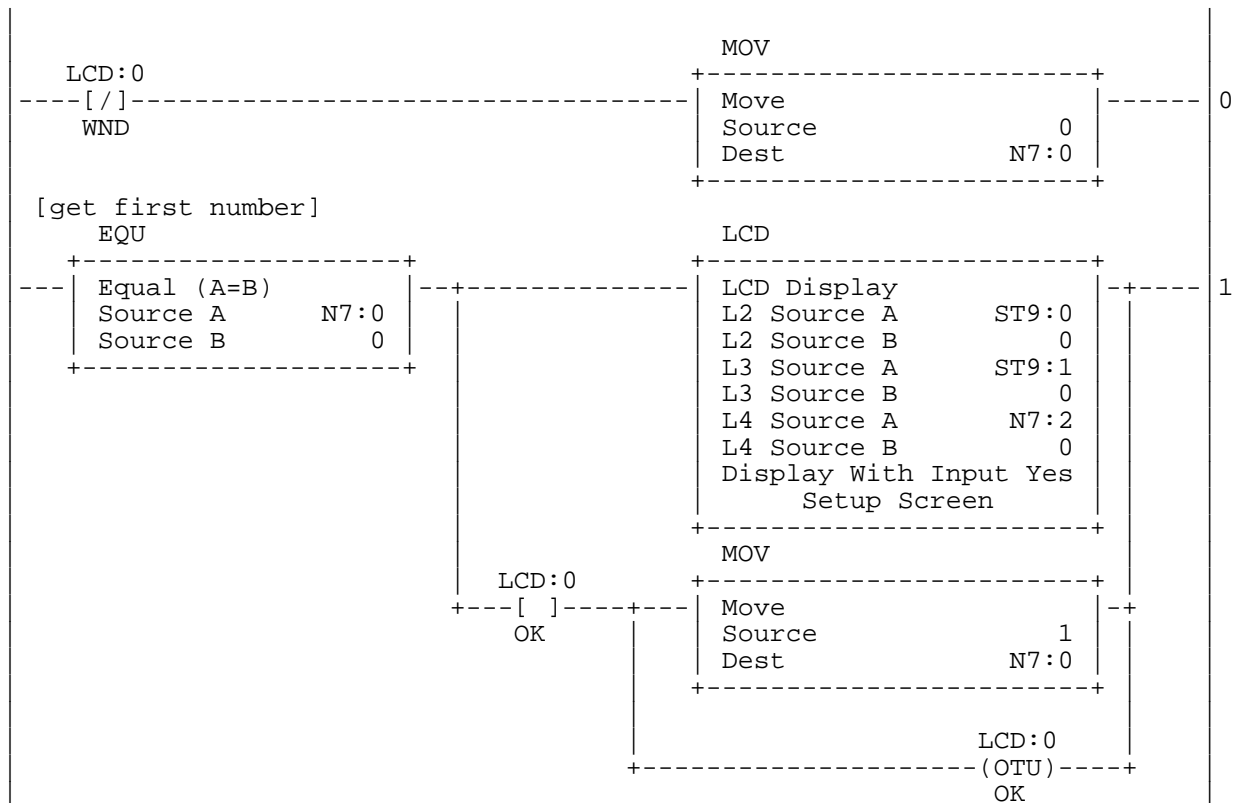
Example LCD Code 11 Forget your calculator? Build one into your controller!

This is a fun example what you can do with the versatility of the LCD and its keypad

String, Integer, and Status File Values

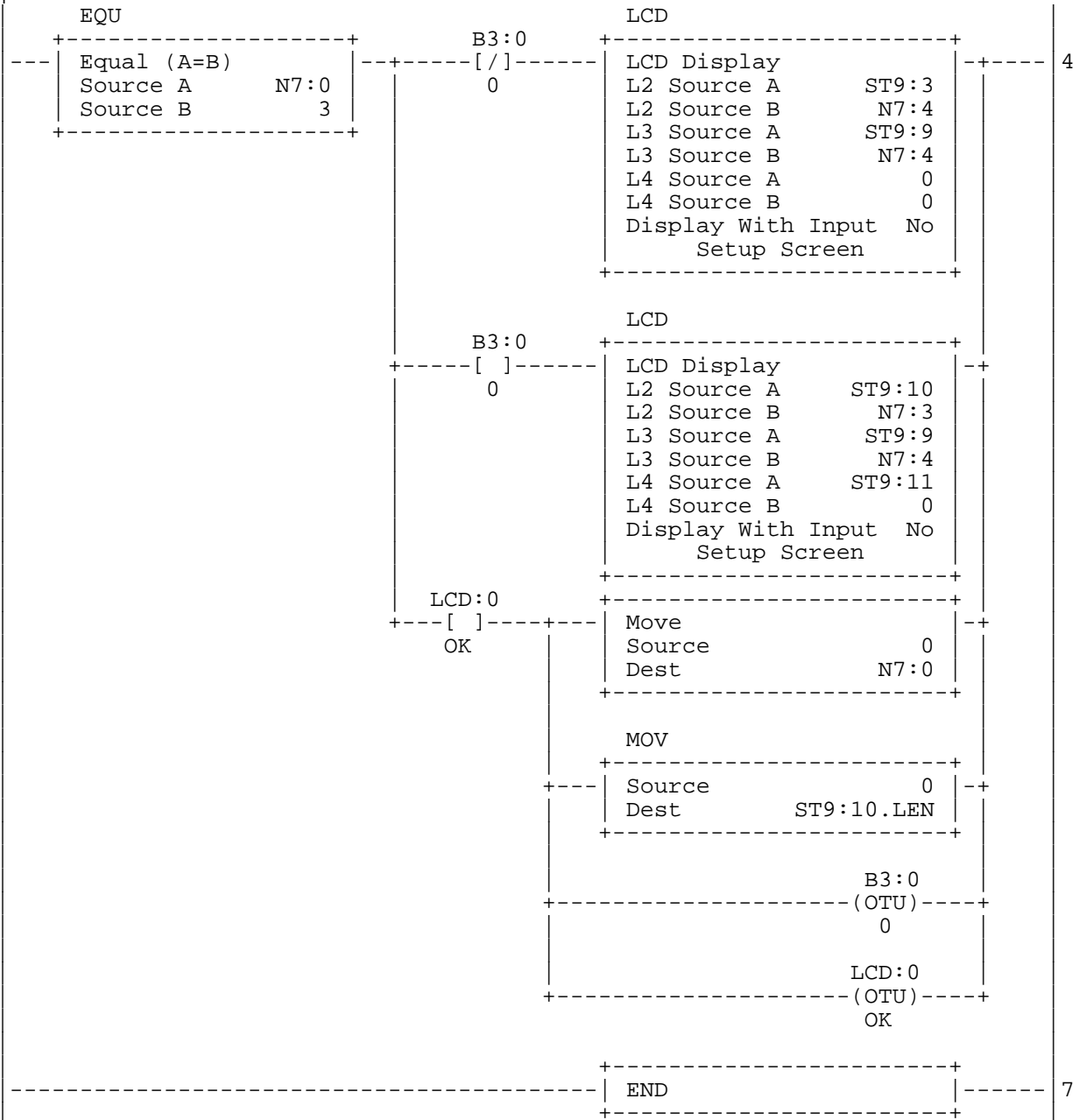
Data File	Initial Value	Description
ST9:0	"-CALCULATOR-	Line 2 string of window 0, 2
ST9:1	" Number 1?"	Line 3 string of window 0
ST9:2	" Number 2?"	Line 3 string of window 2
ST9:3	"0:+ 1:-"	Line 2 string of window 1
ST9:4	"2.* 3:/"	Line 3 string of window 1
ST9:5	"+"	Addition operand symbol
ST9:6	"_"	Subtraction operand symbol
ST9:7	"**"	Multiplication operand symbol
ST9:8	"/"	Division operand symbol
ST9:9	"="	Equal symbol
ST9:10	""	Temporary string buffer for string operation
ST9:11	"Range Over!"	Math Error Message
N7:0	0	used for window Index
N7:1	0	Calculation operand selection (0=add, 1=subtract, 2=multiply, 3=divide)
N7:2	0	Number 1
N7:3	0	Number 2
N7:4	0	Destination
S2:29	3	Fault Routine

Ladder Diagram



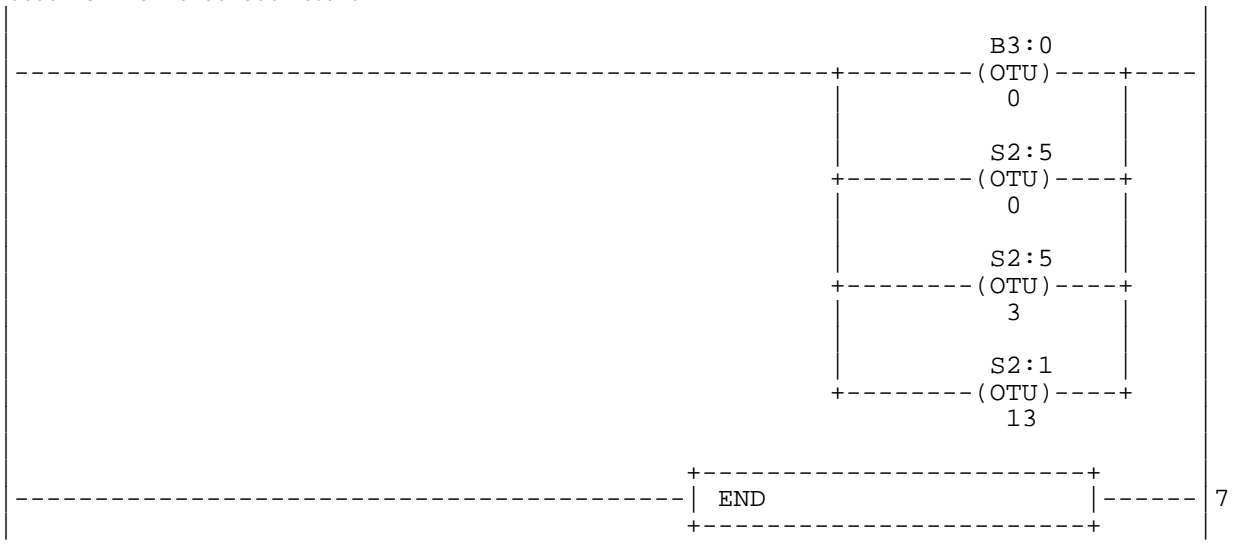
continued...

[show the result, or indicate math error, "Range Over!"]



continued...

Ladder #3: Error Hander Sub Routine



continued...

LCD Operation

Start by inputting the first number, Number 1, in the equation. Select the operand to determine which mathematical operation to perform. Select your second number, Number 2, and display your results.

Just press the OK key to navigate to the next window.

If any math error occurs while calculating the numbers, the "Range Over!" message will pop-up. As an example, refer to the following diagram. Since 2500×50 equals to 100,000 (which is greater than the maximum integer value), it is considered a math error in this program. The Error Handler Routine is programmed on the Ladder#3 file. If math error occurs, some of the error flags will be cleared.

