

Changing Programs

In This Chapter. . . .

- Two Ways to Edit a Program
 - Displaying a Program
 - Finding a Specific Instruction
 - Changing an Instruction
 - Inserting an Instruction
 - Deleting an Instruction
 - Using Search and Replace
 - Editing Programs During Run Mode
-

Two Ways to Edit a Program

Program Mode

You should select Program Mode for the majority of program changes. In this mode the CPU does not execute the application program so you do not have to worry about how the program changes can affect the machine operation. Obviously you can use Program Mode with either the DL430 or DL440 CPUs. The majority of this chapter shows you how to change various aspects of the program during Program Mode.

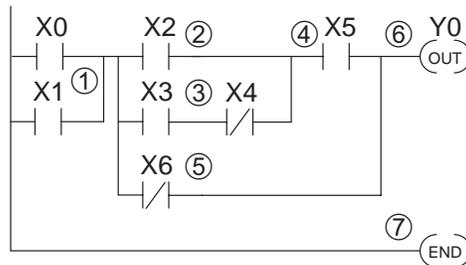
Run Mode Edits

With the DL440 CPU you also have the capability to edit the program during Run Mode. This is especially helpful if you only have to make minor adjustments such as adding or changing a single input contact or output coil. There is a separate section of this chapter that discusses using the Handheld for editing programs during Run Mode.

Displaying a Program

Since the Handheld displays the mnemonic instructions, you can step through the individual program instructions. If the CPU is in the RUN or TEST-RUN mode, the upper right corner displays the status for bit instructions (X, Y, C, S, T, CT, SP, or GX).

Ladder Representation



Mnemonic Listing and Addresses

| ADDRESS | INSTRUCTION | DESCRIPTION |
|---------|-------------|--|
| 0 | STR X0 | Starts branch 1 with X0 |
| 1 | OR X1 | Joins X1 in parallel with X0 |
| 2 | STR X2 | Starts branch 2 with X2 |
| 3 | STR X3 | Starts branch 3 with X3 |
| 4 | ANDN X4 | Joins X4 (NOT) with X3 |
| 5 | ORSTR | Joins branches 2 and 3 |
| 6 | AND X5 | Starts branch 4 with X5 |
| 7 | ORN X6 | Joins X6 (NOT) in parallel with X5 |
| 8 | ANDSTR | Joins branches 4 and 5 with 1-3 |
| 9 | OUT Y0 | Stores the output and finishes the network |
| 10 | END | Ends the program |

Use NXT or PREV to scroll through the addresses

\$(AD) NXT

```

$    0
STR  X 0000
    
```

Black indicates ON

NXT

```

$    2
OR   X 0001
    
```

S indicates OFF

PREV

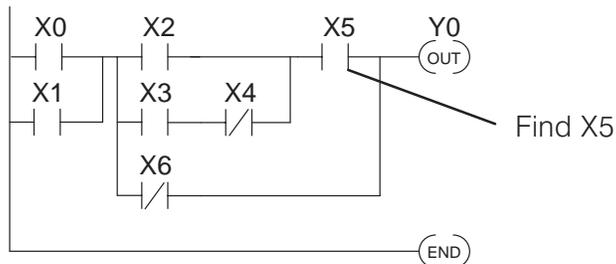
```

$    1
STR  X 0000
    
```

Finding a Specific Instruction

If you do not want to scroll through the program, you can use the FIND feature to automatically search for an instruction. The following example shows the instructions, addresses, and corresponding Handheld displays for a small program.

Ladder Representation



Identify the Instruction

| ADDRESS | INSTRUCTION | DESCRIPTION |
|---------|-------------|------------------------------|
| 0 | STR X0 | Starts branch 1 with X0 |
| 1 | OR X1 | Joins X1 in parallel with X0 |
| — | — | — |
| — | — | — |
| 6 | AND X5 | Starts branch 4 with X5 |
| — | — | — |
| — | — | — |
| 10 | END | Ends the program |

FIND the Address

AND X(IN) 5 FIND

SEARCHING
AND X5

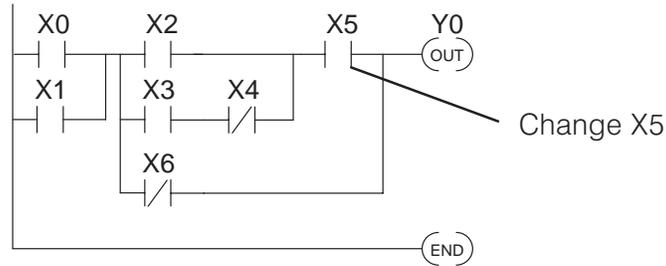
\$ 6
AND X 0005

You can continue searching for other instances of the instruction just by pressing the **FIND** key again. If the instruction is not found, the error message “E602 INSTRUCTION MISSING” is displayed. (If you think the message is incorrect, check your keystrokes to make sure you’re entering the instruction correctly and try the operation again.)

Changing an Instruction

Once you've found the instruction you can change it very easily. The following example shows you how to change the X5 contact to X10.

Ladder Representation



Identify the Instruction

| ADDRESS | INSTRUCTION | DESCRIPTION |
|---------|-------------|------------------------------|
| 0 | STR X0 | Starts branch 1 with X0 |
| 1 | OR X1 | Joins X1 in parallel with X0 |
| — | — | — |
| — | — | — |
| 6 | AND X5 | Starts branch 4 with X5 |
| — | — | — |
| — | — | — |
| 10 | END | Ends the program |

FIND the Address

AND X(IN) 5 FIND

SEARCHING
AND X5

\$ 6
AND X 0005

Change the Instruction

AND X(IN) 10

\$ 6
AND X10

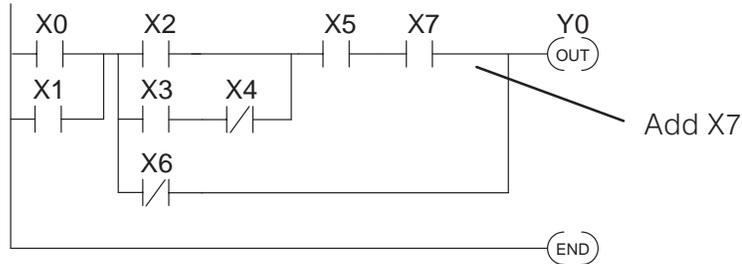
ENT
(New instruction is entered and display moves to next address.)

\$ 7
ORN X 0006

Inserting an Instruction

Use the INSERT feature to add an instruction to the program. INSERT adds an instruction *before* the instruction that is being displayed, so make sure you are at the correct program address. Once you've inserted the new instruction, the remaining addresses increment. The following example shows the instructions, addresses, and corresponding Handheld displays for a small program.

Ladder Representation



Identify the Instruction

| | ADDRESS | INSTRUCTION | DESCRIPTION |
|---------------|---------|-------------|------------------------------|
| | 0 | STR X0 | Starts branch 1 with X0 |
| | 1 | OR X1 | Joins X1 in parallel with X0 |
| | — | — | — |
| Insert before | 6 | AND X5 | Starts branch 4 with X5 |
| | | AND X7 | Adds X7 in series with X5 |
| | 7 | ORN X6 | Joins X6 (NOT) in parallel |
| | — | — | — |
| | 10 | END | Ends the program |

FIND the Address

OR NOT X(IN) 6 FIND

SEARCHING
ORN X6

\$ 7
ORN X 0006

Insert the New Instruction

AND X(IN) 7 SHFT INS

\$ 7 INSERT INST?
AND X7

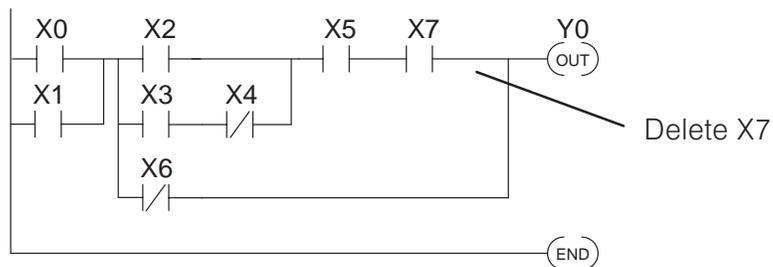
ENT

\$ 8
ORN X 0006

Deleting an Instruction

Use the DELETE feature to remove an instruction from the program. This operation deletes the instruction that is currently being displayed, so make sure you are at the correct program address. Once you've deleted the instruction, the remaining addresses decrement. The following example shows the instructions, addresses, and corresponding Handheld displays for a small program.

Ladder Representation



Identify the Instruction

| ADDRESS | INSTRUCTION | DESCRIPTION |
|---------|-------------|------------------------------|
| 0 | STR X0 | Starts branch 1 with X0 |
| 1 | OR X1 | Joins X1 in parallel with X0 |
| — | — | — |
| 6 | AND X5 | Starts branch 4 with X5 |
| 7 | AND X7 | Adds X7 in series with X5 |
| — | — | — |
| 11 | END | Ends the program |

FIND the Address

AND X(IN) 7 FIND

SEARCHING
AND X7

\$ 7
AND X 0007

Delete the Instruction

SHFT DEL

\$ 7 DELETE INST?
AND X 0007

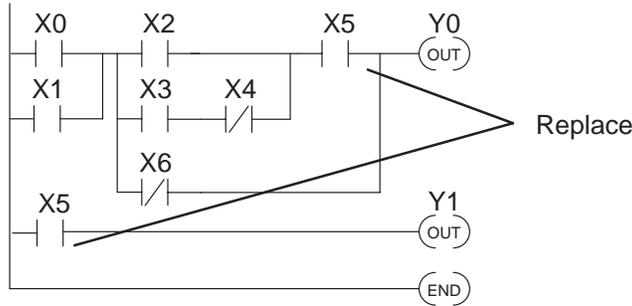
ENT

\$ 7
ORN X 0006

Using Search and Replace

Search and Replace allows you to quickly change all occurrences of a specific instruction. For example, you can replace every instance of X5 with X10.

Ladder Representation



Use AUX 22, Change Reference

AUX 2 2 ENT ENT

AUX 22 CHANGE REFERENCE
OLD █

X(IN) 5 ENT

AUX 22 CHANGE REFERENCE
TO █

X(IN) 1 0 ENT

AUX 22 CHANGE REFERENCE
X0005 --> X0010 ?

ENT

REFERENCE CHANGE OK
OLD █

SHFT EXIT (Clears display)

New Mnemonic Listing

| | ADDRESS | INSTRUCTION | DESCRIPTION |
|---------------|---------|-------------|---------------------------|
| | 0 | STR X0 | Starts branch 1 with X0 |
| New Reference | — | — | — |
| | 6 | AND X10 | Starts branch 4 with X5 |
| | — | — | — |
| | 10 | STR X10 | Begins a new rung with X5 |
| | — | — | — |

Editing Programs During Run Mode

With the DL440 CPU you can edit programs during Run Mode and Test-Run Mode. (See Chapter 6 for more information on Test Modes.) You use AUX 14, Run Time Edit, to modify the program.

Most of the things you can do in Program Mode also apply. For example, you can use the same techniques to search for a specific instruction, search for a specific address, etc. However, you cannot use Search and Replace during Run Mode.

The Run Mode Edits are not “bumpless.” Instead, the CPU maintains the outputs in their last state while it accepts the new program information. If an error is found in the new program, then the CPU will turn all the outputs off and enter the Program Mode.

WARNING: Only authorized personnel fully familiar with all aspects of the application should make changes to the program. Changes during Run Mode become effective immediately. Make sure you thoroughly consider the impact of any changes to minimize the risk of personal injury or damage to equipment. Edits during Run Mode are ideally suited to small changes.

If the program requires major changes it is strongly recommended you switch the system to program mode and take all necessary precautions just as if you were starting the machine for the first time.

Selecting AUX 14

We've already shown you how to select the various AUX functions, but a few things are different with AUX 14.

- Once you select AUX 14 the RUN LED starts blinking. This indicates the a Run Mode edit is in progress.
- If you had displayed an address just before selecting AUX 14, that address will automatically appear. So, you can search for an address or instruction *before* you select AUX 14 or *after* you select AUX 14.

Select AUX 14, Run Time Edit

AUX 1 4 ENT


```
AUX 1* OPERATING MODE
AUX 14 RUN TIME EDIT
```

Press ENT to select AUX 14 and display the address

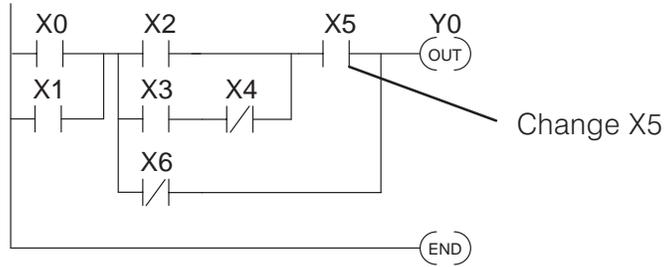
ENT

```
$xxxxxx
STR     X 0001
```

Changing an Instruction During Run Mode

Once you've found the instruction you can change it very easily. The following example shows you how to change the X5 contact to X10.

Ladder Representation



Identify the Instruction

| ADDRESS | INSTRUCTION | DESCRIPTION |
|---------|-------------|------------------------------|
| 0 | STR X0 | Starts branch 1 with X0 |
| 1 | OR X1 | Joins X1 in parallel with X0 |
| — | — | — |
| — | — | — |
| 6 | AND X5 | Starts branch 4 with X5 |
| — | — | — |
| — | — | — |
| 10 | END | Ends the program |

FIND the Address

AND X(IN) 5 FIND

SEARCHING
AND X5

\$ 6
AND X 0005

Change the Instruction

AND X(IN) 10 ENT

\$ 6
WANT TO ALTER?

Press CLR to abort the edit or ENT to accept

ENT

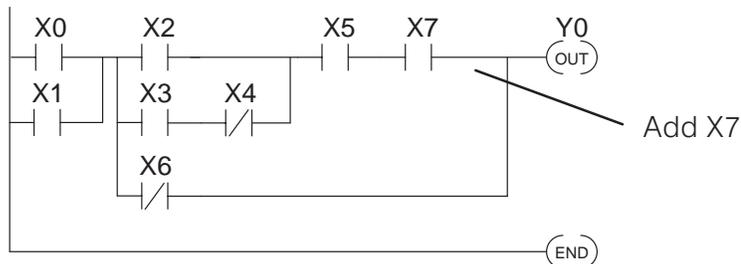
\$ 7
ORN X 0006

(If you press ENT, the change is accepted and the next address is displayed. If you pressed CLR, the current address is displayed.)

Inserting an Instruction During Run Mode

Inserting an instruction during Run Mode works almost exactly the same as it does during Program Mode. Remember, INSERT adds an instruction *before* the instruction that is being displayed and the remaining addresses increment.

Ladder Representation



Identify the Instruction

| | ADDRESS | INSTRUCTION | DESCRIPTION |
|---------------|---------|-------------|------------------------------|
| | 0 | STR X0 | Starts branch 1 with X0 |
| | 1 | OR X1 | Joins X1 in parallel with X0 |
| | — | — | — |
| Insert before | 6 | AND X5 | Starts branch 4 with X5 |
| | | AND X7 | Adds X7 in series with X5 |
| | 7 | ORN X6 | Joins X6 (NOT) in parallel |
| | — | — | — |
| | 10 | END | Ends the program |

FIND the Address

OR
 NOT
 X(IN)
 6
 FIND

SEARCHING

ORN X6

\$ 7

ORN X 0006

Insert the New Instruction

AND
 X(IN)
 7
 SHFT
 INS

\$ 7

WANT TO INSERT?

Press CLR to abort the edit or ENT to accept

ENT
 ←

\$ 8

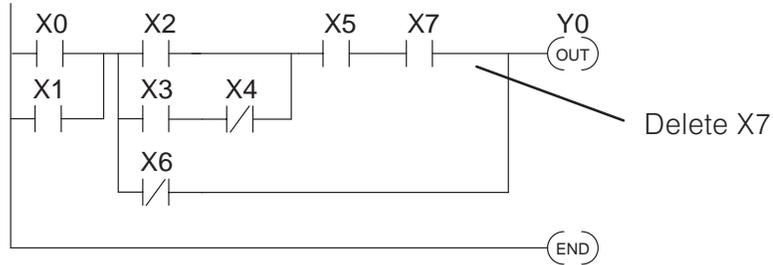
ORN X 0006

(If you press ENT, the change is accepted and the next address is displayed. If you pressed CLR, the current address is displayed.)

Deleting an Instruction During Run Mode

Deleting an instruction during Run Mode works almost exactly the same as it does during Program Mode. Remember, this operation deletes the instruction that is currently being displayed and the remaining addresses decrement.

Ladder Representation



Identify the Instruction

| ADDRESS | INSTRUCTION | DESCRIPTION |
|---------|-------------|------------------------------|
| 0 | STR X0 | Starts branch 1 with X0 |
| 1 | OR X1 | Joins X1 in parallel with X0 |
| — | — | — |
| 6 | AND X5 | Starts branch 4 with X5 |
| 7 | AND X7 | Adds X7 in series with X5 |
| — | — | — |
| 11 | END | Ends the program |

An arrow labeled "Delete" points to the instruction at address 7 (AND X7).

FIND the Address

AND X(IN) 7 FIND

SEARCHING
AND X7

\$ 7
AND X 0007

Delete the Instruction

SHFT DEL

\$ 7
WANT TO DELETE?

Press CLR to abort the edit or ENT to accept

ENT

\$ 7
ORN X 0006

(If you press ENT, the change is accepted and the next address is displayed. If you pressed CLR, the current address is displayed.)