6

System Monitoring and Troubleshooting

In This Chapter. . . .

Troubleshooting Suggestions

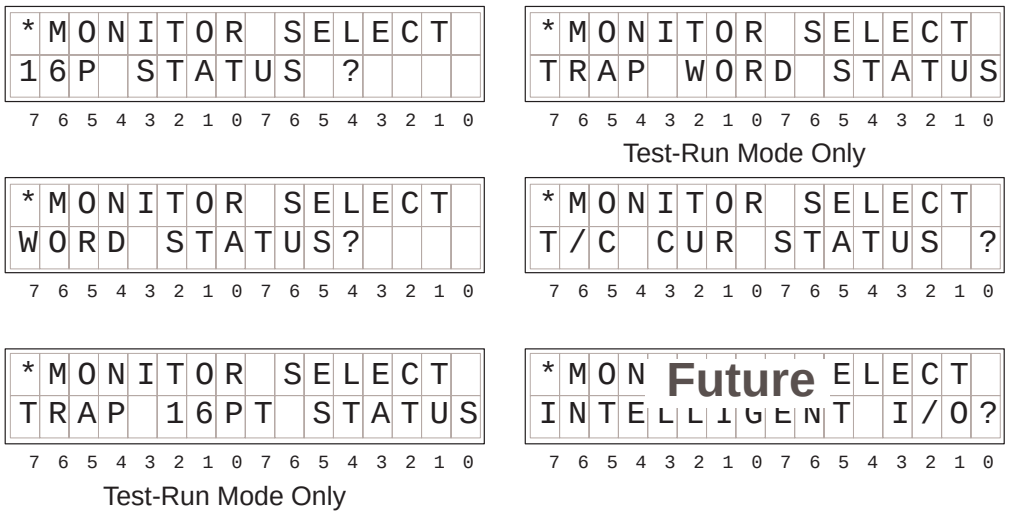
The Handheld programmer is useful for monitoring and troubleshooting your PLC and machine operation. There are several operations and features which help debug and isolate potential PLC problems. Below are some troubleshooting and maintenance features commonly used.

- **Monitor Discrete I/O Points** — to examine I/O power flow for individual I/O points.
- **Force Discrete I/O Points** — to examine machine sequences or inconsistencies.
- **Monitor V-Memory Locations** — to examine word locations to determine if correct values are being used.
- **Change V-Memory Values** — to force word locations with different values.
- **Monitor Timer/Counter Values** — to adjust machine timing elements.
- **Monitor CPU scan time** (in milliseconds) — view the maximum, minimum, and current scan times to adjust scan related problems.
- **Use Test Modes** — to examine output status.
- **Use I/O Diagnostics** — to pinpoint I/O errors.
- **Understand Error Codes** — to utilize many automatic error checks.

Understanding the Status Monitor Options

The Monitor Status display may be selected by pressing the **STAT** key. You may scroll status options using the **NEXT/PREV** keys. Some options may require the Handheld programmer to be on-line. The displays may change format depending on the CPU mode selected when the Status display operation is performed.

Example displays for Monitor Status options



HINT: The Handheld programmer will buffer up to 5 Status displays which may be scrolled by just pressing the **STAT** key. You can also scroll the display to adjacent memory locations by pressing the **PREV** and **NEXT** keys.

Monitoring Discrete I/O Points

The Handheld programming unit will allow Status Monitoring on the following data types. You may monitor 16 data points at one time.

X inputs	T–Timer/Counter bits
Y output	S, SP–Special relays
C control relays	GX remote I/O points (future)
Stage bits	

Bit Status Monitor

The Status Monitor may be displayed using the **STAT** menu or by directly typing in the memory reference. The following figure shows both methods of selecting Status monitor.

Press these Keystrokes

1. To select Bit Status Monitor mode

STAT ENT

2. To select Status type and enter reference

NEXT NEXT A 0 ENT

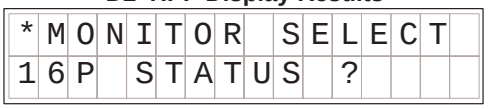
To call Bit Status directly

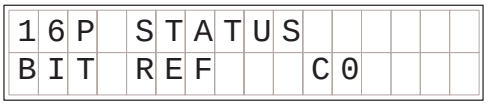
1. To directly call specific Status with reference

SHFT C 2 A 0 STAT

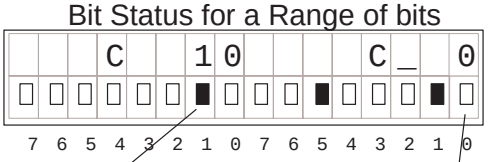
- Press CLR to exit Status function

D2–HPP Display Results



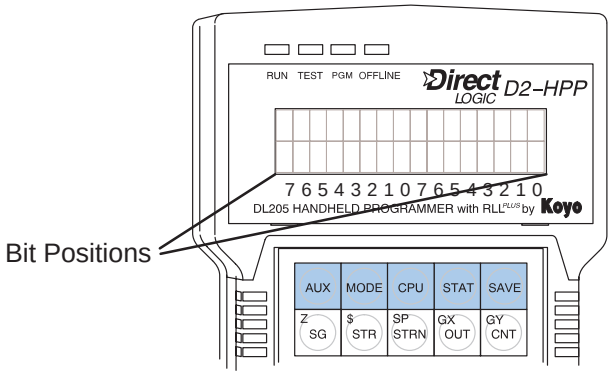


Bit Status for a Range of bits



Solid fill indicates ON ■ Blank indicates OFF □

If you examine the Handheld programmer, you will notice several numbers printed on the case, below the LCD display screen. These numbers help you identify which data points you are monitoring.



Forcing Discrete I/O Points

The DL105 and DL205 supports two methods which may be used to force I/O points. Both systems will allow you to use the Status Monitor mode and change individual bit conditions by pressing the **ON** and **OFF** keys. With certain DL205 CPU's you can also force I/O using **AUX 59** the Bit Override function.

The following paragraphs describe the two forcing methods available. (Please refer to the DL105 and DL205 User Manuals for detailed description of how the CPU processes each type of forcing request.)

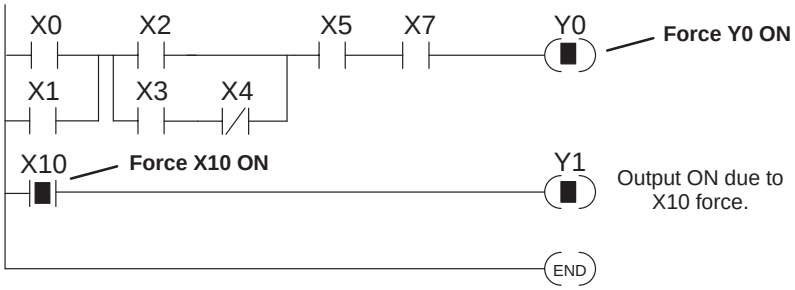
- **Regular Forcing —** This type of forcing can temporarily change the status of a discrete bit. For example, you may want to force an input on, even though it is really off. This allows you to change point status which is in the image register for one scan. **This value will be valid until the image register location is written to during the next scan.** This is primarily useful during testing situations when you just need to force a bit on to trigger another event. An example of regular forcing is on the next page.
- **Bit Override — (DL240 Only)** Bit override can be enabled on a point-by-point basis by using AUX 59. You can use Bit Override with X, Y, C, T, CT, and S data types. Bit override basically disables any changes to the discrete point by the CPU. For example, if you enable Bit Override for input X1, and X1 is OFF at the time, then the CPU *will not* change the state of X1. This means that even if input X1 turns ON, the CPU will not acknowledge the change. So, if you used X1 in the program, it would always be evaluated as OFF in this case. Of course, if X1 was on when the bit override was enabled, then X1 point would always be evaluated as ON.

WARNING: Depending on your application, forcing I/O points may cause unpredictable machine operation that can result in a risk of personal injury or equipment damage. The Force function is usually performed during troubleshooting only. Be sure all I/O is unforced when operation testing is done.

Using Force during Bit Override

As mention, the Bit Override feature is not supported by all PLC models. When using the Bit Override this does *not* disable using Regular forcing. For example, if you enabled the Bit Override for Y0 and it was off at the time, then the CPU would not change the state of Y0. However, you *can* still use a programming device to change the bit status. Now, if you use the programming device to force Y0 ON, it will remain forced and the CPU program will not change the state of the Y0 output. If you then force Y0 OFF, the CPU will maintain Y0 in the OFF condition. The CPU will never update the point with the results from the application program or from the I/O update until the bit override is removed from the point.

Forced I/O Example

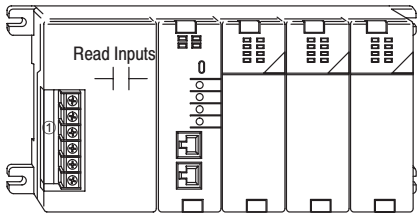


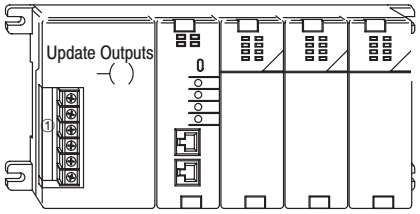
CPU Process Update

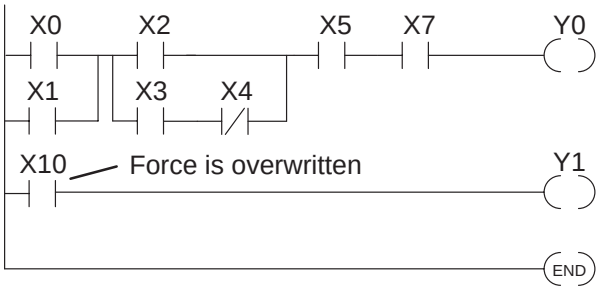
1. The CPU first reads the I/O status from the modules. If discrete input point X10 is off, the CPU overwrites the force command and turns off X10.

2. While X10 is off, even though previously forced on, Y0 will remain turned off. The CPU will scan and process all program instructions.

3. At the end of the program scan, the CPU updates the output status with the results obtained from the logic execution. Y0 and Y1 were turned off.







Regular Bit Force using the Status Monitor

The Force Bit operation will allow controlling a specific bit ON and OFF within memory tables. The force function does *not* overrule the regular execution of your program logic. Even when a bit has been forced, your program will control the bit through executed program instruction.

Press these Keystrokes

1. Begin the Bit Status Monitor mode.

STAT

2. To select 16 Point status.

ENT

3. Else you may select different Status type or Data type using the PREV and NEXT keys.

NEXT | A 0

4. While displaying 16P Status beginning at Y0

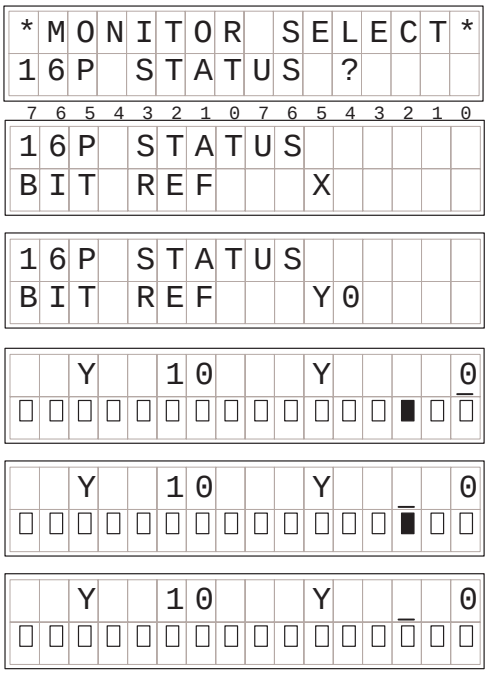
ENT

5. Position cursor and Force Y2 OFF.

← | ← | SHFT | OFF DEL

- Press CLR to exit Bit Forcing function

D2–HPP Display Results



With Y2 forced and the CPU in the Run mode, the executed instructions and result of logic will overwrite the Force command. In other words, if the program logic solves Y2 true, then the output coil Y2 will be turned ON at the end of the scan.

Regular Bit Force with Direct Access

Press these Keystrokes

1. To perform the direct Bit Force ON

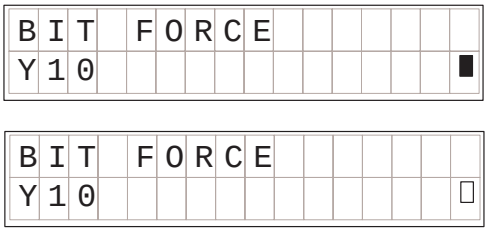
SHFT | Y MLS | B 1 | A 0 | SHFT | ON INS

2. To perform the direct Bit Force OFF

SHFT | Y MLS | B 1 | A 0 | SHFT | OFF DEL

- Press CLR to exit Direct Forcing function

D2–HPP Display Results



Bit Override Indicators

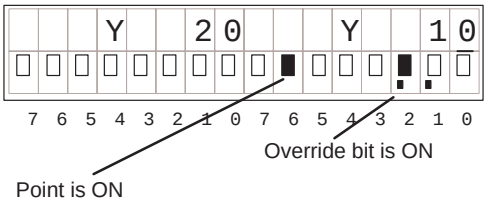
Override bit indicators are also shown on the Handheld programmer status display. Below are the keystrokes to call the status display for Y10 – Y20.

Press these Keystrokes

1. To display the status of Y10 – Y20

STAT | ENT | NEXT | B 1 | A 0 | ENT

D2–HPP Example Display



NOTE: Take care not to confuse the Override Bit marker with the marker used when Test Operations have been set for a point. The Override Bit marker is on the left side below the status bit.

Direct bit Forcing (DL240 ONLY)

The following figures demonstrate how to use Direct Bit Forcing. The Bit force function is helpful to determine if your PLC I/O is responding according to the ON/OFF condition.

NOTE: This example uses Y10 for demonstration purpose. Please insure to use a memory reference which may be forced safely in your PLC. Forcing I/O Bits may change your control program outputs which can cause personal injury or equipment damage on your PLC system.

Press these Keystrokes

1. To Set Bit Override ON and Force Y10 ON

X SET | → | B 1 | A 0 | SHFT | ON INS

2. Set Bit Override OFF and Force Y10 ON

X SET | → | B 1 | A 0 | SHFT | OFF DEL

3. To Reset Bit Override OFF and turn Y10 ON

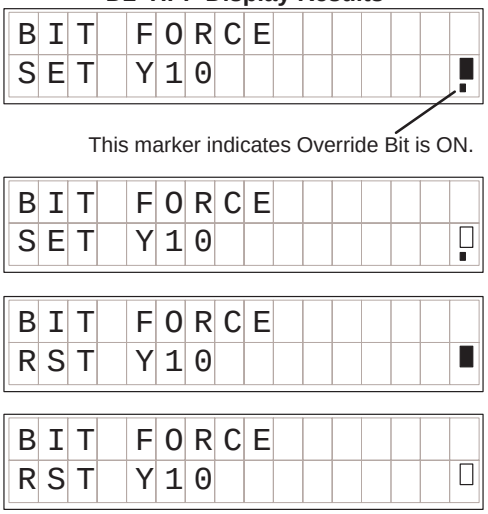
S RST | → | B 1 | A 0 | SHFT | ON INS

4. To Reset Bit Override OFF and turn Y10 OFF

S RST | → | B 1 | A 0 | SHFT | OFF DEL

- In the example above use the NEXT and PREV keys to move to adjacent memory locations.

D2–HPP Display Results



Bit Override (DL240 Only)

The **AUX 59** function may be used to Set or Reset either a single point or a group of data points. The default is the entire data range for the specified data type. To change the default enter a data type and address. The figure below shows how to set the override bit on for Y10 to Y20.

Press these Keystrokes

1. To Select Aux 59 Bit Override Function

F 5 | J 9 | AUX | ENT

2. To Select Area option

→ | ENT

3. Enter Data type and Starting Reference Address or Press ENT to accept displayed defaults

SHFT | Y MLS | B 1 | A 0 | ENT

4. Enter Ending Reference Address or Press ENT to accept displayed default

SHFT | Y MLS | C 2 | A 0 | ENT

5. Use ON / OFF keys to command the override on or off, then press ENT to confirm

SHFT | OFF DEL | ENT

- Press the CLR key to exit Bit Override

D2–HPP Display Results

```
AUX 59 BIT OVRID
PT/AREA
```

```
AUX 59 BIT OVRID
PT/AREA
```

```
AUX 59 BIT OVRID
1st X0000 Y10
```

```
AUX 59 BIT OVRID
END Y0477 Y20
```

```
AUX 59 BIT OVRID
 Y0010-0020 OFF?
```

WARNING: Once again, depending on your application, forcing I/O points may cause unpredictable machine operation that can result in a risk of personal injury or equipment damage. Please take notice how PLC will respond prior to using the force function.

Monitoring V–Memory Locations

You may use the Handheld programmer to monitor and change V memory locations. This is an especially useful feature, since almost all DL105 and DL205 system data is mapped into V memory. The following steps show you how to monitor V-memory locations.

Press these Keystrokes

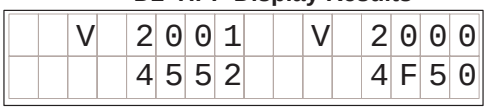
1. Select the location to monitor

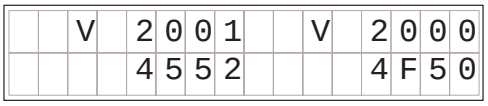
SHFT | V AND | C 2 | A 0 | A 0 | A 0 | STAT

2. Use the PREV and NEXT keys to scroll through adjacent memory locations

NEXT

D2–HPP Display Results





Changing V-Memory Values

Press these Keystrokes

1. Select the location to monitor

SHFT | V AND | C 2 | A 0 | A 0 | A 0 | STAT

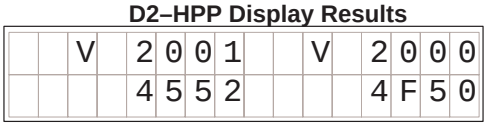
2. Use K (constant) to load a new value in memory location V2000

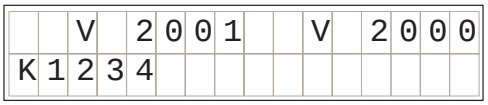
SHFT | K JMP | B 1 | C 2 | D 3 | E 4

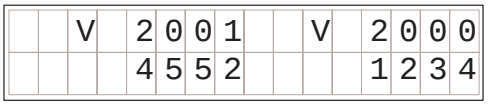
3. Press ENT to enter new value

ENT

D2–HPP Display Results







Monitoring Pointer Locations

Data in V-memory locations may be used to indirectly reference other memory locations (this is also known as using pointers). You may monitor Pointer Memory locations on the Handheld programmer by accessing the “P” data type when using the **STAT** key.

In our example V2000 has the value of 0 and V 2001 has the value of 100 (both values are in octal). At address V0 the value is 1111 and at V100 the value is 2222. When the status display is called with the pointer P2000 the values stored in memory locations V0 and V100 will be displayed, since the addresses stored in V2000 and V2001 point to these respective locations.

Press these Keystrokes

1. To display the status P2000 and P2001

SHFT | P CV | C 2 | A 0 | A 0 | A 0 | STAT

D2–HPP Display Results

			P	2	0	0	1			P	2	0	0	0
				2	2	2	2				1	1	1	1

- For Pointers containing an invalid address, the value displayed on the screen will be “——”.

Monitoring Timer/Counter Values

Timer and Counter current values are mapped into V-memory locations, and may be displayed the same as any V-memory location, the Handheld programmer also provides specialized displays to monitor the status of the Timer and Counter current values and associated status bits. (Appendix A provides a complete listing of the memory map for the DL105 and DL205 PLC’s.

The display for the timer is similar in form to the one shown for the counter.

Press these Keystrokes

1. To display the status of CT16 – CT17

STAT | PREV | PREV | PREV | ENT

NEXT | B 1 | G 6 | ENT

D2–HPP Display Results

		□	C	T		1	7			■	C	T		1	6
				0	0	0	5					0	0	5	0

Counter bit is OFF

Counter bit is ON

Changing Timer/Counter Current Values

To change Timer and Counter current values is much the same as changing V-memory.

Press these Keystrokes

1. To enter a new counter current value

SHFT | K JMP | A 0 | ENT

D2–HPP Display Results

		□	C	T		1	7			□	C	T		1	6
				0	0	0	5					0	0	0	0

Monitoring the CPU Scan Time

The DL105 and DL205 CPU's have a Watchdog Timer that is used to monitor the scan time. The default value set from the factory is 200 ms. If the scan time exceeds the watchdog time limit, the CPU automatically leaves RUN mode and enters the PGM mode. The Handheld programmer displays the following message E003 S/W TIMEOUT when the scan overrun occurs.

You can use **AUX 53** to view the minimum, maximum, and current scan time. Use AUX 55 to increase or decrease the Watchdog timer value.

Press these Keystrokes

1. To call AUX 53 function.

F 5 | D 3 | AUX | ENT

D2–HPP Display Results

```
SCAN   MAX   MIN
0004   0030  0002
```

To Change Watchdog Timer

The CPU must be in PGM, TEST–PGM, or Test–Run mode before you can change the watchdog timer value.

Press these Keystrokes

1. Use AUX 55 to change the watchdog timer value

F 5 | F 5 | AUX | ENT

D2–HPP Display Results

```
AUX 55 WATCHDOG
0200 mSEC
```

Current setting — Entry location

2. Enter the new time value (in milliseconds)

B 1 | A 0 | A 0 | ENT

```
AUX 55 WATCHDOG
OK
```

TEST-RUN and TEST-PGM Modes (DL240 Only)

Test Mode allows you to maintain output status while you switch between TEST-PGM and TEST-RUN Modes and it allows you to trap a value in the middle of program execution. You can select this operation by using the **MODE** key.

The primary benefit of using the TEST mode is to maintain certain outputs and other parameters when the CPU transitions back to Test-Program mode. For example, you can use AUX 58 to configure the individual outputs, CRs, etc. to hold their output state. Also, the CPU will maintain timer and counter current values when it switches to TEST-PGM mode.

Different Test modes are available depending on the mode of operation you are in when make the selection request. If the CPU is in Run Mode mode, then TEST-RUN is available. If the mode is Program, then TEST-PGM is available. Once you've selected the TEST Mode, you may switch between TEST-RUN and TEST-PGM modes. The LED on the Handheld programmer is on while in the Test Mode. The following figure shows how to select the Test Mode, while in the Run mode.

Press these Keystrokes

D2–HPP Display Results

1. To go to Test–Run mode

MODE | NEXT | ENT

```
*MODE CHANGE*
GO TO PGM MODE
```

2. Press ENT to confirm TEST-RUN Mode

ENT

```
*MODE CHANGE*
GO TO T-RUN MODE
```

- The TEST LED on the Handheld programmer indicates that the CPU is in TEST Mode.

```
*MODE CHANGE*
CPU T-RUN
```

Begin this example in PROGRAM Mode

1. You can return to Run Mode, enter Program Mode, or enter TEST-PGM Mode by using the Mode Key

CLR | MODE | NEXT | NEXT | ENT

```
*MODE CHANGE*
GO TO RUN MODE
```

2. Press ENT to confirm TEST-PGM Mode

ENT

```
*MODE CHANGE*
CPU T-PGM
```

(Note, the TEST LED on the Handheld indicates that the CPU is in TEST Mode.)

- Press the CLR key to exit Mode change.

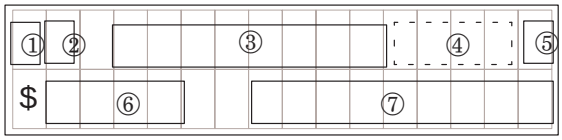
WARNING: The following items should be considered during Run Time Edits.
1. If he program has any instruction syntax errors, the CPU will *not* enter the Run Mode.
2. If you delete an output reference while the output is ON, the output will remain ON until it is forced OFF with a programming device.
3. Input point changes are not acknowledged during Run Time Edits. So, if you're using a high-speed operation and a critical input comes on, the CPU may not see the change.

Test Displays

With the Handheld Programmer you also have a more detailed display when you use TEST Mode. The areas which are active are dependant on the instruction being displayed. For most instructions, the TEST-RUN mode display is more detailed than the status displays shown in RUN mode.

TEST-RUN Displays

With the Handheld programmer in the Test–Run mode and the instruction addresses displayed, various groups of information are available. The different groups of information are labeled and described below.



① Displays the power flow through the instruction just after the instruction is executed.
■ indicates power flow and **Y** indicates no power flow.

② Displays the power flow of the power rail.
■ indicates power flow and **M** indicates no power flow.

③ Displays the contents of the following (where applicable to the instruction):
– the accumulator
– the timer current value
– the counter current value

④ If the operand is a data register, this field displays the contents of the data register.

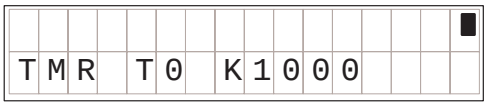
⑤ If the operand is a bit, this field displays the bit status.
■ indicates ON and **S** indicates OFF

⑥ Displays the instruction address.

⑦ Displays the mnemonic instruction and reference number

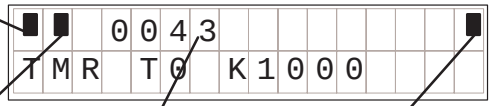
T0 Contact (S is OFF)
(■ is ON)

Example RUN Mode Display



Input to Timer

Example TEST-RUN Mode



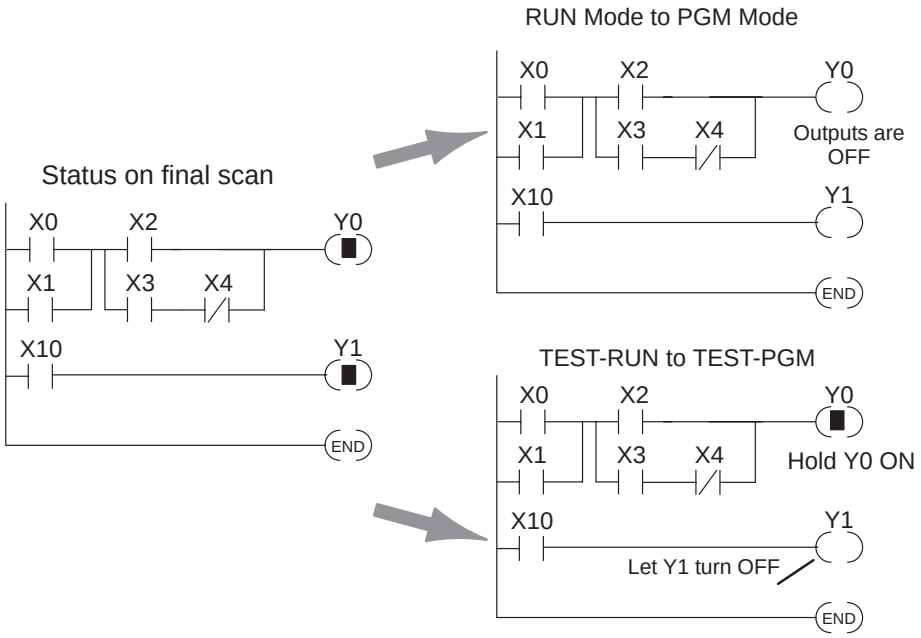
Power flow to rail

Current Value

T0 Contact (S is off)
(■ is on)

Holding Output States

The ability to hold output states allows you to maintain key system I/O points. In some cases you may need to modify the program, but you don't want certain operations to stop. In normal Run Mode, the outputs are turned off when you return to Program Mode. In TEST-RUN mode individual outputs can be set to hold the last output state on the transition to TEST-PGM mode. This is done with **AUX 58** on the Handheld programmer. The following diagram shows the differences between RUN and TEST-RUN modes.



Before you decide that Test Mode is the perfect choice, remember that the DL205 CPUs also allow you to edit the program during Run Mode. The primary difference between the Test Modes and the Run Time Edit feature is that you do not have to configure each individual I/O point to hold the output status. When you use Run Time Edits, the CPU automatically maintains all outputs in their current states while the program is being updated. Run Time Edits should only be performed by trained personnel.

The following is an example of using **AUX 58** to configure the output state for Y15 to Y25 when the CPU transitions from TEST-RUN to TEST-PGM.

Using the Test Operation

Press these Keystrokes

D2–HPP Display Results

1. Select AUX 58 Test Operation.

[F 5] [I 8] [AUX] [ENT]

```
AUX 58 TEST OPER
PT/AREA
```

2. Select AREA to test

[→] [ENT]

```
AUX 58 TEST OPER
1st Y
```

3. Enter the first address

[B 1] [F 5] [ENT]

```
AUX 58 TEST OPER
1st Y15
```

4. Enter the ending address

[C 2] [F 5] [ENT]

```
AUX 58 TEST OPER
END Y25
```

5. Use ON / OFF keys to command the override on or off
Then press ENT to confirm

[ENT]

```
AUX 58 TEST OPER
Y0015-0025 ON?
```

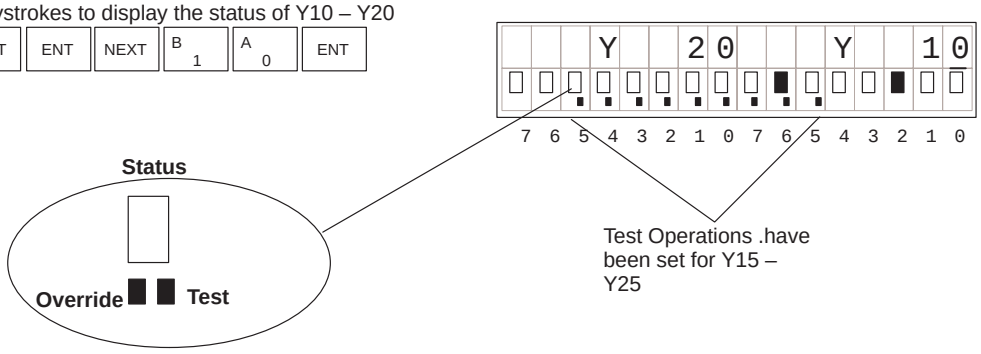
- Press the CLR key to leave AUX 58

Test Operation Indicators

The Test Operation indicators may be displayed on the Handheld programmer during Status Monitor mode. Below are the keystrokes to call the status display for Y10 – Y20.

1. Keystrokes to display the status of Y10 – Y20

[STAT] [ENT] [NEXT] [B 1] [A 0] [ENT]

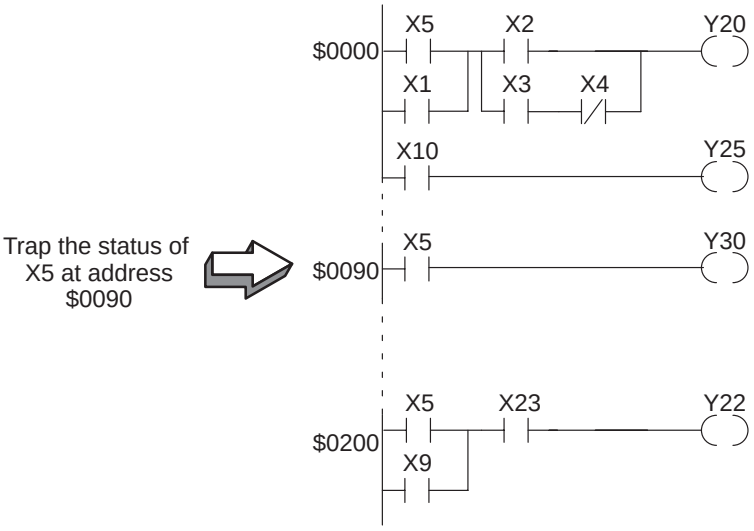


NOTE: Take care not to confuse the Test Operation marker with the marker used for Bit Override. The Test marker is a small box indicated on the right side below the Status point.

Trapping a Discrete Point or Word of Data (DL240 Only)

With the DL240 CPU, you may use the TEST mode to trap the status of a point or word during normal program execution. To use this feature you must select the memory location and address in the program where you wish to check the contents of the memory location.

For example you may read X5 three times during a program scan (at address $0000, $0090 and $0200) and you want to know what the status of X5 is at address $0090.



This example only works in the TEST-RUN Mode (DL240 Only)

Press these Keystrokes

1. Use the STAT key to trap the status

STAT NEXT NEXT ENT

2. Enter the program address to trap on

J 9 A 0 ENT

3. Enter the memory location to trap.

F 5 ENT

- PREV / NEXT keys can be used to scroll through the valid data types

D2–HPP Display Results

```
*MONITOR  SELECT*
TRAP 16PT  STATUS
```

```
TRAP 16PT  STATUS
PGM  ADDR  90
```

```
TRAP 16PT  STATUS
BIT  REF   X5
```

```
   X   10   X   0
□□□■□□□□□□□■□□■□□
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
```

To trap a word of memory instead of a discrete point use the status monitoring option of TRAP WORD STATUS and enter the parameters the same way as in the example above.

I/O Diagnostics

Diagnostics (DL205 ONLY)

The DL205 system provides diagnostic features to that help identify I/O errors. **AUX 42** will report missing modules and new I/O configuration. For the DL240 CPU I/O communication errors will also be reported. The error codes are listed later in this chapter, and shows all of the possible I/O error messages. This function is only possible with the DL205. If present, the error display will automatically be displayed when Handheld programmer is connected to the CPU. If an I/O error occurs, use the follow example figure below to determine which base and slot has failed.

Press these Keystrokes

1. Select the AUX 42 I/O Diagnostics
   [E 4] [C 2] [AUX]
2. To run the diagnostics
   [ENT]
3. Use arrow keys to see more information
   [→]

Example Error Display

```
E252
NEW I/O CFG
```

D2–HPP Display Results

```
AUX 4* I/O CFG
AUX 42 I/O DIAGN
```

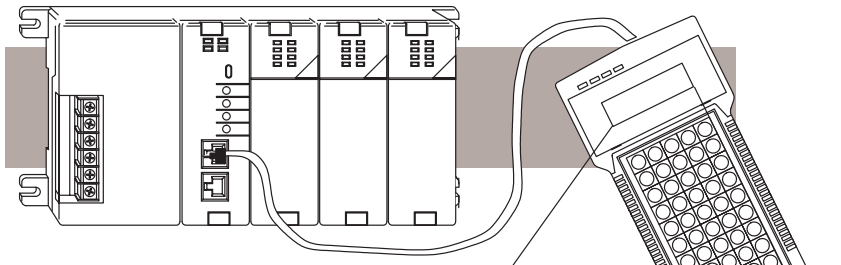
```
AUX 42 I/O BASE
E252   I/O CONF
```

```
/O BASE 0/SLOT3
I/O CONFIG Error
```

If the PLC system detects a change in the I/O configuration at power-up or an I/O fault, and error message will be displayed. To get more detail on the location of the error, internal diagnostic locations also exist which specify the module type, module location and an error code.

The following figure shows a example of the failure indicators. The table on the left lists the module codes which may be displayed in the failure indicators.

Code (Hex)	Component Type
04	CPU
03	I/O Base
20	8 pt. Output
21	8 pt. Input
24	4input/output combination
28	12 pt. Output, 16 pt. Output
2B	16 pt. Input
36	Analog Input
37	Analog Output
4A	Counter Interface
7F	Abnormal
FF	No module detected



Program Control Information

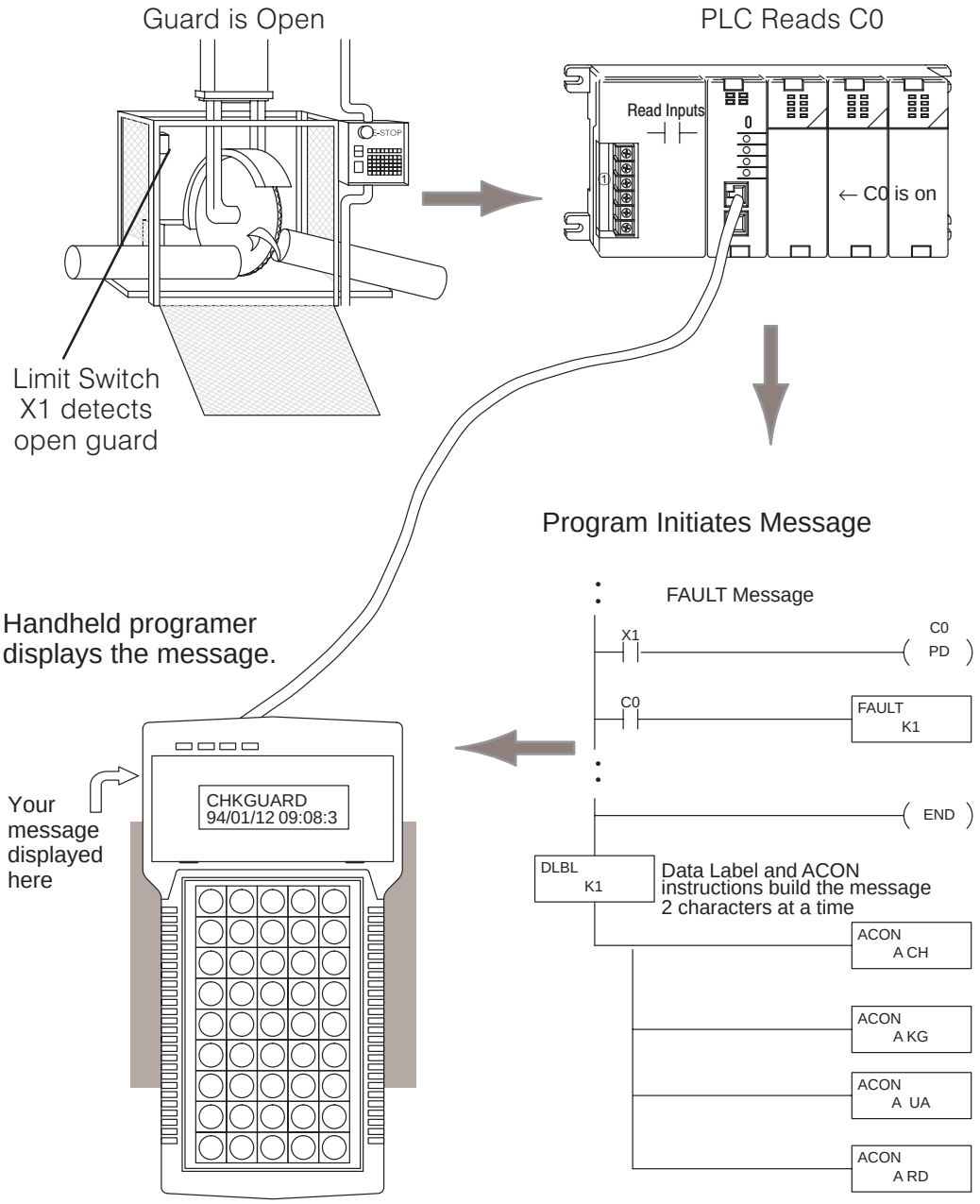
V7752	0020	Desired module ID code
V7753	0021	Current module ID code
V7754	0002	Location of conflict
V7755	0252	Fatal error code
SP47	⊣⊢	I/O configuration Error

E252
NEW I/O CFG

Custom Messages

The FAULT message instruction may be used to log messages which can be view with **AUX 5C** on the Handheld programmer. The Fault messages must be triggered with a positive differential (one shot) instruction, otherwise the message log buffer will repeat storing the same message over and over.

The following figure shows how the message display capability works.

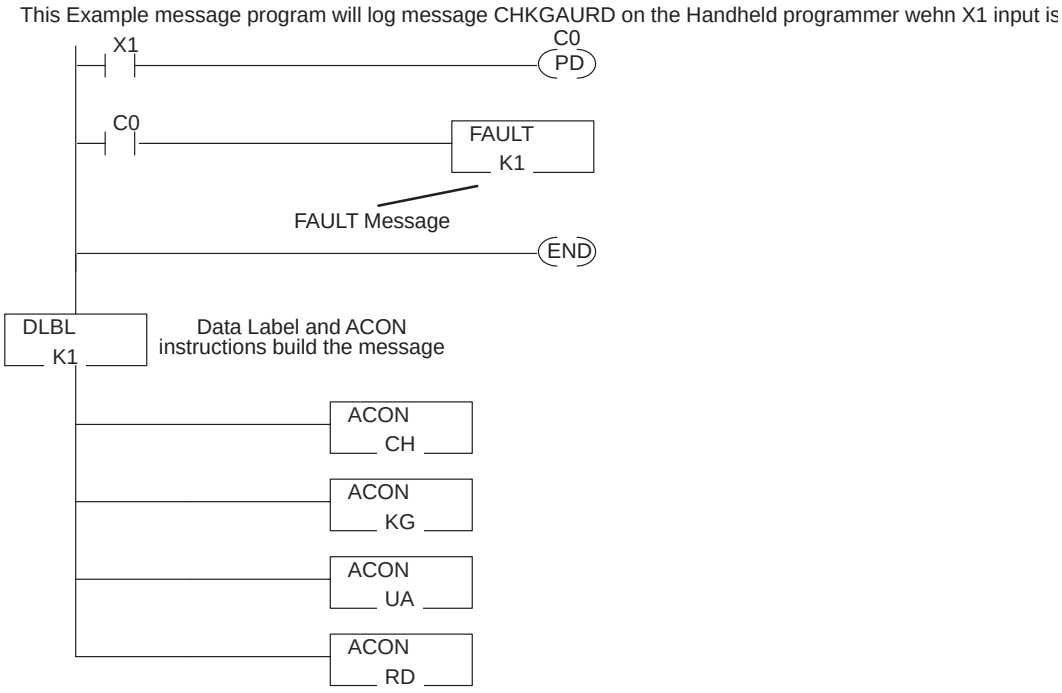


Message Instructions

A total of 64 Message instructions may be programmed. The messages can be up to 23 characters in length and contain both text and numeric values. These messages are part of the RLL program and are displayed automatically on the Handheld Programmer during RUN mode.

There are several instructions that are used to build operator messages. Detailed explanations of the following instructions are included in the DL105 and DL205 User Manual.

- FAULT — the Fault instruction is an output box instruction that lets the program know which message to display.
- DLBL — the Data Label instruction is included *after* the END statement and notes the beginning of a message.
- ACON — the ASCII Constant instruction is used as an output box for the ASCII portion of the message. (You can also display the contents of a V-memory location instead of ASCII text.)
- NCON — the Numeric Constant instruction is used as an output box for any numeric constant portion of the message.
- MOVMC (DL240 only) — the Move Data Label to V-memory Area instruction is used to embed variables, such as timer or counter values, into a text message.



NOTE: It is *much* easier to enter text message programs with ***Direct***SOFT™ than it is with the Handheld Programmer. This is because you can only enter two ASCII characters per ACON instruction with the Handheld programmer. This is not the case with ***Direct***SOFT, which allows you to enter several per ACON instruction. ***Direct***SOFT™ also supports other characters not available on the handheld keypad.

Message Program Example

The following example figure demonstrates how to program a message using the Handheld Programmer. Once you've entered the program, put the CPU in RUN mode and force X1 ON to log the message.

Press these Keystrokes

1. Enter the first contact

[$ STR] [→] [B 1] [ENT]

2. Enter the PD (always use the one shot when generating FAULT messages)

[SHFT] [P CV] [SHFT] [D 3]
[→] [A 0] [ENT]

3. Enter the control relay

[$ STR] [→] [NEXT] [NEXT] [A 0] [ENT]

4. Enter the FAULT instruction

[SHFT] [F 5] [A 0] [U ISG] [L ANDST] [T MLR]
[→] [B 1] [ENT]

5. Enter the END statement

[SHFT] [E 4] [N TMR] [D 3] [ENT]

6. Enter the DLBL instruction

[SHFT] [D 3] [L ANDST] [B 1] [L ANDST]
[→] [B 1] [ENT]

7. Enter the ACON instruction and the first two letters

[SHFT] [A 0] [C 2] [O INST#] [N TMR]
[→] [SHFT] [C 2] [H 7]

8. Enter the ACON instruction and the next two letters

[SHFT] [A 0] [C 2] [O INST#] [N TMR]
[→] [SHFT] [K JMP] [G 6]

9. Enter the ACON instruction and the next two letters

[SHFT] [A 0] [C 2] [O INST#] [N TMR]
[→] [SHFT] [U ISG] [A 0]

D2–HPP Display Results

```
STR X1
NOP
```

```
PD C0
NOP
```

```
STR C0
NOP
```

```
FAULT K1

```

```
END
NOP
```

```
DLBL K1
NOP
```

```
ACON ACH
NOP
```

```
ACON AKG
NOP
```

```
ACON AUA
NOP
```

```
ACON ARD
NOP
```

Checking the Error Message Tables

Two Types of Tables

The DL240 CPU will automatically log any system error codes and custom messages created with the FAULT instructions. The CPU logs the error code, the date, and the time the error occurred. There are two separate tables that store this information.

- Error Code Table – the system logs up to 32 errors in the table. When an error occurs, the errors already on the table are pushed down and the most recent error is loaded into the top slot. If the table is full when an error occurs, the oldest error is pushed out (erased) from the table.
- Message Table – the system logs up to 16 messages in this table. When a message is triggered, the messages already stored in the table are pushed down and the most recent message is loaded into the top slot. If the table is full when an error occurs, the oldest message is pushed out (erased) of the table.

The following diagram shows an example of an error table for messages.

Date	Time	Message
1996–01–26	08:41:51:11	*Conveyor–2 stopped
1996–02–30	17:01:11:56	* Conveyor–1 stopped
1996–02–30	17:01:11:12	* Limit SW1 failed
1996–02–28	03:25:14:31	* Saw Jam Detect

Viewing the Error Table

The Handheld programmer maintains a history of Errors and Messages. You may display the Errors and Messages on the Handheld programmer by using the **AUX 5C** function. The figure below demonstrates how to use the AUX 5C function.

Press these Keystrokes

1. Use AUX 5C to view Errors

F 5 | SHFT | C 2 | AUX | ENT

2. Press ENT to select Error Messages

ENT

- The most recent error is displayed. You can also use the **PREV** and **NXT** keys to sequentially step through the errors. The arrow keys can be used to scroll the display for more detail.

D2–HPP Display Results

```
AUX 5C HISTORY D
   ERROR/MESAGE
```

```
E252NEW I/O CFG
94/01/14 13:18:4
```

Viewing the Message Table

The **AUX 5C** function, is used to view messages on the Handheld programmer.

Press these Keystrokes

1. Use AUX 5C to view FAULT messages

F 5 | SHFT | C 2 | AUX | ENT

2. Use the arrow key to select MESSAGE

→ | ENT

- The most recent message is displayed. You can also use the **PREV** and **NXT** keys to sequentially step through the messages. The arrow keys can be used to scroll the display for more detail.

D2–HPP Display Results

```
AUX 5C HISTORY D
   ERROR/MESAGE
```

```
CHKGUARD
94/01/14 13:35:2
```

Error Codes

The following table lists the error codes that may appear on the D2–HPP Handheld programmer.

DL105/DL205 Error Code	Description
E003 SOFTWARE TIME-OUT	If the program scan time exceeds the time allotted to the watchdog timer, this error will occur. SP51 will be on and the error code will be stored in V7755. To correct this problem add RSTWT instructions in FOR NEXT loops and subroutines or use AUX 55 to extend the time allotted to the watchdog timer.
041 CPU BATTERY LOW	The CPU battery is low and needs replacement. SP43 will be on and the error code will be stored in V7757.
EE099 PROGRAM MEMORY EXCEEDED	If the compiled program length exceeds the amount of available CPU RAM this error will occur. SP52 will be on and the error code will be stored in V7755. Reduce the size of the application program.
E104 WRITE FAILED	A write to the CPU was not successful. Disconnect the power, remove the CPU, and make sure the EEPROM is not write protected. If the EEPROM is not write protected, make sure the EEPROM is installed correctly. If both conditions are OK, replace the CPU.
E151 BAD COMMAND	A parity error has occurred in the application program. SP44 will be on and the error code will be stored in V7755. This problem may possibly be due to electrical noise. Clear the memory and download the program again. Correct any grounding problems. If the error returns replace the EEPROM or the CPU.
E155 RAM FAILURE	A checksum error has occurred in the system RAM. SP44 will be on and the error code will be stored in V7755. This problem may possibly be due to a low battery, electrical noise or a CPU RAM failure. Clear the memory and download the program again. Correct any grounding problems. If the error returns replace the CPU.
E202 MISSING I/O MODULE	An I/O module has failed to communicate with the CPU or is missing from the base. SP45 will be on and the error code will be stored in V7756. Run AUX42 to determine the slot and base location of the module reporting the error.
E210 POWER FAULT	A short duration power drop-out occurred on the main power line supplying power to the base.
E250 COMMUNICATION FAILURE IN THE I/O CHAIN	A failure has occurred in the local I/O system. The problem could be in the base I/O bus or the base power supply. SP45 will be on and the error code will be stored in V7755. Run AUX42 to determine the base location reporting the error.
E252 NEW I/O CFG	This error occurs when the auto configuration check is turned on in the CPU and the actual I/O configuration has changed either by moving modules in a base or changing types of modules in a base. You can return the modules to the original position/types or run AUX45 to accept the new configuration. SP47 will be on and the error code will be stored in V7755.
E262 I/O OUT OF RANGE	An out of range I/O address has been encountered in the application program. Correct the invalid address in the program. SP45 will be on and the error code will be stored in V7755.

DL105/DL205 Error Code	Description
E312 HP COMM ERROR 2	A data error was encountered during communications with the CPU. Clear the error and retry the request. If the error continues check the cabling between the two devices, replace the Handheld programmer, then if necessary replace the CPU. SP46 will be on and the error code will be stored in V7756.
E313 HP COMM ERROR 3	An address error was encountered during communications with the CPU. Clear the error and retry the request. If the error continues check the cabling between the two devices, replace the Handheld programmer, then if necessary replace the CPU. SP46 will be on and the error code will be stored in V7756.
E316 HP COMM ERROR 6	A mode error was encountered during communications with the CPU. Clear the error and retry the request. If the error continues replace the Handheld programmer, then if necessary replace the CPU. SP46 will be on and the error code will be stored in V7756.
E320 HP COMM TIME-OUT	The CPU did not respond to the Handheld programmer communication request. Check to ensure cabling is correct and not defective. Power cycle the system if the error continues replace the CPU first and then the Handheld programmer if necessary.
E321 COMM ERROR	A data error was encountered during communication with the CPU. Check to ensure cabling is correct and not defective. Power cycle the system and if the error continues replace the CPU first and then the Handheld programmer if necessary.
E352 BACKGROUND COMM. ERROR	Communications error between CPU and intelligent module. Incorrect slot reference while attempting to use the READ/WRITE commands e.g. DCM module interface. The slot number of module which I/O error occured is stored in V7760–V7764.
E4** NO PROGRAM	A syntax error exists in the application program. The most common is a missing END statement. Run AUX21 to determine which one of the E4** series of errors is being flagged. SP52 will be on and the error code will be stored in V7755.
E401 MISSING END STATEMENT	All application programs must terminate with an END statement. Enter the END statement in appropriate location in your program. SP52 will be on and the error code will be stored in V7755.
E402 MISSING LBL	A GOTO, GTS, MOVMC or LDLBL instruction was used without the appropriate label. Refer to the programming manual for details on these instructions. SP52 will be on and the error code will be stored in V7755.
E403 MISSING RET (DL240 ONLY)	A subroutine in the program does not end with the RET instruction. SP52 will be on and the error code will be stored in V7755.
E404 MISSING FOR (DL240 ONLY)	A NEXT instruction does not have the corresponding FOR instruction. SP52 will be on and the error code will be stored in V7755.

DL105/DL205 Error Code	Description
E405 MISSING NEXT (DL240 ONLY)	A FOR instruction does not have the corresponding NEXT instruction. SP52 will be on and the error code will be stored in V7755.
E406 MISSING IRT	An interrupt routine in the program does not end with the IRT instruction. SP52 will be ON and the error code will be stored in V7755.
E412 SBR/LBL>64 (DL240 ONLY)	There is greater than 64 SBR, LBL or DLBL instructions in the program. This error is also returned if there is greater than 128 GTS or GOTO instructions used in the program. SP52 will be on and the error code will be stored in V7755.
E413 FOR/NEXT>64 (DL240 ONLY)	There is greater than 64 FOR/NEXT loops in the application program. SP52 will be on and the error code will be stored in V7755.
E421 DUPLICATE STAGE REFERENCE	Two or more SG or ISG labels exist in the application program with the same number. A unique number must be reserved for each Stage and Initial Stage. SP52 will be on and the error code will be stored in V7755.
E422 DUPLICATE SBR/LBL REFERENCE	Two or more SBR or LBL instructions exist in the application program with the same number. A unique number must be allowed for each Subroutine and Label. SP52 will be on and the error code will be stored in V7755.
E423 NESTED LOOPS (DL240 ONLY)	Nested loops (programming one FOR/NEXT loop inside of another) is not allowed in the DL240 series. SP52 will be on and the error code will be stored in V7755.
E431 INVALID ISG/SG ADDRESS	An ISG or SG must not be programmed after the end statement such as in a subroutine. SP52 will be on and the error code will be stored in V7755.
E432 INVALID JUMP (GOTO) ADDRESS (DL240 ONLY)	A LBL that corresponds to a GOTO instruction must not be programmed after the end statement such as in a subroutine. SP52 will be on and the error code will be stored in V7755.
E433 INVALID SBR ADDRESS (DL240 ONLY)	A SBR must be programmed after the end statement, not in the main body of the program or in an interrupt routine. SP52 will be on and the error code will be stored in V7755.
E435 INVALID RT ADDRESS (DL240 ONLY)	A RT must be programmed after the end statement, not in the main body of the program or in an interrupt routine. SP52 will be on and the error code will be stored in V7755.

DL105/DL205 Error Code	Description
E436 INVALID INT ADDRESS	An INT must be programmed after the end statement, not in the main body of the program. SP52 will be on and the error code will be stored in V7755.
E438 INVALID IRT ADDRESS	An IRT must be programmed after the end statement, not in the main body of the program. SP52 will be on and the error code will be stored in V7755.
E440 INVALID DATA ADDRESS	Either the DLBL instruction has been programmed in the main program area (not after the END statement), or the DLBL instruction is on a rung containing input contact(s).
E441 ACON/NCON (DL240 ONLY)	An ACON or NCON must be programmed after the end statement, not in the main body of the program. SP52 will be on and the error code will be stored in V7755.
E451 BAD MLS/MLR	MLS instructions must be numbered in ascending order from top to bottom.
E452 X AS COIL	An X data type is being used as a coil output.
E453 MISSING T/C	A timer or counter contact is being used where the associated timer or counter does not exist.
E454 BAD TMRA	One of the contacts is missing from a TMRA instruction.
E455 BAD CNT	One of the contacts is missing from a CNT or UDC instruction.
E456 BAD SR	One of the contacts is missing from the SR instruction.
E461 STACK OVERFLOW	More than nine levels of logic have been stored on the stack. Check the use of OR STR and AND STR instructions.
E462 STACK UNDERFLOW	An unmatched number of logic levels have been stored on the stack. Ensure the number of AND STR and OR STR instructions match the number of STR instructions.
E463 LOGIC ERROR	A STR instruction was not used to begin a rung of ladder logic.
E464 MISSING CKT	A rung of ladder logic is not terminated properly.
E471 DUPLICATE COIL REFERENCE	Two or more OUT instructions reference the same I/O point.
E472 DUPLICATE TMR REFERENCE	Two or more TMR instructions reference the same number.

DL105/DL205 Error Code	Description
E473 DUPLICATE CNT REFERENCE	Two or more CNT instructions reference the same number.
E480 INVALID CV ADDRESS	The CV instruction is used in a subroutine or program interrupt routine. The CV instruction may only be used in the main program area (before the END statement).
E481 CONFLICTING INSTRUCTIONS	An instruction exists between convergence stages.
E482 MAX. CV INSTRUCTIONS EXCEEDED	Number of CV instructions exceeds 17.
E483 INVALID CVJMP ADDRESS	CVJMP has been used in a subroutine or a program interrupt routine.
E484 MISSING CV INSTRUCTION	CVJMP is not preceded by the CV instruction. A CVJMP must immediately follow the CV instruction.
E485 NO CVJMP	A CVJMP instruction is not placed between the CV and the SG, ISG, BLK, BEND, END instruction.
E486 INVALID BCALL ADDRESS	A BCALL is used in a subroutine or a program interrupt routine. The BCALL instruction may only be used in the main program area (before the END statement).
E487 MISSING BLK INSTRUCTION	The BCALL instruction is not followed by a BLK instruction.
E488 INVALID BLK ADDRESS	The BLK instruction is used in a subroutine or a program interrupt. Another BLK instruction is used between the BCALL and the BEND instructions.
E489 DUPLICATED CR REFERENCE	The control relay used for the BLK instruction is being used as an output elsewhere.

DL105/DL205 Error Code	Description
E490 MISSING SG INSTRUCTION	The BLK instruction is not immediately followed by the SG instruction.
E491 INVALID ISG INSTRUCTION ADDRESS	There is an ISG instruction between the BLK and BEND instructions.
E492 INVALID BEND ADDRESS	The BEND instruction is used in a subroutine or a program interrupt routine. The BEND instruction is not followed by a BLK instruction.
E493 MISSING REQUIRED INSTRUCTION	A [CV, SG, ISG, BLK, BEND] instruction must immediately follow the BEND instruction.
E494 MISSING BEND INSTRUCTION	The BLK instruction is not followed by a BEND instruction.
E501 BAD ENTRY	An invalid keystroke or series of keystrokes were entered into the Handheld programmer.
E502 BAD ADDRESS	An invalid or out of range address was entered into the Handheld programmer.
E503 BAD COMMAND	An invalid instruction was entered into the Handheld programmer.
E504 BAD REF/VAL	An invalid value or reference number was entered with an instruction.
E505 INVALID INSTRUCTION	An invalid instruction was entered into the Handheld programmer.
E506 INVALID OPERATION	An invalid operation was attempted by the Handheld programmer.
E520 BAD OP–RUN	An operation which is invalid in the RUN mode was attempted by the Handheld programmer.
E521 BAD OP–TRUN	An operation which is invalid in the TEST RUN mode was attempted by the Handheld programmer.
E523 BAD OP–TPGM	An operation which is invalid in the TEST PROGRAM mode was attempted by the Handheld programmer.
E524 BAD OP–PGM	An operation which is invalid in the PROGRAM mode was attempted by the Handheld programmer.

DL105/DL205 Error Code	Description
E525 MODE SWITCH (DL240 ONLY)	An operation was attempted by the Handheld programmer while the CPU mode switch was in a position other than the TERM position.
E526 OFF LINE	The Handheld programmer is in the OFFLINE mode. To change to the ONLINE mode use the MODE the key.
E527 ON LINE	The Handheld programmer is in the ON LINE mode. To change to the OFF LINE mode use the MODE the key.
E528 CPU MODE	The operation attempted is not allowed during a Run Time Edit.
E540 CPU LOCKED	The CPU has been password locked. To unlock the CPU use AUX82 with the password.
E541 WRONG PASSWORD	The password used to unlock the CPU with AUX82 was incorrect.
E542 PASSWORD RESET	The CPU powered up with an invalid password and reset the password to 00000000. A password may be re-entered using AUX81.
E601 MEMORY FULL	Attempted to enter an instruction which required more memory than is available in the CPU.
E602 INSTRUCTION MISSING	A search function was performed and the instruction was not found.
E604 REFERENCE MISSING	A search function was performed and the reference was not found.
E610 BAD I/O TYPE	The application program has referenced an I/O module as the incorrect type of module.
E620 OUT OF MEMORY	An attempt to transfer more data between the CPU and Handheld programmer than the receiving device can hold.
E621 EEPROM NOT BLANK	An attempt to write to a non-blank EEPROM was made. Erase the EEPROM and then retry the write.
E622 NO HPP EEPROM	A data transfer was attempted with no EEPROM (or possibly a faulty EEPROM) installed in the Handheld programmer.
E623 SYSTEM EEPROM	A function was requested with an EEPROM which contains system information only.
E624 V-MEMORY ONLY	A function was requested with an EEPROM which contains V-memory data only.
E625 PROGRAM ONLY	A function was requested with an EEPROM which contains program data only.

DL105/DL205 Error Code	Description
E627 BAD WRITE	An attempt to write to a write protected or faulty EEPROM was made. Check the write protect jumper and replace the EEPROM if necessary.
E628 EEPROM TYPE ERROR	The wrong size EEPROM is being used. The DL230 and DL240 CPUs use different size EEPROMs.
E640 COMPARE ERROR	A compare between the EEPROM and the CPU was found to be in error.
E650 HPP SYSTEM ERROR	A system error has occurred in the Handheld programmer. Power cycle the Handheld programmer. If the error returns replace the Handheld programmer.
E651 HPP ROM ERROR	A ROM error has occurred in the Handheld programmer. Power cycle the Handheld programmer. If the error returns replace the Handheld programmer.
E652 HPP RAM ERROR	A RAM error has occurred in the Handheld programmer. Power cycle the Handheld programmer. If the error returns replace the Handheld programmer.