Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

Product Family:	DL205	**Date**:	October 2018
Manual Number	D2-HP-M		
Revision and Date	1st Edition, Rev. A; May, 1998		

Changes to Chapter 1. Getting Started

Pages 1-6 and 1-7. Physical Characteristics and Specifications; Connections to the CPU and Specifications

On both of these pages the part number given for the programming cable is incorrect. The correct part number is “DV-1000CBL”, not “D2-DSCBL”.

Page 1-6. Handheld Programmer Layout

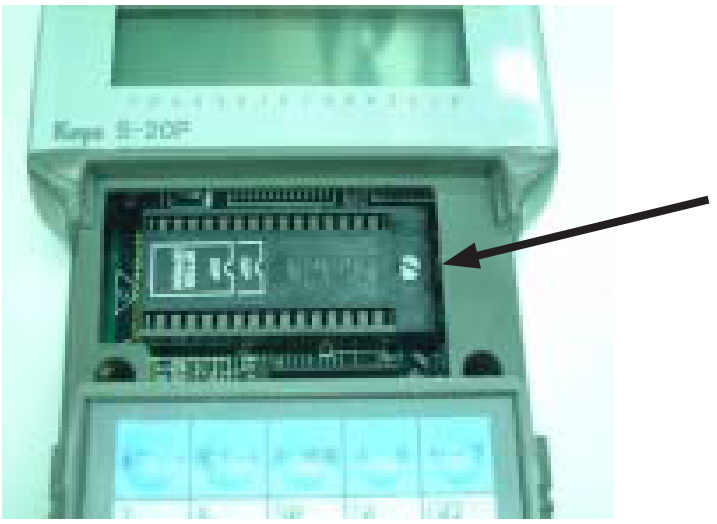
Changes to Chapter 5. Naming and Storing Program; Saving Programs to EEPROM

Page 5-4. Types of EEPROMs (DL205 Only)

Page 5-5. Inserting a EEPROM in the Handheld Programmer

In late 2004, a design change occured that changed how the EEPROM locks into place. The drawings and text on pages 1-6, 5-4 and 5-5 show the old design, whereby the EEPROM was held into place by a small lever. To replace the EEPROM, raise the lever to loosen the EEPROM. Once the replacement EEPROM is inserted in place, press the lever down to secure it.

On re-designed models, the lever arrangement was replaced by a locking screw, as shown in the photo below. Turn the screw clockwise to loosen and counterclockwise to tighten the EEPROM.



Locking screw secures EEPROM in place. Locking screw replaces lever on models manufactured after October 2004.

D2-HPP Handheld Programmer

Manual Number D2-HP-M

WARNING

Thank you for purchasing automation equipment from PLC***Direct***™. We want your new ***Direct***LOGIC™ automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 1-800-633-0405.

This publication is based on information that was available at the time it was printed. At PLC***Direct***™ we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. PLC***Direct***™ disclaims any proprietary interest in the marks and names of others.

Stage is a trademark of Koyo Electronics Industries Co., LTD. Think & Do Software is a trademark of Think & Do Software, Inc. Texas Instruments is a registered trademark of Texas Instruments, Inc. TI, TIWAY, Series 305, Series 405, TI305, and TI405 are trademarks of Texas Instruments, Inc. Siemens and SIMATIC are registered trademarks of Siemens, AG. GE is a registered trademark of General Electric Corporation. Series One is a registered trademark of GE Fanuc Automation North America, Inc. MODBUS is a registered trademark of Gould, Inc. IBM is a registered trademark of International Business Machines. MS-DOS and Microsoft are registered trademarks of Microsoft Corporation. Windows and Windows NT are trademarks of Microsoft Corporation. OPTOMUX and PAMUX are trademarks of OPTO 22.

Manual Revisions

If you contact us in reference to this manual, be sure and include the revision number.

Title: D2-HPP Handheld Programmer
Manual Number: D2-HP-M

Issue	Date	Effective Pages	Description of Changes
Original	1/94	Complete User Manual	Original Issue
2nd Edition	7/96	New Release	Upgraded Firmware for DL105 & DL205
Rev A	5/98	Manual Revisions Pages 2-10, 3-7, 4-7	Made minor corrections before reprinting

Table of Contents

Chapter 1: Getting Started

Chapter 2: D2-HPP Setup

Chapter 3: Entering Programs

Chapter 4: Changing Programs

Chapter 5: Naming and Storing Programs

Chapter 6: System Monitoring and Troubleshooting

Appendix A: DL105/DL205 Memory Map

Appendix B: Special Relays

1

Getting Started

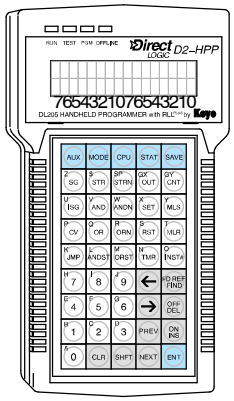
In This Chapter. . . .

- Introduction
- How can I use the Handheld?
- Physical Characteristics and Specifications
- Keypad Layout
- Mode Indicators
- Display Panel

Introduction

D2-HPP Handheld Programmer

The D2-HPP (Handheld Portable Programmer) is a general purpose tool for use with the DL105 or DL205 PLC products. It is well suited for performing basic PLC maintenance and troubleshooting of machine automation equipment. The Handheld programmer is *not* ideal for entering large complex PLC programs. In this case please consider using ***Direct***SOFT™, our PC-based programming software.



Purpose of this manual

This manual provides information on the D2-HPP capabilities and how to operate the Handheld programmer. Although this manual does *not* cover all instructions possible with the Handheld programmer, it should detail all key features and how they should be used.

Who should read this manual

This manual *is* a reference manual for the D2-HPP Handheld programmer, *not* a tutorial on the DL105/DL205 instruction set or system operations. It is intended for new user to become familiar with using the D2-HPP features and functions.

Supplemental Manuals

The DL105 and DL205 User Manuals may occasionally be referenced by this manual. As you become more efficient with the Handheld Programmer, this manual may not be absolutely necessary, but it may useful as a reference on procedures and related subjects.

Technical Support

We realize that even though we strive to be the best, we may have arranged our information in such a way you cannot find what you are looking for. First, check these resources for help in locating the information:

- **Table of Contents** - chapter and section listing of contents, in the front of this manual
- **Quick Guide to Contents** - chapter summary listing on the following page
- **Appendices** - reference material for key topics
- **Index** - alphabetical listing of key words, at the end of this manual

You can also check our online resources for the latest product support information:

- **Internet** - the address of our Web site is http://www.plcdirect.com
- **Bulletin Board Service (BBS)** - call (770)-844-4209

If you still need assistance, please call us at 800-633-0405. Our technical support group is glad to work with you in answering your questions. They are available Monday through Friday from 9:00 A.M. to 6:00 P.M. Eastern Standard Time. If you have a comment or question about any of our products, services, or manuals, please fill out and return the 'Suggestions' card that was shipped with this manual.

Chapters The main contents of this manual are organized into the following six chapters:

	**Getting Started**	provides an overview of the Handheld Programmer and provides general specifications.
	**D2-HPP Setup**	provides as overview on general Handheld Programmer features and how to use them.
	**Entering Programs**	discusses all the operations used to enter a program.
	**Changing Programs**	shows you how to edit an existing program.
	**Naming and Storing Programs**	discusses using program names, password protection, and how to store programs on EEPROM memory chips.
	**System Monitoring and Troubleshooting**	provides an overview of the various features used to monitor and troubleshoot your PLC system.

Appendices Additional reference information is in the following two appendices:

	**DL105/DL205 Memory Map**	Appendix A provides a detailed listing of the DL105/DL205 memory map for I/O, timers, counters, etc.
	**Special Relays**	Appendix B lists the special relay contacts which are available to the ladder program to indicate system status, error conditions, instruction execution results, etc.

How can I use the Handheld?

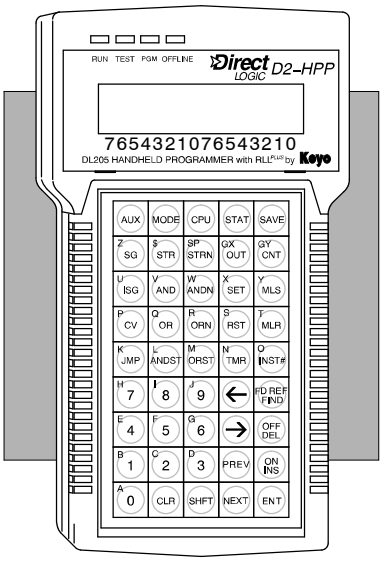
As a Programming Tool

The D2-HPP handheld programming unit is convenient for on-site setup, maintenance and minor PLC program changes. With the Handheld programer, you can change almost any system setting within the PLC. These settings include I/O configuration, retentive memory range selection, clock and calender setup, and many more.

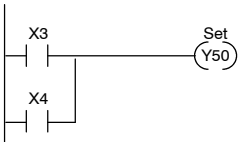
The Handheld programmer may be used to program the complete DL105 and DL205 PLC systems. The unit only allows programming the PLC with instruction mnemonics. Mnemonics are commands and operand data which will be processed by the CPU. Both on-line and off-line features will be described in detail within this manual.

The diagram to the right shows ladder logic which was programmed using the PC based ***Direct***SOFT™ programming software, and the equivalent mnemonics program using the Handheld programmer. Both methods of programming have advantages and can easily be used together or independently to support your PLC application. Once again, if you are creating a *large* program, it is recommended that you use ***Direct***SOFT™, which is better suited for the development environment.

Handheld Programmer



***Direct*SOFT RLL**



Handheld Mnemonics

STR X3
OR X4
SET Y50

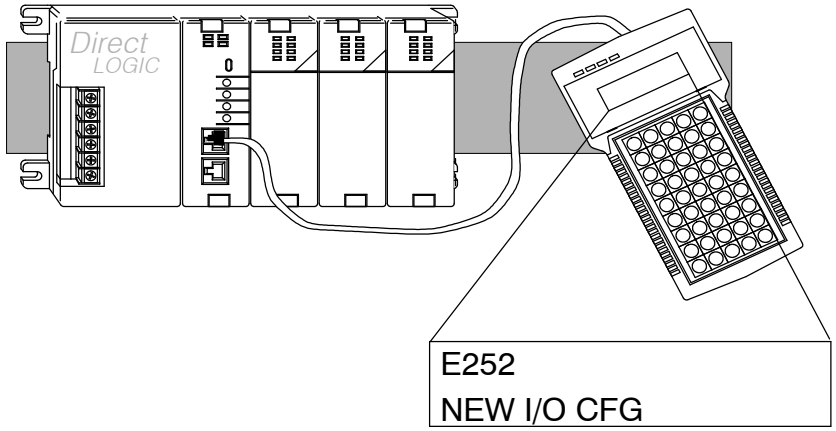
To Monitor Machine Operations

The Handheld programmer may be used to monitor memory status of the PLC system. The memory locations such as; V-memory, I/O information, timer/counter values, and system data may be selectively examined. The monitor status functions are performed in either Test/Run and Run modes. These monitoring modes help confirm all PLC conditions. Details on how to use the Handheld programmer to monitor your PLC system are described in later chapters.

As a Debugging Tool

If your PLC automation system appears to have a problem, you may use the Handheld programmer to quickly debug both hardware and software. Auxiliary functions, when executed, provide information to help diagnose PLC problems. Here are a few examples of commonly used diagnostics available.

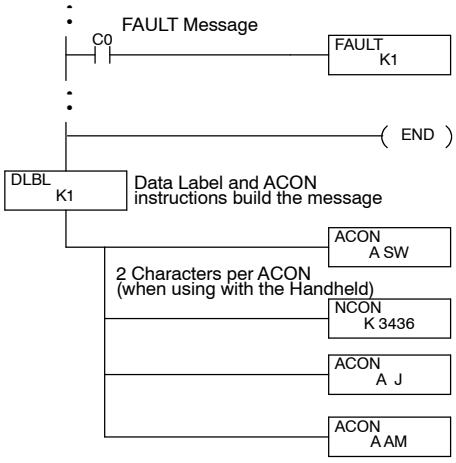
- **Program Diagnostics** — help locate instruction syntax errors, and potential duplicate output referencing.
- **I/O Diagnostics** — displays I/O errors and allows examination of special V-memory locations. This information may be viewed to help determine exact base and slot number having a problem.
- **Test Mode** — allows program logic to be verified without output status. While changing between Test-Program and Test-Run modes the digital output conditions are controlled.



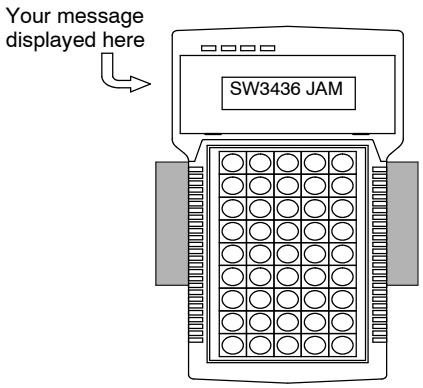
As a Low-Cost Message Log

The DL105 and DL205 CPU's allow embedded message instructions to be programmed in your control program. The Handheld programmer displays the messages saved within the CPU message log. If properly programmed, the fault messages are automatically displayed when the Handheld programmer is connected to the CPU. Please refer to the proper DL105 or DL205 User Manuals for examples on how to program these fault messages in your PLC system.

Program Initiates Message



Handheld Displays Message

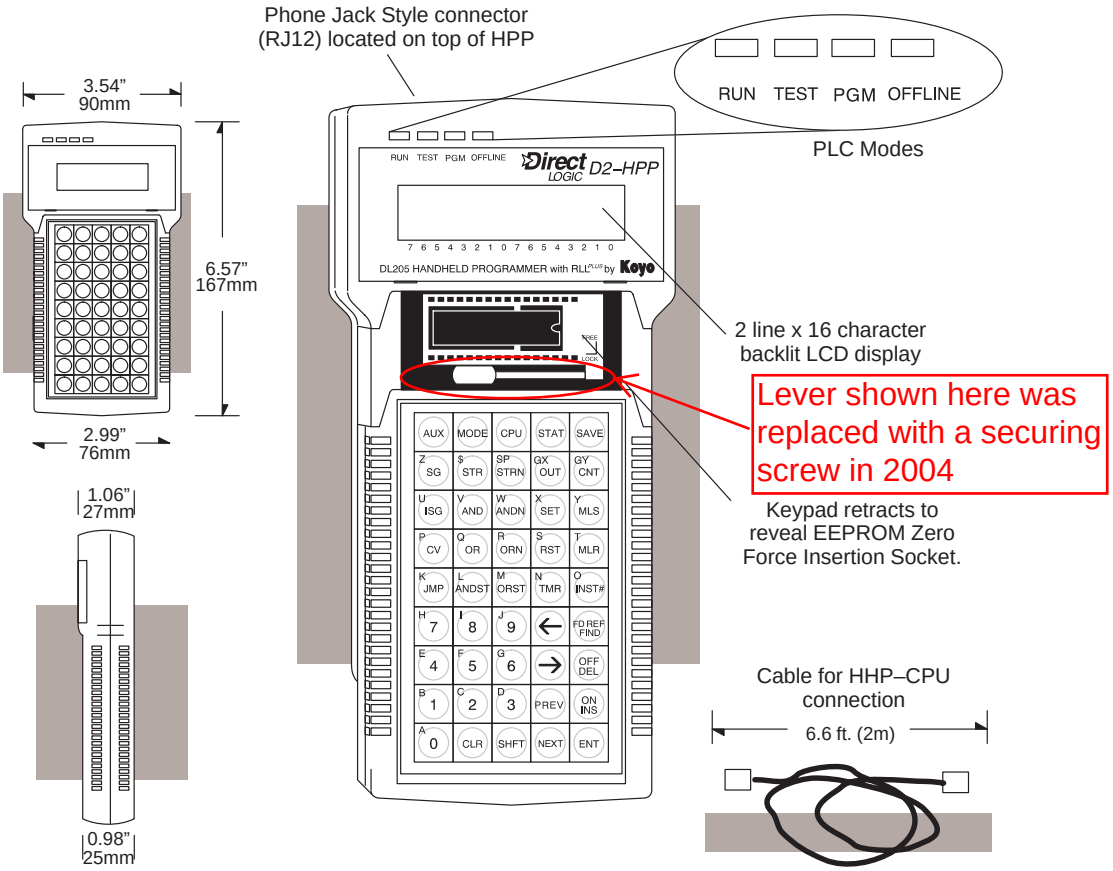


Physical Characteristics and Specifications

Handheld Programmer Layout

The Handheld programmer is designed for versatility. It provides features commonly *not* found on other handheld programmers. The figure below shows the basic physical characteristics of the Handheld programmer.

See Errata Sheet at the beginning of this file. In late 2004 the lever that secures the EEPROM was replaced by a securing screw.



The Handheld programmer has a two line, 16 character per line LCD display, which makes it easy to view the program, examine status and accesss other PLC data. The Handheld programmer contains a EEPROM socket which is located underneath the keypad. The EEPROM socket may be accessed by firmly holding the programmer and sliding the front keypad bezel down. The EEPROM programming feature may be used to:

- Store DL105 and DL205 CPU data to EEPROM non–volatile memory
- Compare the contents of a CPU to data stored on EEPROM
- Copy data from EEPROM to a CPU

Connection to the CPU

Correct cable is part number DV-1000CBL

The Handheld programmer is provided with a 6.6ft. (2m) programming cable (part number ~~D2–DSCBL~~). The cable is manufactured with RJ12 connectors at both ends. Connect the cable between the Handheld programmer and CPU programming port. When power is applied to the CPU, the Handheld programmer LED indicator(s) and LCD display should become active.

Specifications

The D2-HPP Handheld Programmer Specifications.

CPUs Supported DL130, DL230, DL240	**Programming Operations** Read, write, or erase programs Insert or delete an instruction Search and replace instructions Locate a specific address Read, write, or clear EEPROM Run time edit Password protection
Cables ~~D2-DSCBL~~ 6.6ft. (2m) Programmer Cable	**Machine Monitoring Operations** I/O status (up to 16 simultaneously) On / Off status for contacts, coils, control relays, and bit locations Timer and counter contacts, current values, and preset values Displays values in either HEX, BCD, Octal or ASCII
Message Display Up to 64, 23-character messages may be programmed (must be in RLL program). Maximum of 16 messages stored in each log (history and fault).	**Debugging Operations** Forcing (one scan only) Override forcing (multiple scans) Run, Program Mode, and Test Mode (DL240 only) Program syntax check Duplicate reference check Predefined error codes

Environmental

Operating Temperature	32 to 122 F° (0 to 50 C°)
Storage Temperature	-4 to 158 F° (-20 to 70 C°)
Humidity	30 to 95% (non-condensing)
Environmental Air	No corrosive gases
Vibration	MIL STD 810C 514.2
Shock Resistance	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304
Power	200 mA obtained through PLC port,
Dimensions	5.7" L x 4.6" H x 1.2" D 145mm W x 118mm H x 30mm D
Weight	1.7 oz. (48.2 g.)

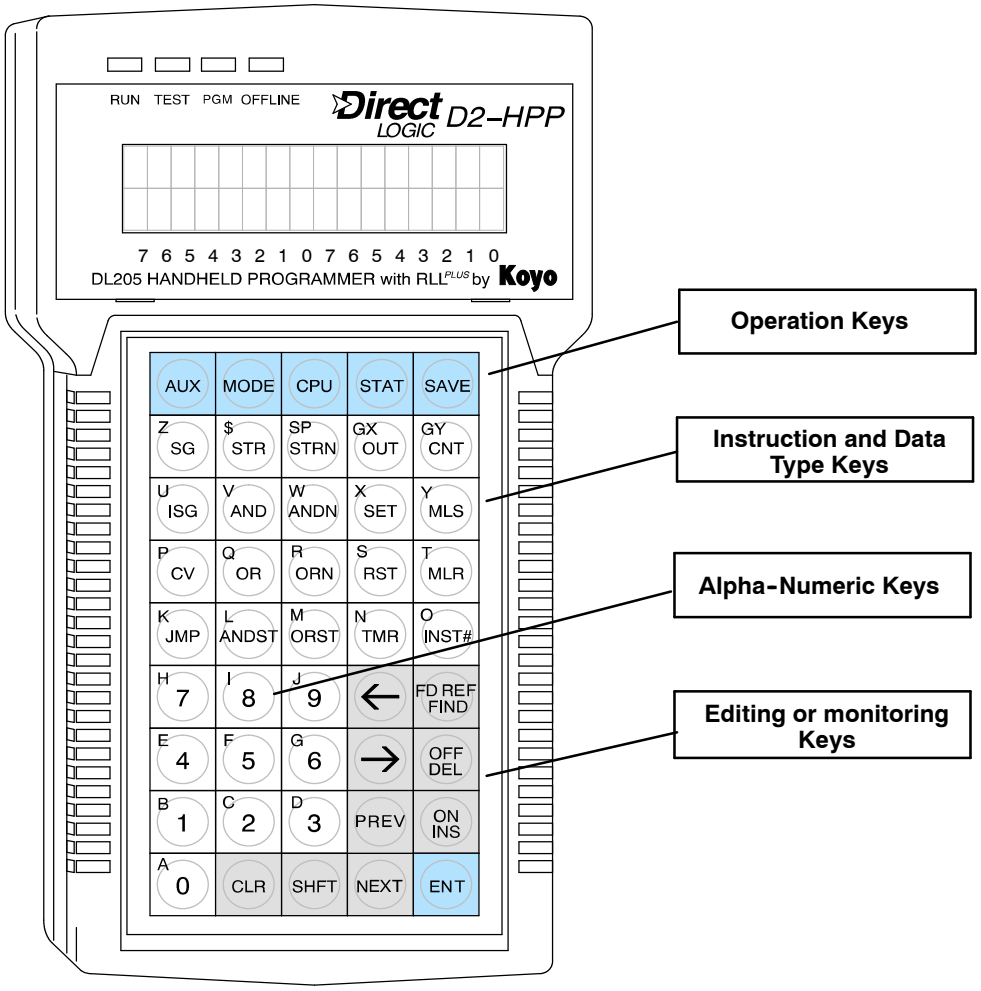
Correct cable is part number DV-1000CBL

Keypad Layout

Four Groups of Keys

The Handheld programmer keypad is organized into four key groups as defined below.

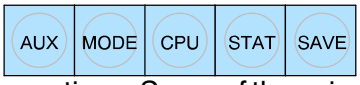
- **Operation keys** — used to call AUX functions, change programmer/CPU modes, monitor status and save program changes.
- **Instruction/Data type keys** — used to select the instruction and data type.
- **Numeric keys** — used to enter values in various formats (BCD, decimal, octal, HEX)
- **Editing/Monitoring keys** — used to move through the program (search, delete, etc.)



As you examine the keys, you'll notice some of the keys have more than one label. The top label describes the key when the **SHFT** (Shift) key is pressed. (These keys work just like the number keys on a computer keyboard.)

Operation Keys

These keys are used to select the following operations and perform various tasks with the Handheld programmer.



AUX key — is used to perform various types of operations. Some of these include program management, I/O Configuration/Diagnostics, CPU configuration, EEPROM operations, and password protection.

MODE key — is used to select the different modes available with your PLC (RUN, TEST, PGM and RUNTIME EDITS).

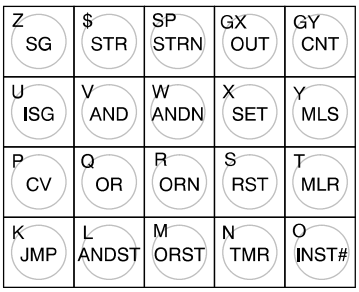
CPU key — is used to select the Handheld programmer programming mode. You may choose on-line or off-line communications to the PLC.

STAT key — is used to select status monitoring operations.

SAVE key — is used to store offline generated programs to the Handheld programmer's EEPROM.

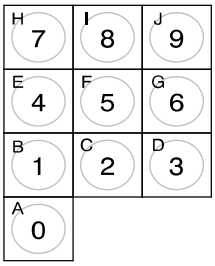
Instruction/Data Type Keys

These instruction keys allow you to select corresponding instructions when pressed. When closely examining the keypad, notice only some instructions have dedicated keys. All other instructions are entered by typing the instruction characters (mnemonics) using the secondary alphabet keys. The **INST#** key will allow for instruction numbers to be entered if selected.



Numeric Keys

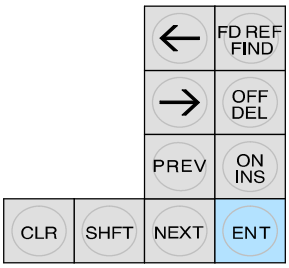
The numeric keys can be used to enter instruction identifiers and numbers for or constants. Some instructions require Hexadecimal numbers by pressing the **SHFT** key to access the alphabetic characters A — F.



Editing / Monitoring Keys

These keys are used to navigate, edit, create, and search through the PLC program and data.

The **PREV** and **NEXT** keys not only allow you to scroll through your program, they also provide scrolling list of valid mnemonics/data types while the cursor is positioned in the appropriate field location.



with the **INST#** key. While in status displays, **PREV** and **NEXT** can be used to show the status of adjacent memory locations.

The **CLR** key can be used to exit entry operations and clear the display. It may be necessary to press this key multiple times to clear the entire display.

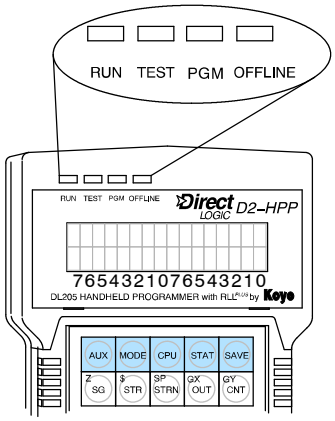
The **SHFT** key will allow use of the secondary property located in the top left corner of the keys. When the shift key is activated, the ^ character is displayed in the top right corner of the display screen.

Mode Indicators

The Mode LED's are located near the top of the Handheld programmer and indicate the CPU mode. The figure below shows all possible LED status, depending on the PLC mode selected. For additional information see the section titled " Changing the CPU mode" located in Chapter 2.

❑ = ON

Mode	RUN LED	TEST LED	PGM LED	OFFLINE LED
Run	❑			
Program			❑	
Test-Run (DL240 Only)	❑	❑		
Test-Pgm (DL240 Only)		❑	❑	
Handheld in Offline				❑
Runtime Edit	Flashing ❑			

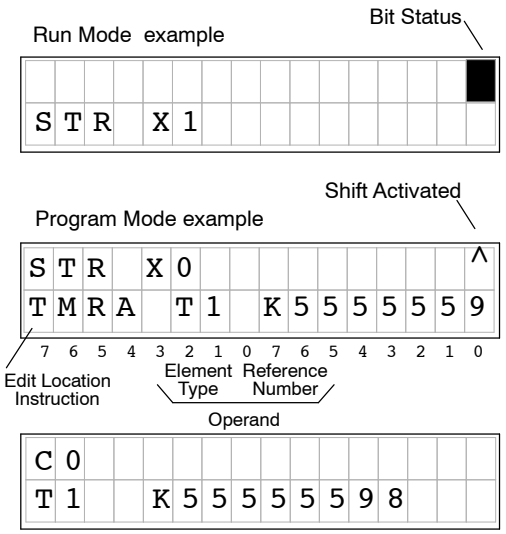


Display Panel

As mentioned, the Handheld programmer contains a two line, 16 character per line, LCD display screen. The user information and display format will change depending on the mode selected and the function being performed. The different mode display formats are discussed in later sections of this manual.

Viewing a Program

While in Run mode the Handheld programmer will display instruction and bit status. The example display on the right shows a Run Mode screen.
During the Program mode, the display screen allows viewing two instructions in your program as shown in the second example.
Some instruction, as with the Accumulative Timer (TMRA) will allow up to eight digits for a reference number. To view instructions or messages greater than 16 characters in length, press the right arrow key (→) to move viewing display. You may use the left arrow key ( ) to move the display to include viewing the instruction address.



Status Displays

If the Handheld programmer is placed in Run or Test-Run modes, different memory status options are available. The Status displays will indicate if instruction or bit status is ON or OFF. The display will contain the ■ symbol which indicates ON and the S character to indicate status is OFF. The first example demonstrates Bit status of input contact (X1) which is ON.

The **STAT** key will allow viewing status of a 16 bit range. The display for a range of bits are shown to the right. Note the underscore at the C2 position, which indicates the current cursor position. The cursor may be moved left and right, by pressing the corresponding arrow keys ( ). In this mode ■ indicates ON and □ indicates OFF.

The remaining example displays are Word Status for register addresses V2011 and V2010. The examples are showing the same registers in four different data formats.
The keystrokes used to switch between display formats are:

SHFT	K JMP	ENT	To select HEX
SHFT	O INST#	ENT	To select Octal
SHFT	A 0	ENT	To select ASCII
SHFT	D 3	ENT	To select Decimal

The last example display demonstrates status of Timers T20 and T17. The Timer and Counter status displays both maintain typical formats. Timer/Counter status bits are indicated with the box symbols. If the box shows solid (■), this indicates the timer/counter has attained the preset value.

Bit Status

```
                ■
STR X1
```

Bit Status for a Range of bits

```
   C  10     C_  0
□□□□□■□□□■□□□■□
```

Word Status - HEX

```
  V 2011   V 2010
    0041     0042
```

Word Status - Octal

```
  V 2011   V 2010
 0001010  0001020
```

Word Status - ASCII

```
  V 2011   V 2010
      AA       BA
```

Word Status - Decimal

```
  V 2011   V 2010
  00065D   00066D
```

Timer/Counter Status

```
  ■ T 20   □ T 17
    0120     0000
```

AUX Displays

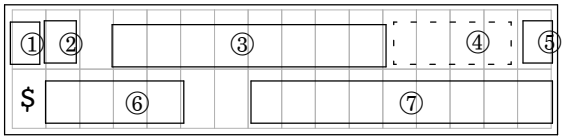
The Handheld programmer allows access to various Auxiliary functions by pressing the **AUX** key. All Auxiliary function have a unique display format. The example display shown to the right is the AUX 65 Diagnostic display.

Example Auxiliary Display

A	U	X		6	*		C	F	G		H	P	P		
A	U	X		6	5		R	U	N		D	I	A	G	

TEST-RUN Display (DL205 Only)

Some CPU's, such as, the DL240 support the Test-Run mode. With the Test-Run mode various groups of information are available. The different groups of information are labeled and described below. More details concerning Test-Run mode are provided in Chapter 6.



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

2

D2-HPP Setup

In This Chapter. . . .

- — Handheld Programmer Setup
- — CPU Setup
- — I/O Configuration
- — Auxiliary Functions

Handheld Programmer Setup

This section provides information on some basic Handheld programmer features and characteristics. Regardless of which DL105 or DL205 PLC system you are using, the following operations will apply.

Clearing the Display

To begin a new function, it may be necessary to clear the Handheld programmer entry buffer and display screen. Pressing the **CLR** (clear) key will clear the buffer and display. You must press the **CLR** key several times to prepare for new entries. The **CLR** key does *not* delete instructions or data.

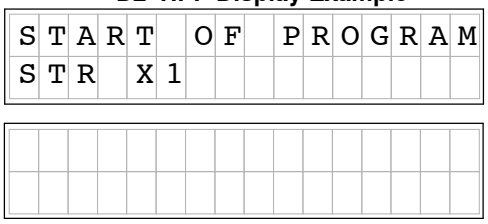
Press these keystrokes

1. To clear entry buffer and display screen

   CLR CLR ...

- Repeat pressing CLR until display screen is blank.

D2-HPP Display Example



Using the Cursor

The always flashing ■ symbol indicates the current cursor position. You can move the cursor position by using the left or right arrow keys (,). The arrow left key performs just like the backspace key on a PC keyboard, deleting the character position contents. The figure below is a example how the display changes by pressing the left arrow key.

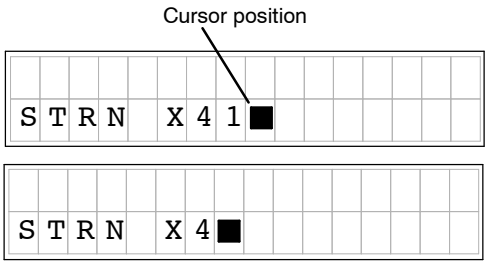
Press these keystrokes

1. To delete the previous character

   ←

2. To move cursor position right

   →



CPU Setup

A Few Things to Know

Below is a brief list of CPU operations discussed in this section.

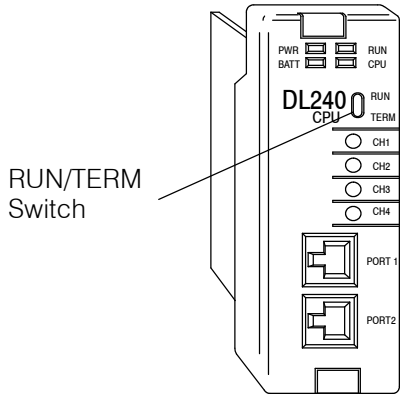
- Changing the CPU Modes
- Clearing the program (and other memory areas)
- How to initialize system memory
- Setting the CPU Network address
- Setting Retentive memory ranges
- Setting the Clock and Calendar

CPU Modes

With the Handheld Programmer connected to the CPU, you should examine the four mode LED's located near the top of the programming unit. The LED's will show the current mode status. Below is a definition for each of the Mode LED's. Test mode is not supported by all ***Direct***LOGIC™ PLC systems. Please refer to the appropriate DL105 or DL205 User Manual concerning the different CPU modes supported.

- **RUN** — executes the program and updates I/O modules.
- **PGM** — allows program entry, does not execute program or update I/O modules.
- **TEST** — allows CPU to maintain outputs, CRs, and Timer/Counter values when the CPU is changed from TEST-RUN to TEST-PGM mode. (See Chapter 6 for additional information.)
- **RUNTIME EDIT** — allows for program editing while the CPU is in RUN mode. These edits are *not* "bumpless." Instead, the CPU scan is momentarily interrupted (and the outputs are maintained in their current state) until the program change is complete.

NOTE: If your CPU has an external mode switch, it must be placed in the TERM position to change modes. This switch does *not* exist on the DL130 and DL230 CPU's.



Changing the CPU Mode

The Handheld programmer **MODE** key may be used to change the CPU mode. Pressing the MODE key will begin the process of changing modes. The keystrokes below will change the CPU mode from Run to Program.

Press these keys

1. To begin Mode Change

MODE

2. To select displayed mode

ENT

3. To accept mode change

ENT

- Use the NEXT/PREV keys to scroll available modes.

HPP Display Results

*	M	O	D	E		C	H	A	N	G	E	*				
G	O		T	O		P	G	M		M	O	D	E			

*	M	O	D	E		C	H	A	N	G	E	*				
P	G	M		M	O	D	E	?								

*	M	O	D	E		C	H	A	N	G	E	*				
C	P	U		P	G	M										

Selecting Different CPU Mode

You may use the **PREV** and **NEXT** key while performing a Mode Change, to choose a different mode. Always examine the Handheld programmer LED indicators to insure proper mode change, and desired CPU mode is selected.

WARNING: Only authorized personnel, familiar with all equipment concerning the PLC, should make mode and program changes. Changes during the RUN mode become effective immediately. Make sure to consider the impact of any mode change or program changes to minimize the risk of personal injury or equipment damage.

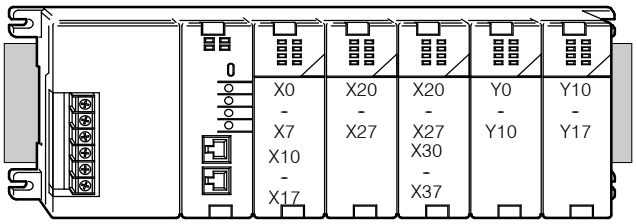
I/O Configuration

Automatic I/O Configuration (DL205 Only)

The DL205 PLC system's are designed to automatically examine installed I/O modules (including specialty modules) and establish the correct configuration and addressing when power is applied to the CPU.

The I/O addresses are assigned using octal numbering, meaning the I/O numbering always starts at zero and does not include 8 or 9. For example, a 16 point input module located in slot zero (the first slot next to the CPU) would be labeled X0-X7 for the first 8 points and X10-X17 for the second 8 points (never using the number 8 or 9) The addresses are assigned in groups of 8 or 16, depending on the number of points for the I/O module. Please refer to the DL 205 User Manual for details on automatic addressing. The following diagram shows a DL205 example I/O scheme.

- Slot 0 - 16pt Input
- Slot 1 - 8pt Input
- Slot 2 - Analog Input
- Slot 3 - 8pt Output
- Slot 4 - 8pt Relay Output



Checking I/O Configuration

The Handheld programmer may be used to view the current I/O configuration, by using the **AUX 41** function. While connected to your PLC, use the following example to display your I/O configuration.

Press these keystrokes	D2-HPP Display Results
1. Clear complete display screen CLR CLR CLR	
2. To display I/O configuration display E 4 B 1 AUX	AUX 4* I/O CFG AUX 41 SHOW CFG
3. To check I/O information ENT	AUX 41 I/O BASE D2-240: I/O BAS
4. Use arrow keys to display additional text → →	/O BASE 0/SLOTB I/O BASE
5. NEXT/PREV keys to view next and previous slots NEXT	/O BASE 0/SLOTP P/S
6. NEXT slot NEXT	/O BASE 0/SLOTC CPU V #.#
7. NEXT slot NEXT	/O BASE 0/SLOT0 8PT Input MDL
NEXT etc..	

Auxiliary Functions

What are Auxiliary Functions?

Handheld programmer keypad contains a key labeled **AUX**, which allows you to perform various Auxiliary Functions. Auxiliary Functions are divided into several different categories. Some AUX functions are for the Handheld programmer itself, and others for the PLC system. If an error occurs while performing a auxiliary function, the CPU may be in the wrong mode, or invalid data may have been entered.

Throughout this manual, step-by-step procedures for using Auxiliary functions are provided. Please refer to the DL105 or DL205 User Manual for details on AUX functions which may not be covered in this manual.

AUX Function and Description		DL130/ DL230	DL240
AUX 2* — RLL Operations			
21	Check Program	❍	❍
22	Change Reference	❍	❍
23	Clear Ladder Range	❍	❍
24	Clear All Ladders	❍	❍
AUX 3* — V-Memory Operations			
31	Clear V Memory	❍	❍
AUX 4* — I/O Configuration (DL205 CPU's Only)			
41	Show I/O Configuration	❍	❍
42	I/O Diagnostics	❍	❍
44	Power-up I/O Configuration Check	❍	❍
45	Select Configuration	❍	❍
AUX 5* — CPU Configuration			
51	Modify Program Name	❍	❍
52	Display / Change Calendar	❍	❍
53	Display Scan Time	❍	❍
54	Initialize Scratchpad	❍	❍
55	Set Watchdog Timer	❍	❍
56	Set CPU Network Address	X	❍
57	Set Retentive Ranges	X	❍
58	Test Operations	X	❍
59	Bit Override	X	❍
5B	Counter Interface Configuration	X	❍
5C	Display Error / Message History	X	❍

AUX Function and Description		DL130/ DL230	DL240
AUX 6* — Handheld Programmer Configuration			
61	Show Revision Numbers	❍	❍
62	Beeper On / Off	HP	HP
65	Run Self Diagnostics	HP	HP
AUX 7* — EEPROM Operations			
71	Copy CPU memory to HPP EEPROM	HP	HP
72	Write HPP EEPROM to CPU	HP	HP
73	Compare CPU to HPP EEPROM	HP	HP
74	Blank Check (HPP EEPROM)	HP	HP
75	Erase HPP EEPROM	HP	HP
76	Show EEPROM Type (CPU and HPP)	HP	HP
AUX 8* — Password Operations			
81	Modify Password	❍	❍
82	Unlock CPU	❍	❍
83	Lock CPU	❍	❍

❍ — supported

× — not supported

HP — Handheld Programmer function

Handheld Programmer Diagnostics

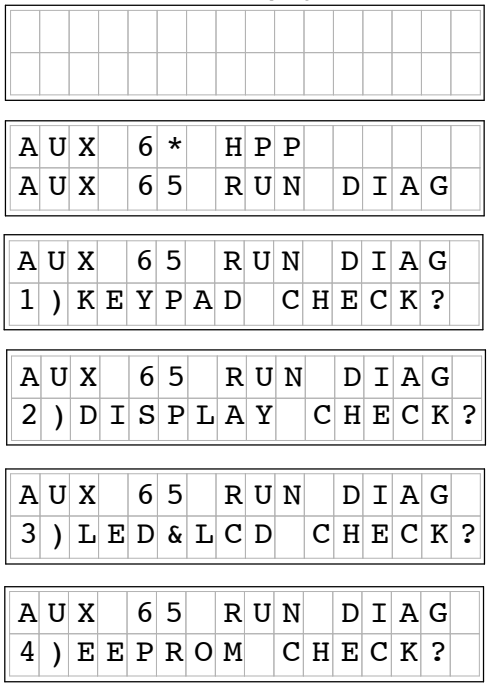
The Handheld programmer has built-in self checking diagnostics. You can select the HPP Diagnostics with the **AUX 65** function. When the diagnostic operation is finished, the main HPP diagnostic menu will be displayed. You may execute any of the Diagnostics by pressing the ENT key. The following example demonstrates using the D2-HPP Diagnotic operations.

Press these keystrokes

1. Clear complete display screen
   CLR CLR CLR
2. Select the Diagnostic operation
   G 6 F 5 AUX
3. Run Diagnostic operation
   ENT
4. To continue with next Diagnostic operation
   NEXT
5. To run Diagnostic operation
   ENT
6. This diagnostic will flash the LCD display and all the LEDs.
   NEXT
7. The EEPROM check will test the EEPROM installed in the handheld programmer.
   NEXT

- Press ENT to execute Diagnostic operation being displayed.
- Press CLR to exit the diagnostic operation being displayed.

D2-HPP display results



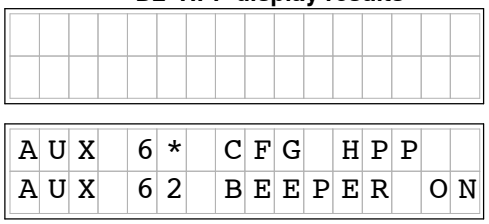
Beeper ON/OFF

The Handheld programmer contains a beeper which sounds to confirm the operator keystrokes. This beeper may be toggled ON and OFF with the **AUX 62** function.

Press these keystrokes

1. Clear complete display screen
   CLR CLR CLR
2. To toggle beeper ON/OFF
   G 6 C 2 AUX ENT

D2-HPP display results



Clearing an Existing Program

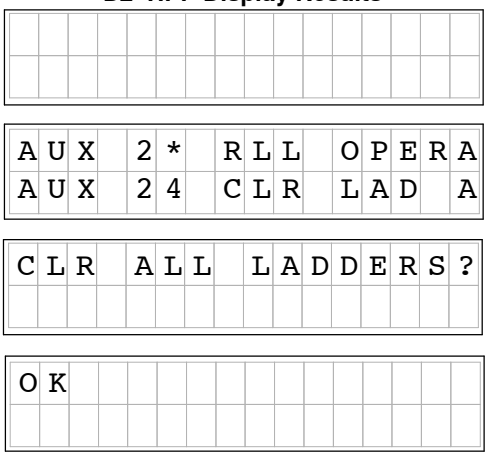
Important note, using this function will delete the PLC ladder program. With the CPU in Program mode, use the **AUX 24** function to clear the entire PLC application program.

Press these keystrokes

1. Clear complete display screen
   CLR CLR CLR
2. Select AUX 24 Clear Ladders display
   C 2 E 4 AUX
3. To select Clear Ladders operation
   ENT
4. To clear all ladders
   ENT

- The PLC must be in Program mode.

D2-HPP Display Results



To clear specified range of ladder program, or V memory, use the following Auxiliary functions.

- **AUX 23** — Clear Ladder Range
- **AUX 31** — Clear V Memory Range

Initializing System Memory

The CPU setup and configuration data are stored in memory which is called Scratchpad Memory. The Scratchpad memory may require initializing if major changes are introduced to your PLC system configuration or setup. For example, if you specify a range of Control Relays (CRs) as retentive, this setup data will be stored in scratchpad memory. Basic program changes or loading new programs do *not* always demand that the Initialize Scratchpad function be executed. If required you may default Scratchpad memory with the **AUX 54** function.

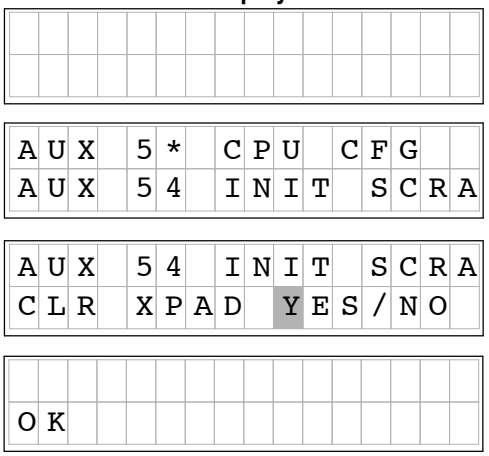
NOTE: This function may change PLC setup and configuration data in your system.

Press these keystrokes

1. Clear complete display screen
   CLR CLR CLR
2. Select AUX 54 Initialize Scratchpad
   F 5 E 4 AUX
3. To select YES/NO option
   →
4. To begin initialization
   ENT

- Select NO option to exit without initializing.

D2-HPP Display Results



Setting the CPU Network Address

Some CPU's, such as the DL240, contain a built-in ***Direct***NET™ port (PORT2). The Handheld programmer may be used to set the Port 2 network address and parameters. The default parameter settings are:

- Station address 1
- HEX mode (the handheld programmer will only support HEX mode)
- Odd parity
- 9600 baud rate

The ***Direct***NET™ User Manual provides additional information about network and communication parameter settings.

The following example demonstrates how to use the AUX 56 function.

Press these keystrokes	D2-HPP Display Results
1. Clear complete display CLR CLR CLR	
2. To select the diagnostic operation F 5 G 6 AUX	AUX 5* CPU CFG AUX 56 CPU N/W A
3. To change the network address ENT	AUX 56 CPU N/W A N/W # 01
4. Type new address number 1-90	AUX 56 CPU N/W A N/W # 01 02
5. To save the newly entered address ENT	AUX 56 CPU N/W A HEX / ASCII
6. To select communications mode use arrow keys to move cursor position → ←	AUX 56 CPU N/W A NONE / ODD
7. To save communications mode ENT	AUX 56 CPU N/W A 9600/19.2
8. To select communications parity → ←	AUX 56 CPU N/W A OK
9. To save communications parity ENT	
10. To select communications parity →	

- Press the CLR key to exit the AUX 56 function.
- Shaded box indicates cursor position.

Retentive Memory Ranges

The DL105 and DL205 CPU's all contain Retentive memory. Retentive memory is memory ranges which may store information in case of power loss. A super capacitor will maintain latest register values in case of short period CPU power loss or failure. If retentive memory ranges are important in your application, make sure to install a optional backup battery. Battery installation is covered in appropriate DL105 and DL205 User Manuals. Factory defaults for Retentive memory ranges are suitable for most applications. To change Retentive memory range, use **AUX 57** to select and set the desired range. The table below lists the Retentive memory factory defaults for the DL105 and DL205 CPU's.

Memory Area	DL130	
	Default Range	**Available Range**
Control Relays	C300 – C377	C0 – C377
V Memory	V2000 – V2377	V0 – V7777
Timers	None by default	T0 – T77
Counters	CT0 – CT77	CT0 – CT77
Stages	None by default	S0 – S377
Memory Area	**DL230**	
	Default Range	**Available Range**
Control Relays	C300 – C377	C0 – C377
V Memory	V2000 – V2377	V0 – V7777
Timers	None by default	T0 – T77
Counters	CT0 – CT77	CT0 – CT77
Stages	None by default	S0 – S377
Memory Area	**DL240**	
	Default Range	**Available Range**
Control Relays	C300 – C377	C0 – C377
V Memory	V2000 – V7777	V0 – V7777
Timers	None by default	T0 – T177
Counters	CT0 – CT177	CT0 – CT177
Stages	None by default	S0 – S777

Changing Retentive Memory Ranges

The **AUX 57** function may be used to change the Retentive memory ranges. When changing Retentive Memory ranges, keep in mind all memory ranges are defined with Octal addresses (8 bit boundaries, except for V-memory). For example, the Retentive memory factory default for CRs (control relays) are C300 thru C377, and could be reduced to C177 thru C300. You should always enter desired memory ranges in Octal numbers (e.g. xxx0-xxx7).The following figure demonstrates changing the Retentive memory range as described.

Press these keystrokes

1. Clear entire display screen
CLR CLR CLR

2. Select AUX 57 function
F 5 | H 7 | AUX

3. To change memory range
ENT

4. Enter new start address
B 1 | H 7 | H 7

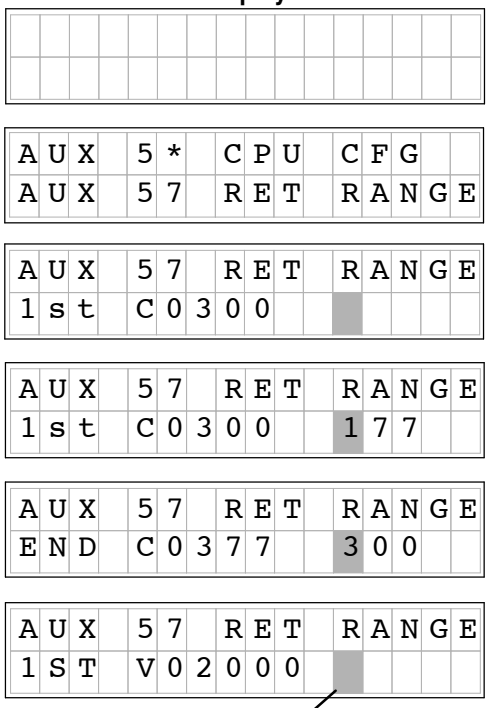
5. Accept entry
ENT

6. Enter new end address
D 3 | A 0 | A 0

7. Accept entry
ENT

- Press ENT to continue with other memory types.
- Press CLR to exit AUX 57 function.

D2-HPP Display Results



Setting the Clock and Calendar

The **AUX 52** function allows you to set the Real-time clock and calender. Not all ***Direct***Logic™ PLC's support the hardware clock and calender feature. For the CPU's which feature a clock and calendar the following format is used.

- Date — Year, Month, Date, Day of week (0 - 6, Sunday thru Saturday)
- Time — 24 hour format, Hours, Minutes, Seconds

If you change the date without updating the day of week (0-6), the CPU will *not* automatically correct any discrepancy between the date and the day of the week. For example, if you change the date to the 15th of the month and the 15th is on a Thursday, you will also have to change the day of the week (unless the CPU already shows the date as Thursday). Use the following example to change any component of the date or time settings.

NOTE: Verify the clock and calender is supported by your CPU, before attempting to use this Auxiliary function.

Press these keystrokes

1. Clear complete display screen
CLR CLR

2. Select AUX 57
F 5 C 2 AUX

3. Select date and clock display
ENT

4. Enter new date if required
→ → → → A 0 C 2

5. To accept press ENT twice
ENT ENT

6. Enter new time if required
B 1 C 2 D 3 A 0

7. To accept new entry press ENT twice
ENT ENT

- The shaded area indicates cursor position.
- Press the CLR key to exit date and clock function.

D2-HPP Display Results

```
                
                
```

```
AUX  5*  CPU  CFG
AUX  52  CALENDER
```

```
AUX  52  CALENDAR
96/01/01/6(SAT)
```

```
AUX  52  CALENDAR
96/01/02/7(SUN)
```

```
AUX  52  CALENDAR
TIME  00:06:00
```

```
AUX  52  CALENDAR
TIME  12:30:00
```

```
          96/01/02
          12:30:15
```

NOTE: If the CPU is without power for an extended period of time a battery is required to maintain the proper date and time.

3

Entering Programs

In This Chapter. . . .

— Entering Simple Ladder Programs
— Checking for Program Errors

Entering Ladder Programs

Purpose of Section

This section will demonstrate how to use the Handheld programmer for mnemonic programming. The D2-HPP is commonly used for program changes and creating simple RLL programs. Again, for larger more complex PLC applications, we recommend ***Direct***SOFT™ ,our PC based programming software. Basic knowledge of boolean logic and PLC programming is helpful to better understand the examples provided. For more programming examples, you should reference the appropriate DL105 or DL205 User Manuals for details on specific instructions.

Handheld Programmer Key Sequences

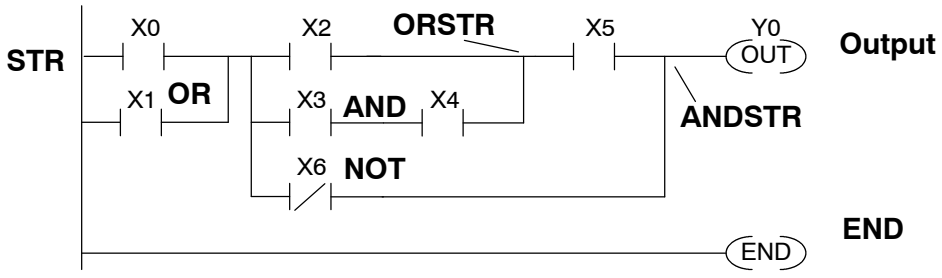
The Handheld programmer will buffers all keystrokes until the **ENT** (enter) key is pressed. The instruction syntax is checked for validity, when the enter key is pressed. If an instruction or data type is invalid an error message will be displayed. For a complete listing of error messages, please refer to Chapter 6.

Instruction Overview

The Handheld programmer only allows mnemonic instruction programming. A brief description of the most common used instructions are given below. The combination in which the mnemonics are entered will determine the Relay Ladder Logic (RLL) network structure and result.

- **STR** - Stores a normally open element and indicates the beginning of a rung or network.
- **AND** - Joins one element (such as a contact) in series with another element or group of elements.
- **AND STR** - Joins a group of elements in series with another group of elements. (not available with DL 105)
- **OR** - Joins one element in parallel with a previous element or group of elements.
- **OR STR** - Joins parallel branches (not available with DL 105)
- **Output** - Each rung must have at least one output (Y, C, or box instruction)
- **NOT** - used with other instructions to utilize normally closed elements.
- **END** - All programs must contain an END statement.

All networks must begin with the STR (store) or STRN (Store Not) instruction and are then combined with other instruction entries. Networks must conclude with at least one output instruction (Y coil, C coil, or Box instruction). Below is a ladder network showing how various mnemonics instructions are combined in a single network.



Navigating the Program

The Handheld programmer display screen, allows program instructions and their associated data to be viewed by the operator. All instructions are stored with a instruction addresses (*not* the same as rung addresses used in ***Direct***SOFT™). Newly entered instructions may be saved by pressing the **ENT** (enter) key.

Previous / Next Keys

Pressing the **NEXT** or **PREV** keys, allow scrolling through the mnemonic instructions in your program. It is not necessary to clear the display, before using these keys.

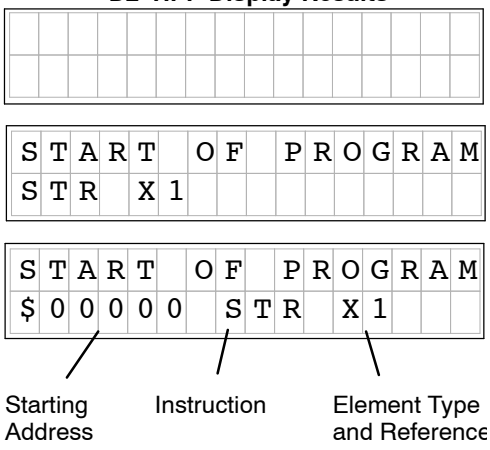
Starting at Address 0

When creating a new program, you should always begin the first program instruction at address zero ($00000). If you are in the Program mode, the 'START OF PROGRAM' message will appear, when positioned at the beginning address of your program. Use the left arrow **(←)** key to display the instruction addresses. To search the first address of your program, follow the example figure below.

Press these keystrokes

1. Clear entire display screen
   CLR CLR CLR
2. To Search first address ($00000)
   SHFT $ STR NEXT
3. To view instruction address
   ←

D2-HPP Display Results



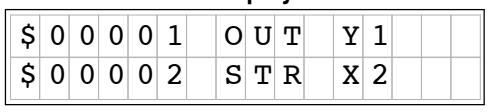
Searching for Addresses

You may search for and display instructions in your program by entering the specific addresses number. The following examples below demonstrate how to search and find different program items. The entire display screen must be cleared before performing the following examples.

Press these keystrokes

1. To Search specific address
   SHFT $ STR C 2 NEXT
   Enter desired address number to search

D2-HPP Display Results



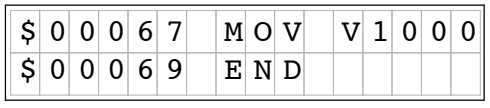
Searching for END Instruction

To search for the **END** command, follow the example below.

Press these keystrokes

1. To Search END instruction
   SHFT E 4 N TMR D 3 SHFT FD REF FIND

D2-HPP Display Results



Entering END Instruction

All programs require a **END** command. To enter the END instruction press the following keys.

Press these keystrokes

To program END instruction

1. SHFT E 4 N TMR D 3 ENT

Program Mode

The Program Mode is most commonly used to enter program instructions. After entering instructions, the changes are not executed until the CPU is placed in the Run mode. This will prevent unexpected machine operation which may be caused by changes which are performed.

With the Handheld programmer connected to the CPU, press the **MODE** key to select the Mode Change display. You may access the various modes by pressing the **NEXT** and **PREV** keys, while viewing the Mode Change display. To change to program mode follow the example below.

Press these keystrokes

1. To change modes.

MODE | NEXT | PREV | ENT | ENT

D2-HPP Display Results

Mode to Select

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

Entering a Simple Network

All programs begin starting at instruction address $00000. Use the **STR** (store) key to start programming your first network which contains a normally open contact (element) and output coil. The following will create a simple Store network.

Press these keystrokes

1. To enter Input contact.

$ STR | → | A 0 | ENT

2. Enter Ouput coil

GX OUT | → | B 1 | ENT

3. To type END instruction

SHFT | E 4 | N TMR | D 3 | ENT

D2-HPP Display Results

Begin Program entry here

```
START OF PROGRAM
STR X0
```

```
STR X0
NOP
```

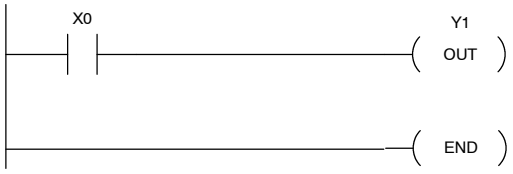
```
STR X0
OUT Y1
```

```
OUT Y1
NOP
```

```
OUT Y1
END
```

```
END
NOP
```

Equivalent Ladder Logic



Selecting Different Element Types

In the example above, you may press the **PREV** / **NEXT** keys, after the right ( ) arrow key, to scroll the different element types available. While displaying the desired element type enter the element address, then press **ENT** (enter).

Now that you have completed your first mnemonic instruction network, please continue through each of the following program examples. Append each of the remaining examples to the first network. To continue adding the examples begin each new networks at the last instruction programmed (END command).

NOTE: Always ensure the last instruction of your program is the **END** command. If the END command is missing, the Handheld programmer will not allow you to change modes, or run the program. Error #4 'No Program' may be displayed.

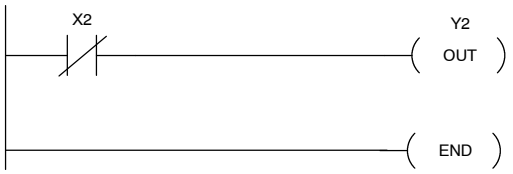
Entering Normally Closed Elements

To enter a network which contains a normally closed contact, begin with the **STRN** (Store Not) instruction. The following example demonstrates how to enter a network using the STRN instruction.

Press these keystrokes

1. To enter normally closed input
   SP STRN → C 2 ENT
2. Enter Ouput coil
   GX OUT → C 2 ENT
3. END instruction
   SHFT E 4 N TMR D 3 ENT

Equivalent Ladder Logic



HPP Display Results

Continue program entry here

OUT Y 1	
END	
OUT Y 1	
STRN X 2	
STRN X 2	
OUT Y 2	
OUT Y 2	
NOP	
END	
NOP	

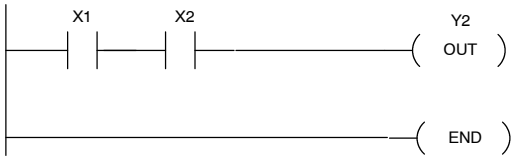
Entering Series Elements

Some networks require more than one element on a branch, this is referred to as contacts in series. To program elements in series, you begin the network as before using the store (STR,STRN) instruction. The **AND** instruction is used to join two elements in series. The following example demonstrates how to enter two series contacts and a single output coil.

Press these keystrokes

1. To enter first Input contact
   $ STR → B 1 ENT
2. To enter second Input contact
   $ STR → C 2 ENT
3. To enter Ouput coil
   GX OUT → C 2 ENT
4. END instruction
   SHFT E 4 N TMR D 3 ENT

Equivalent Ladder Logic



HPP Display Results

Continue program entry here

OUT Y 2	
END	
STR X 1	
NOP	
STR X 1	
STR X 2	
STR X 2	
NOP	
OUT Y 2	
NOP	
END	
NOP	

Entering Parallel Elements

To program a network with parallel elements (more than one branch per network), you will use the **OR** instruction. Once again, you begin the network as before using the store (STR,STRN) instruction for first element, then continue the **parallel** branch with the to create and second element data. You join the two parallel rungs using the coil OUT command. Follow the example below to create the most simple form of a parallel branch network.

Press these keystrokes

1. Enter first Input contact

$ STR → B 1 ENT

2. To start second branch and element

Q OR → C 2 ENT

3. To join parallel branch and enter Ouput coil

GX OUT → C 2 ENT

4. END instruction

SHFT E 4 N TMR D 3 ENT

D2-HPP Display Results

Continue program entry here

OUT	Y2
END	

STR	X1
NOP	

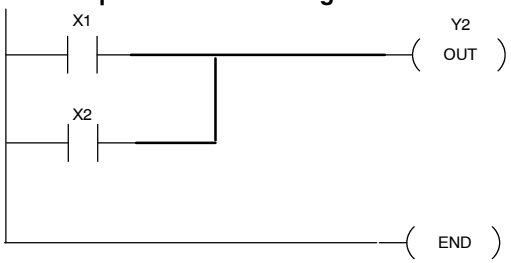
STR	X1	
OR	X2	

OR	X2	
OUT	Y2	

OUT	Y2
NOP	

END	
NOP	

Equivalent Ladder Logic



Later in this section, various examples using parallel element programming are provided. Branch programming examples require close observation of which order the mnemonic instructions are entered. If the instruction or data are not properly entered, the Handheld programmer display will response with a error message. Please take care and caution that the result of entering parallel logic does not present logical result problems.

Joining Series Elements in Parallel

Often it is necessary to program networks which contain parallel branches and series elements together to accomplish desired control. The **ORST (or store)** key allows you to program parallel branches with serial elements. The following example shows a simple network using the ORSTR instruction.

Press these keystrokes

1. To enter Input conact X0

$ STR → A 0 ENT

2. To enter second series conact

V AND → B 1 ENT

3. To begin parallel branch and contact X2

MSTR → C 2 ENT

4. To enter second parallel contact

V AND → D 3 ENT

5. To OR parallel branches

ORST → C 2 ENT

6. Ouput coil

GX OUT → C 2 ENT

7. END instruction

SHFT E 4 N TMR D 3 ENT

D2-HPP Display Results

Continue program entry here

OUT	Y2
END	

OUT	Y2
STR	X0

STR	X0
AND	X1

AND	X1
STR	X2

STR	X2
AND	X3

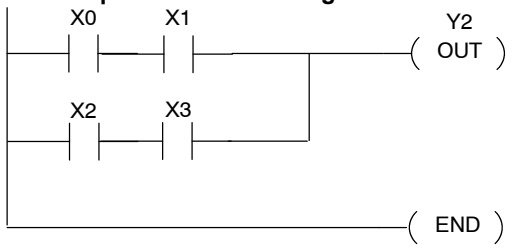
AND	X3
ORSTR	

ORSTR

OUT	Y2
END	

END	
NOP	

Equivalent Ladder Logic



Joining Parallel Branches in Series

The **ANDSTR** instruction joins one or more parallel branches which may be in series. The following example shows a simple network with parallel and series branches.

Press these keystrokes

1. Enter first Input conact

[$ STR] [→] [A 0] [ENT]

2. Enter second Input contact

[$ STR] [→] [B 1] [ENT]

3. Create branch and parallel contact

[Q OR] [→] [C 2] [ENT]

4. To join branch

[L ANDST] [ENT]

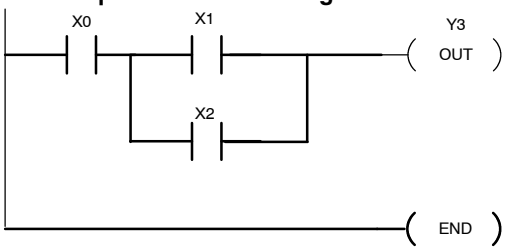
5. Enter Ouput coil

[GX OUT] [→] [D 3] [ENT]

6. END instruction

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

Equivalent Ladder Logic



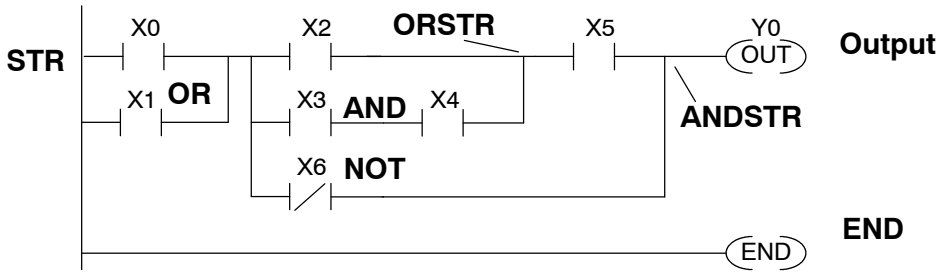
HPP Display Results

Continue program entry here

OUT Y2	
END	
STR X0	
NOP	
STR X0	
STR X1	
STR X1	
OR X2	
OR X2	
ANDSTR	
ANDSTR	
OUT Y3	
OUT Y3	
END	
END	
NOP	

Combination Networks

For combination networks, you may combine both the series elements and parallel branches. Combination logic allows you to solve almost any application problem. The following example is a ladder network, which is marked with **MNEMONIC** instructions and lists the order which the instructions may be entered.



Example Mnemonic Listing

ADDRESS	INSTRUCTION	DESCRIPTION
$00000	STR X0	Starts branch 1 with X0
$00001	OR X1	Joins X1 in parallel with X0
$00002	STR X2	Starts branch 2 with X2
$00003	STR X3	Starts branch 3 with X3
$00004	ANDN X4	Joins X4 (NOT) with X3
$00005	ORSTR	Joins branches 2 and 3
$00006	AND X5	Starts branch 4 with X5
$00007	ORN X6	Joins X6 (NOT) in parallel with X5
$00008	ANDSTR	Joins branches 4 and 5 with 1–3
$00009	OUT Y0	Stores the output and finishes the network
$00010	END	Ends the program

There are limits on how many boolean logic instructions can be combined in one network. The ***Direct***LOGIC™ CPU's use an 8 level stack to evaluate the various logic elements. The stack is a temporary storage area used to help evaluate the various logic combinations. Each time you enter a STR instruction, the instruction is placed on the top of the stack. All other instructions on the stack are pushed down one level. The And Store (ANDSTR) and Or Store (ORSTR) instruction combine levels of the stack when processed. Since the stack storage is eight levels, an error will occur if the CPU encounters a network that uses more than eight combined levels per network. For more details on the 8 level stack, please refer to section titled 'Programming Basics' in the DL105 or DL205 User Manuals.

Entering Timers and Counters

To enter a timer or counter, you also must prepare operand and enter preset values. This can be a constant value (K memory), or a V-memory location in the case of the DL240 CPU.

There are two methods of programming timers. You can have the timer with discrete timer control and status bits, or use comparative contacts, which enable at different time intervals during the control and status.

Timer Example Using Discrete Status Bits

The following timer example uses discrete status, with a preset of 3 seconds. If the timer is enabled for 3 seconds the status bit (T2) will turn ON. The timer will reset if X1 turns off, which in turn will resets the status bit (T2) off, and the accumulative value of the timer.

Press these keystrokes

HPP Display Results

Continue program entry here

1. To enter the first Input contact.

$ STR → B 1 ENT

OUT Y3
END

2. Enter the timer reference and preset value.

N TMR → C 2 → D 3 A 0 ENT

STR X1
TMR T2 K30

3. Begin new network and Timer status element.

$ STR → SHFT T MLR C 2 ENT

TMR T2 K30
TMR T2 V2300

4. To enter Output coil.

GX OUT → A 0 ENT

STR T2
OUT Y0

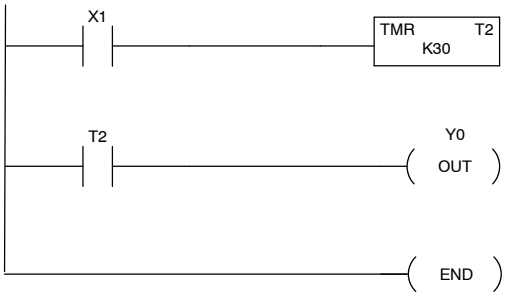
5. END instruction

SHFT E 4 N TMR D 3 ENT

OUT Y0
END

END
NOP

Equivalent Ladder Logic



Accumulating Timers & Counters

The Accumulating Timer which has additional lines connected to the timer instruction, can allow separate **start** and **reset** elements. All input element contacts are entered before the timer or counter instruction. The timer inputs may be of various types, e.g. timer status (T#), control relays (CR), etc. To scroll through the different operand data types, while programming the example below, press the **NEXT** key after the arrow → key is pressed. Although the Handheld programmer may allow you to select various data types, please refer to the DL105 or DL205 User Manual according to which CPU you are programming. For example, the DL240 will allow V-Memory registers for timer presets, where as the DL130 and DL230 will only allow K-Memory to be loaded as presets.

Entering Accumulating Timers (two Inputs)

This example demonstrates how to program a Accumulating Timer with a preset of 5 seconds. The timer discrete status bit (T0) contact will energize when the timer has timed for 5 seconds. The timer will reset when input X1 turns on, turning the timer discrete status bit off and resetting the timer current (timed) value to zero.

Press these keystrokes

1. To enter timer start Input conact

[$ STR] [→] [A 0] [ENT]

2. To enter the reset Input conact

[$ STR] [→] [B 1] [ENT]

3. Select Timer type and reference number

[N TMR] [SHFT] [A 0] [→] [A 0]

4. Enter Timer preset

[→] [F 5] [A 0] [ENT]

5. Begin new network with Timer status bit contact

[$ STR] [→] [SHFT] [T MLR] [A 0] [ENT]

6. Enter Output Coil Y0

[GX OUT] [→] [A 0] [ENT]

7. Enter END

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

HPP Display Results

Continue program entry here

```
OUT  Y0
END
```

```
OUT  Y0
STR  X0
```

```
STR  X0
STR  X1
```

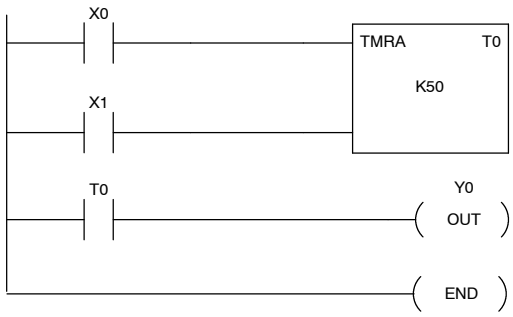
```
STR  X1
TMRA T0  K50
```

```
STR  T0
OUT  Y0
```

```
OUT  Y0
END
```

```
END
NOP
```

Equivalent Ladder Logic



Entering Relational Contacts

Relational contacts may be used to compare various types of information. For example, you may want to compare the current value of a timer with a constant value (K-Memory) or a V-memory register value. There are several types of compare operations that can be programmed, such as, less than, greater than, etc. See the DL105 or DL205 User Manual for more details on all relational contact instructions. The following example demonstrates how to program a greater than or equal to relational contact.

Press these keystrokes

1. Enter first compare reference

[$ STR] [→] [SHFT] [T MLR] [A 0]

2. Select constant reference to compare

[→] [B 1] [A 0] [A 0] [ENT]

3. Enter Ouput coil Y0

[GX OUT] [→] [A 0] [ENT]

4. Begin second Compare network and reference

[$ STR] [→] [SHFT] [T MLR] [A 0]

5. Enter compare V-Memory reference

[→] [SHFT] [V AND] [C 2] [A 0] [A 0] [A 0] [ENT]

6. Enter Output Coil Y1

[GX OUT] [→] [B 1] [ENT]

D2-HPP Display Results

Continue entry here

```
OUT  Y0
END
```

```
OUT  Y0
STR  T0  K100
```

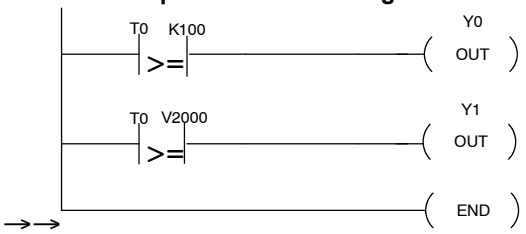
```
STR  T0  K100
OUT  Y0
```

```
OUT  Y0
STR  T0  V2000
```

```
STR  T0  V2000
```

```
STR  T0  K100
OUT  Y0
```

Equivalent Ladder Logic



Entering ASCII Characters

The DL105 and DL205 allow you to enter ASCII characters as part of the ACON instruction used for messages. An overview of the ACON instruction is provided in Chapter 6 of this manual. The example below shows the keystrokes used to enter the ASCII portion of the instruction with the Handheld programmer.

Press these keystrokes

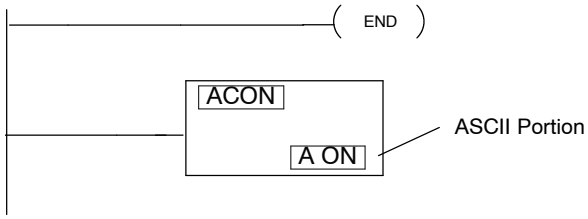
1. Type ACON instruction
   SHFT | A 0 | C 2 | O INST# | N TMR | →
2. Enter ASCII instruction
   SHFT | O INST# | N TMR
3. Enter instruction
   ENT

D2-HPP Display Results

```
END
ACON AON
```

```
ACON AON
NOP
```

Equivalent Ladder Logic



NOTE: More detailed information on the ACON instruction may be referenced in the DL105 and DL205 User Manuals.

Using the INST # key

Some mnemonic instructions may be entered by using a instruction number. The instruction number may also be referred to as *function* number. Use the Handheld programmer **INST#** key to begin the function number entry. If known, you may enter the specific instruction number , or scroll through available function numbers by pressing the **PREV**/**NEXT** keys. The following example demonstrates using the instruction number function.

Press these keystrokes

1. Enter function number
   O INST# | D 3 | G 6 | ENT
2. To scroll previous function number
   PREV

- Press ENT key to except function number
- Press the CLR key to exit

D2-HPP Display Results

```
F036  OUT
```

```
F035  OROUT
```

Entering Octal and Hex Numbers

For some instructions entries , special number formats are used for reference data. For example, the LDA (Load Address) instruction requires an octal number for the address reference. Also, you may want to load a hexadecimal value into the accumulator. The following example demonstrates how to enter octal and hexadecimal numbers using the Handheld programmer. For specific instruction information and optional number formats, please refer to the DL 105 and DL205 User Manuals.

Press these keystrokes

To enter LDA instruction

1. SHFT | L ANDST | D 3 | A 0 | →

Type Octal number

2. C 2 | A 0 | A 0 | A 0

Save entry

3. ENT

D2-HPP Display Results

```
LDA  O2000
```

```
LDA  O2000
```

Press these keystrokes

To enter LD instruction

1. SHFT | L ANDST | D 3 | → | PREV

Type Hexadecimal number

2. B 1 | C 2 | SHFT | F 5

Save entry

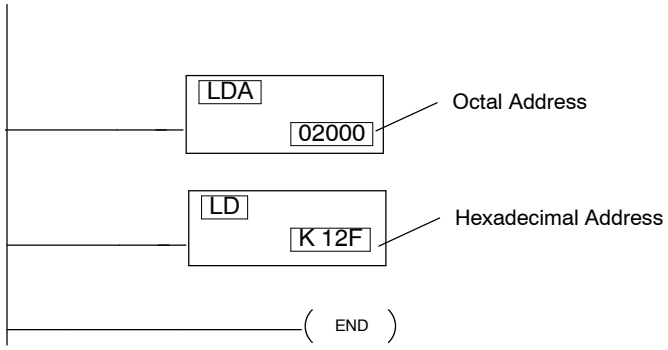
3. ENT

D2-HPP Display Results

```
LD   K12F
```

```
LD   K12F
NOP
```

Equivalent Ladder Logic



Checking for Program Errors

Error Checking

The Handheld programmer may also check your program for errors. You may choose two different types of program error checking.

- Syntax errors check
- Duplicate References check

Syntax Check

Use the **AUX 21** function, to select the 'CHECK PROGRAM' operation. Operation 1 performs a syntax check on the entered program logic. The following figure demonstrates how to access the Syntax check operation.

Press these keystrokes

1. Clear complete display screen
   CLR CLR CLR
2. To begin syntax check
   C 2 B 1 AUX ENT
3. Press ENT to select syntax check
   ENT

- This operation may take a few minutes, depending on the size of your program.
- When syntax check is complete one of two displays will appear.

D2-HPP Display Results

```
AUX 2* RLL OPERA
AUX 21 CHECK PRO
```

```
AUX 21 CHECK PRO
1:SYN 2:DUP REF
```

```
BUSY
```

```
$00029 E401
MISSING END
```

```
NO SYNTAX ERROR
?
```

Each error is labeled with an Error Code when displayed. Please refer to Chapter 6 for a complete listing of Error Code numbers. Upon receiving an error message, attempt correcting the problem and continue running the Syntax check until the message 'NO SYNTAX ERROR' appears.

Duplicate Reference Check

You may also use Check Program, Option 2, for multiple uses of the same output coil. The following example below demonstrates how to access AUX 21 and perform a Duplicate Reference check.

Press these keystrokes

1. Clear complete display screen

CLR CLR CLR

2. To begin syntax check

C 2 | B 1 | AUX | ENT

3. Position cursor on number 2 for DUP REF check

→ | ENT

- When Program Check is complete one of these two displays will appear.

D2-HPP Display Results

```
AUX  2*  RLL  OPERA
AUX  21  CHECK  PRO
```

```
AUX  21  CHECK  PRO
1:SYN  2:DUP  REF
```

```
BUSY
```

Error Display (example)

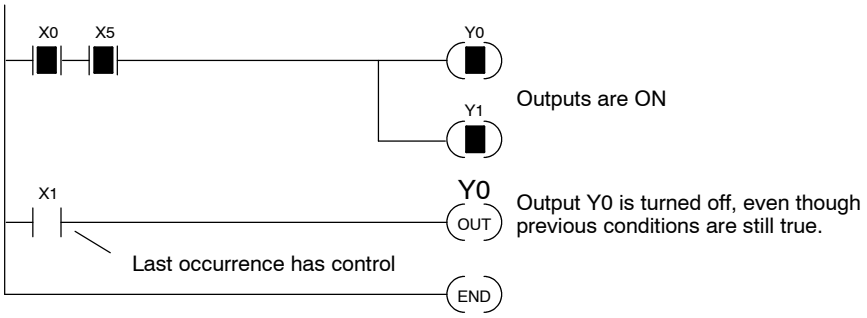
```
$00012  E471
DUP COIL  REF
```

No Duplicate Reference display

```
NO DUP REFS
?
```

If a Duplicate Reference error occurs, please refer to Chapter 6 for a complete listing of Error Code numbers. You should correct the problem and continue running the Duplicate Reference check until the message NO DUP REFS appears.

NOTE: You can use the same coil in more than one location. However, the last occurrence of the element will take priority. Consider the following example.



4

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

Editing Modes

To edit a program you may select either '**PROGRAM**' or '**RUN-TIME EDIT**' mode. The Program Mode is most commonly used for editing programs. The Run-Time Edit mode is helpful for very minor program changes or adjustments. The Handheld programmer will *not* allow changing from Program Mode, to Run Modes, if no program exists or program is missing the END command. This section begins with explaining the Program Mode and later discusses how to use the Run-Time Edit mode.

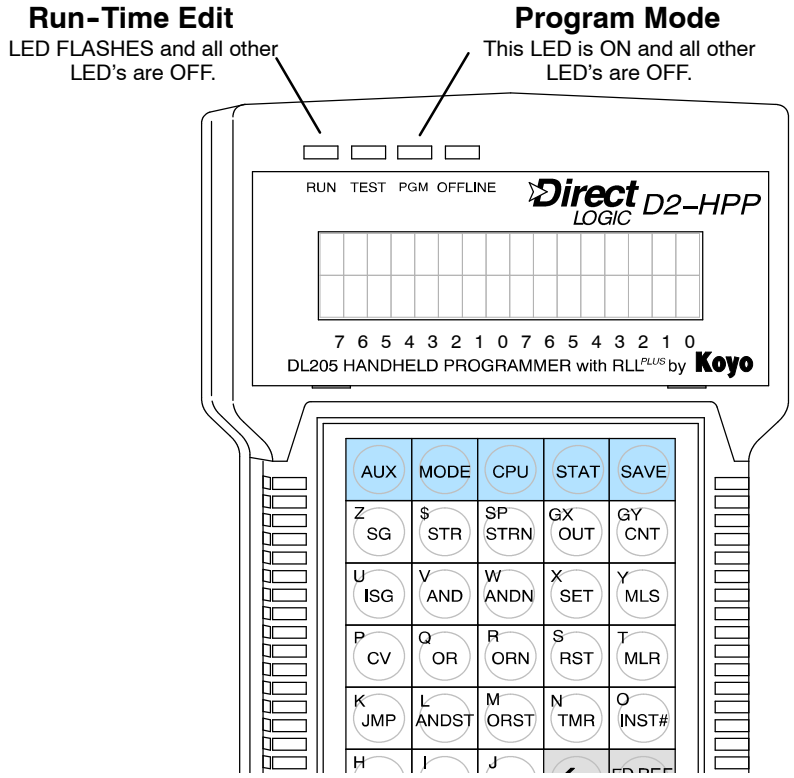
Program Mode

In the Program Mode, you can insert, edit, change, and delete mnemonic instructions. To enter a new network, you must carefully place the new instructions at the END or between the existing networks (after an OUT). During the Program edit mode the DL105 and DL205 CPU does not execute the application program, preventing unexpected machine control while editing the PLC program.

Run-Time Edit Mode (DL205 Only)

The DL240 CPU will allow you to edit programs during Run-Time Edits mode. While in the Run-Time Edits mode, most of the Handheld programmer functions operate the same as Program mode. 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. More details Run-Time Edit mode are discussed later in this chapter.

The figure below, shows the LED indicator status for the Program and Run-Time Edit modes.



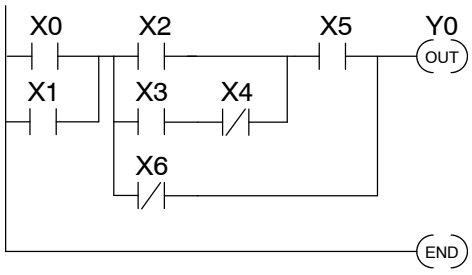
Displaying a Program

The Handheld programmer display screen allows viewing your program in the mnemonic instruction format. You may scroll through the individual instructions which are programmed using the **NEXT** / **PREV** keys. Depending on which mode you have selected, the display will maintain different screen formats. You may view the instruction address by pressing the left arrow key **(←)**. The different display modes and characteristics are discussed in Chapters 1 and 6.

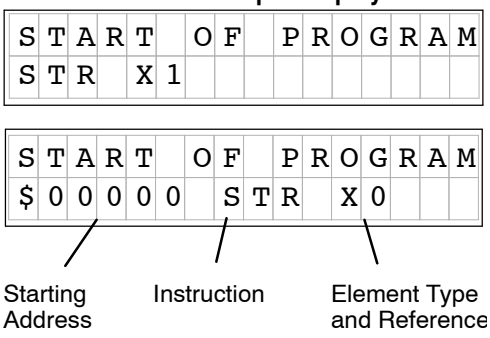
Combination Mnemonic Example

ADDRESS	INSTRUCTION	DESCRIPTION
$00000	STR X0	Starts branch 1 with X0
$00001	OR X1	Joins X1 in parallel with X0
$00002	STR X2	Starts branch 2 with X2
$00003	STR X3	Starts branch 3 with X3
$00004	ANDN X4	Joins X4 (NOT) with X3
$00005	ORSTR	Joins branches 2 and 3
$00006	AND X5	Starts branch 4 with X5
$00007	ORN X6	Joins X6 (NOT) in parallel with X5
$00008	ANDSTR	Joins branches 4 and 5 with 1-3
$00009	OUT Y0	Stores the output and finishes the network
$00010	END	Ends the program

Equivalent Ladder Logic



D2-HPP Example Display



Searching a Program Address

The Handheld programmer allows you to search and view your mnemonic instruction program. Once again, the display screen may have a different format depending on the mode selected. The figures below are display examples during the Run mode. The bit status of the instruction is indicated in the top right corner the display screen. If the ■ symbol appears the instructions bit status is true or (ON). If the S character appears the bit status is false or (OFF). To search the starting instruction or find a specific instruction address in your program, follow the examples below.

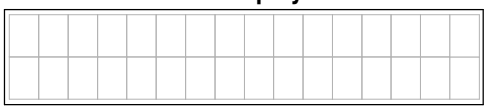
Searching Start of Program

Press these keystrokes

D2-HPP Display Results

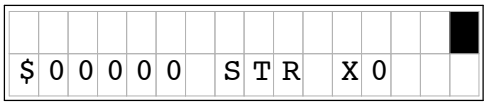
1. Clear complete display screen

CLR CLR CLR



2. To search start of program (address $00000)

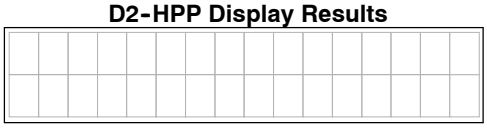
SHFT $ STR NEXT



3. To display instruction address

←

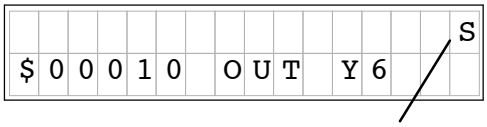
D2-HPP Display Results



Searching Specific Address

1. Clear complete display screen

CLR CLR CLR



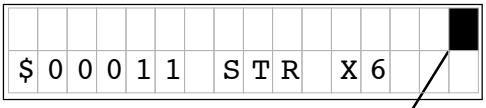
"S" indicates OFF
When in RUN or TEST-RUN mode

2. To search specific instruction address ($00010)

SHFT $ STR B 1 A 0 NEXT

3. To display NEXT instruction

NEXT



Solid fill indicates ON
When in RUN or TEST-RUN mode

Finding Instructions

The Handheld programmer may search instruction or reference numbers using the Find function. Use the **FIND** key to search an instruction or specific reference number. To select different instruction types press the **NEXT** / **PREV** keys during operand entry. After completing the entry press the **FIND** key to begin search. You may search any instruction type used within your PLC program. The following figure demonstrates how to search the instruction STR X1 using the Find function.

Finding Specific Reference

Press these keystrokes

1. Clear complete display screen

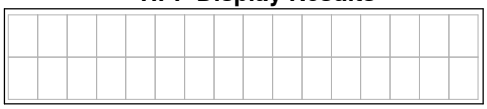
CLR CLR CLR

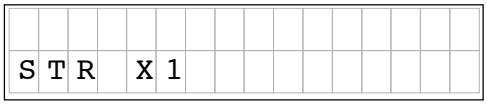
2. To find a instruction type and reference

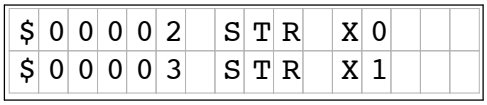
$ STR → D 3 FIND

* Press the NEXT/PREV key after the right arrow key to select different references.

HPP Display Results







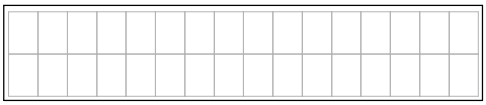
To Search Specific Reference

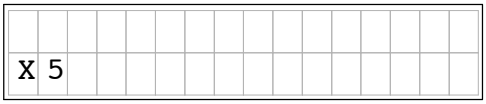
Clear complete display screen

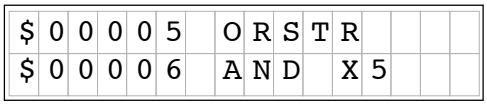
1. CLR CLR CLR

To find specific reference number

2. SHFT X SET F 5 SHFT FD REF FIND







If the memory reference or instruction is not found, one of the following error messages 'E604 REF MISSING' or 'E602INST MISSING' are displayed. (If you think the message is incorrect, re-enter your keystrokes and try the operation again.)

Changing an Instruction

Preparing Mode for Changes

The Handheld programmer allows you to change the Mnemonic instructions. If possible program changes should be performed in Program Mode. When switch from Run Mode to Program Mode the display screen will display your first instruction programmed. You should consider which mode the Handheld programmer is in, prior to attempting a search function. The Handheld programmer must be in one of the following modes to perform program changes.

- Program Mode
- Run-Time Edit Mode
- Test-Program Mode

The following figures and examples should be performed in the Program Mode. This example demonstrates how to find and change the X5 contact to X10.

Press these Keystrokes

HPP Display Results

To Find instruction

Clear complete display screen

1. CLR CLR CLR

```
SEARCHING
```

Enter the reference to search

2. SHFT | X SET | F 5 | SHFT | FD REF FIND

```
$00005 ORSTR
$00006 AND  X5
```

To Change the instruction

Cursor to reference number

3. → → → | B 1 | A 0

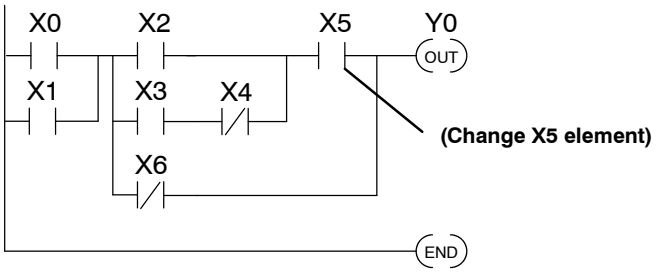
```
ORSTR
AND  X10
```

Enter new instruction and display next

4. ENT

```
AND  X10
ORN  X6
```

Equivalent Ladder Logic



Mnemonic Example Program

ADDRESS	INSTRUCTION
$00000	STR X0
$00001	OR X1
—	—
—	—
$00006	***AND X5***
—	—
—	—
$00010	END

Inserting an Instruction

You may insert mnemonic instructions by using the **INS** (insert) key. Pressing the **INS** key places the instruction *after* the instruction that is being displayed. You should first consider the desired location where to insert. The insert function duplicates the instruction displayed, and increments all remaining addresses. You should then enter a the desired instruction and reference. Use the following example to search and insert a new mnemonic instruction.

Press these Keystrokes

D2-HPP Display Results

To Find insert location

1. Locate NOT X6 contact

R AND → G 5 FD REF FIND

```
SEARCHING
```

2. Press INS key to begin insert operation

ON INS

```
$00006 ORSTR
$00007 AND  X5
```

3. Press ENT to confirm (or CLR to exit)

ENT

```
INSERT INST?
$00007 AND  X5
```

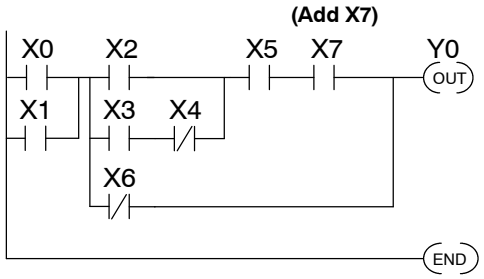
4. Enter NEW instruction and element

V AND → H 7 ENT

```
$00007 AND  X5
$00008 AND  X5
```

```
AND  X5
AND  X7
```

Equivalent Ladder Logic **Mnemonic Example Program**



$00000	STR X0
$00001	OR X1
—	—
$00006	AND X5
	AND X7
$00007	ORN X6
—	—
$00010	END

Location to search

Deleting an Instruction

Use the delete feature to remove an instruction from your program. The **DEL** key deletes the instruction that is currently being displayed. Note to make sure you are at the desired location within program prior to the Delete operation. Once you've deleted the instruction, the remaining addresses will automatically decrement. The following example demonstrates using the Delete function.

Press these Keystrokes

1. Locate X7 contact

[V AND] [→] [H 7] [FD REF FIND]

2. To Delete instruction

[OFF DEL]

3. Press ENT to confirm (or CLR to reject)

[ENT]

D2-HPP Display Results

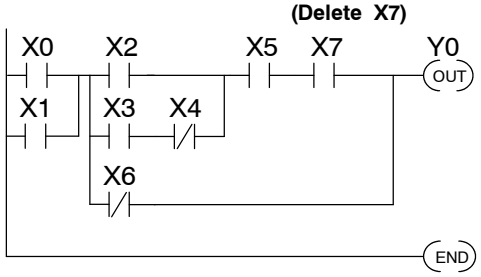
```
SEARCHING
```

```
$00006  AND  X10
$00007  AND  X7
```

```
DELETE  INST?
$00007  AND  X7
```

```
$00007  AND  X7
$00008  AND  X7
```

Equivalent Ladder Logic



Mnemonic Example Program

	$00000	STR X0
	$00001	OR X1
	—	—
Location to search	$00006	AND X5
		AND X7
	$00007	ORN X6
	—	—
	$00010	END

Using Search and Replace

The **AUX 22** function allows you to Search and Replace all occurrences of a specific instruction. The example below demonstrates replacing every instance of X5 with X10.

Press these Keystrokes

D2-HPP Display Results

1. Clear complete display screen

CLR CLR CLR

2. Use AUX 22 to change memory references

C 2 C 2 AUX ENT

AUX 2* RLL OPERA
AUX 22 CHNGE REF

3. Enter beginning address for the change or press ENT to accept the default

ENT

AUX 22 CHNGE REF
1st $00000

4. Enter ending address for the change or press ENT to accept the default

B 1 G 6 ENT

AUX 22 CHNGE REF
END $16

5. Enter old memory reference

F 5 ENT

AUX 22 CHNGE REF
OLD X5

6. Enter new memory reference

B 1 A 0 ENT

AUX 22 CHNGE REF
TO X10

7. Press ENT to confirm change

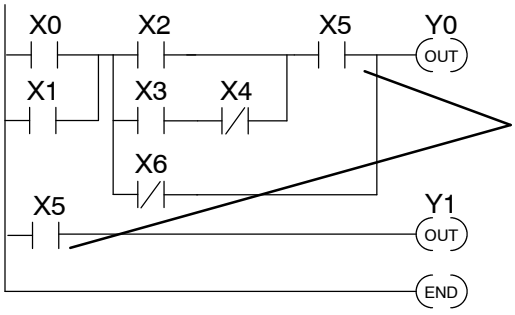
ENT ENT

AUX 22 CHNGE REF
X0005->X0010

- Use CLR key to exit AUX 22 function

REFERENCE CHANGE
OLD X

Equivalent Ladder Logic



Mnemonic Example Program

$00000		STR X0
$00006		AND X10
$00010		STR X10

Editing Programs During Run Mode

The DL205 CPU's allow you to edit programs during Run mode. To modify a program in RUN mode use **MODE** key to select "RUN-TIME EDITS".

The operations you are able to perform in Program mode also apply in the Run-Time mode. 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-Time 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 change to Program Mode.

WARNING: Only authorized personnel fully familiar with all aspects of the PLC 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.

There are some important operation sequence changes during Run Time Edits.
1. If there is a syntax error in the new instruction, the CPU *will not* enter the Run Mode.
2. If you delete an output coil reference and the output was on at the time, 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.

Selecting Runtime Edits

The figure below demonstrates how to select the Runtime edit feature. Note the following are some conditions which pertain during Run-Time Edit mode.

- Once you select RUNTIME EDITS the RUN LED starts blinking. This indicates a edit may be performed while in the Run mode.
- If you had displayed an address just before selecting Run-Time Edit mode, you may immediately jump to the same address by pressing the **NEXT** key. This allows you to search for an address or instruction *before* you change the mode.

Press these Keystrokes	D2-HPP Display Results
1. While in the Run Mode select RUNTIME EDIT mode MODE NEXT ENT	*MODE CHANGE* GO TO PGM MODE
2. Confirm mode change ENT	*MODE CHANGE* RUNTIME EDITS?
3. Confirm mode change CLR NEXT	*MODE CHANGE* RUNTIME EDITS

Changing Constant Values During Run Mode

Changing the value of a constant in an instruction during Run Mode works the same as it does during Program Mode. The following example is not reflected in the short program we have been working with, since it does not contain any instructions with a constant value. This example shows how to modify the preset value of an Up Down Counter.

Press these Keystrokes	D2-HPP Display Results
1. Locate and display the instruction with the constant you want to change	S UDC CT16 K50
2. Position cursor on constant value and enter new value → → → → G 6 A 0 ENT	END

- The change in value is not saved until you press the ENT key.
- After pressing the ENT key, the next instruction programmed will automatically be displayed.

5

Naming and Storing Programs

In This Chapter. . . .

— Program Names and Passwords
— Saving Programs on EEPROM

Program Names and Passwords

Program Names

Both the DL105 and DL205 PLC's allow you to name your application programs. This feature is helpful to store your program in the Handheld programmer EEPROM memory. The program name can be up to eight characters in length and allows all alphanumeric characters (A-Z, 0-9) for valid entry.

Press these Keystrokes

D2-HPP Display Results

1. To call AUX 51 function

F 5 | B 1 | AUX

```
AUX  5*  CPU  CFG
AUX  51  MODIFY  PG
```

2. Press ENT to get the Modify Program display

ENT

```
AUX  51  MODIFY  PG
oooooooo
```

3. Enter program name

SHFT | P CV | R ORN | E 4 | S RST | S RST

SHFT | B 1

```
AUX  51  MODIFY  PG
PRESS1
```

4. Press ENT to accept the name

ENT

```
PROGRAM
PRESS1oo
```

- Press CLR to exit the Modify Program display
- You may also position cursor with arrow keys to change the name and then press ENT

Password Protection

The DL105 and DL205 PLC's provide an extra measure of protection by allowing Password protection. You may enter a password that prevents unauthorized personnel from performing program operations. A password must consist of eight digits. The first digit of the password (most left position), may be an alphanumeric number (A-F, 0-9) and the remaining seven digits may be numeric characters (0-9). To remove a entered password, enter all zeros (00000000), which defaults the CPU having no password protection. (This is the default from the factory.)

Press these Keystrokes

D2-HPP Display Results

1. Use AUX 81 to name the CPU program

I 8 | B 1 | AUX

```
AUX  8*  PASSWORD
AUX  81  MODIFY  PA
```

2. Press ENT to get the Password display

ENT

```
PASSWORD
00000000
```

3. Enter password

B 1 | C 2 | D 3 | E 4 | F 5 | G 6

H 7 | I 8

```
PASSWORD
12345678
```

4. Press ENT to accept the password or use the arrow keys to change it.

ENT

This position will accept (A-F, 0-9 for valid entry.

```
PROGRAM
PRESS1
```

- Press CLR to exit from Password display

The password is stored in the program memory. If you install the program or EEPROM in another CPU or Handheld, the password protection remains in effect.

Locking the CPU with Password Protection

Once you've entered a password, you may use the **AUX 83** to lock the CPU against program access. This function will prevent users from changing CPU setups and modifying the PLC program. There are two ways to lock the CPU.

- The CPU is always locked after a power cycle (if a password is present).
- You can use AUX 83 and AUX 82 to lock and unlock the CPU.

WARNING: Make *sure* you remember the password *before* you lock the CPU. Once the CPU is locked you cannot view, change, or erase the password. You also cannot erase the EEPROM and start over.

The following example uses the AUX 83 function, to lock a CPU with password protection.

Press these Keystrokes

1. Call AUX 83 to Lock the CPU password

I 8 | D 3 | AUX

2. To select Lock CPU display

ENT

3. Confirm Lock operation

ENT

- Press ENT to accept the password or use the arrow keys to change it.
- Press CLR to exit CPU Lock operation

D2-HPP Display Results

```
AUX 8* PASSWORD
AUX 83 LOCK CPU
```

```
CPU LOCK
LOCK?
```

```
CPU LOCKED
```

The message 'CPU UNLOCKED' appears if you attempt to lock a CPU that does not have a password.

Unlocking the CPU Password Protection

Use the **AUX 82** function to unlock CPU's which have been enabled with the Lock protection. The Unlock function works similar to the Lock function, but will require you to enter the password which has been programmed. The following example demonstrates how to unlock a password protected CPU.

Press these Keystrokes

1. Use AUX 83 to Unlock the CPU password

I 8 | C 2 | AUX

2. To select Unlock CPU display

ENT

3. Enter password to authorize Unlock

B 1 | C 2 | D 3 | E 4 | F 5 | G 6 | H 7 | I 8

4. Confirm Unlock operation

ENT

- Press CLR to exit CPU Unlock operation

D2-HPP Display Results

```
AUX 8* PASSWORD
AUX 82 UNLOCK CP
```

```
CPU LOCKED
PASSWD:
```

```
CPU LOCKED
PASSWD: 12345678
```

```
CPU UNLOCKED
```

The error message 'E504BAD REF/VAL' appears if you enter an incorrect password. If you press **CLR** you can attempt to enter the password again.

Saving Programs to EEPROM

As you develop your program with the Handheld programmer, pressing the ENT key saves the entry to the PLC CPU memory. The DL105 and DL205 use different types of CPU memory.
The DL205 series PLC's use a EEPROM IC chip for program and data storage. You may use the Handheld programmer AUX71 function to program this EEPROM. The DL105 Micro PLC's use a non-volatile Flash ROM memory for program information storage. The DL105 *does not* require any EEPROM handling, therefore the following EEPROM functions are *not* normally used.

Before you attempt to save your program to EEPROM, you must first install a EEPROM inside the Handheld programmer. The following table indicates which EEPROM to use in the different DL205 CPU's.

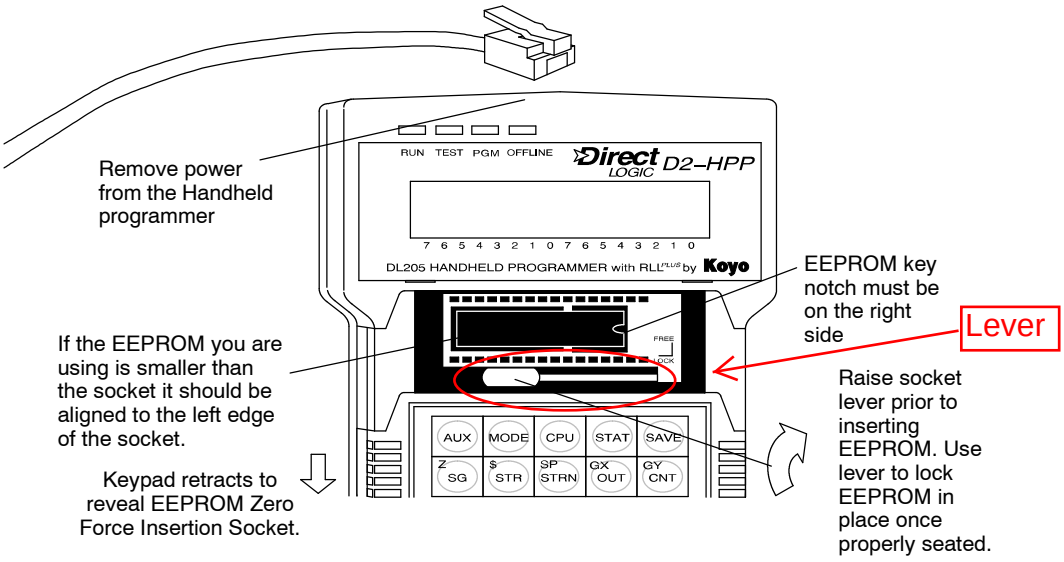
Types of EEPROMs (DL205 ONLY)

The DL230 CPU uses a 2K EEPROM and the DL240 CPU uses a 3K EEPROM for program storage. Either size of EEPROM may be used in the Handheld programmer for offline programming. You may electrically erase already programed EEPROM's as explained later in this chapter.

CPU type	EEPROM Part Number	Capacity
DL230	Hitachi HN58C65P-25	2K words
DL240	Hitachi HN58C256P-20	3K words

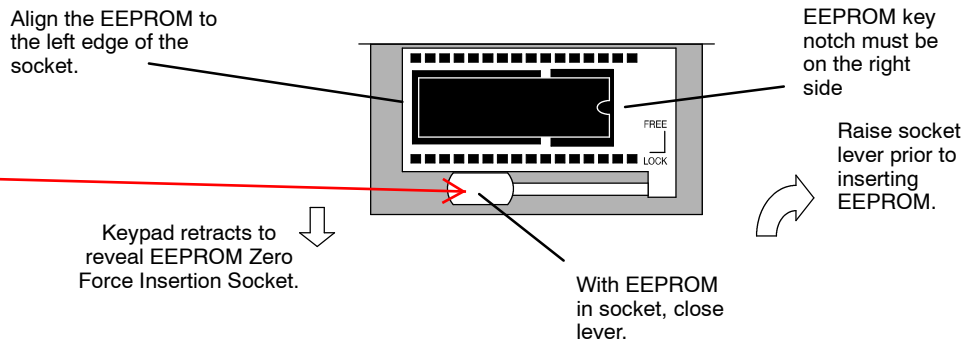
NOTE: Small programs using common instructions supported by *both* CPUs is possible, as long as the program size is within the DL230 capacity (under 2K). These programs may be used between both of the CPU models. However, the EEPROM installed in the Handheld Programmer *must* be the same size (or larger) than the CPU being used. For example, you could not install a DL240 EEPROM in the Handheld Programmer and download the program to a DL230, unless the program size limits are that of a DL230 capacity.

See Errata Sheet at the beginning of this file. In late 2004 this lever that secures the EEPROM was replaced by a securing screw.



Inserting a EEPROM in the Handheld Programmer

The Handheld programmer should *not* be powered during EEPROM installation. EEPROM chips can be damaged if not properly handled and/or proper electrical grounding precautions used. While installing a EEPROM, ensure not to bend any of the electrical pins. Align the EEPROM with the left side of the pin socket and the key notch to the right.



See Errata Sheet at the beginning of this file. In late 2004 this lever that secures the EEPROM was replaced by a securing screw.

1. **Disconnect power** from the Handheld programmer.
2. Slide keypad door down. The keypad door only slides partially open. Do not force!
3. Lift socket lever, to clear socket pin openings.
4. Insert EEPROM in socket. The key notch must be on the right and the EEPROM must be aligned with the left side of the socket.
5. Once EEPROM is inserted, press socket lever down.
6. Slide keypad closed.
7. Reconnect power to the Handheld programmer.

WARNING: EEPROMs can be damaged by static electricity, therefore; you should take precautions to ground yourself before handling the EEPROM. All work performed should be made on a conductive and grounded surface.

Using EEPROM functions with the DL105

The DL105 Micro PLC's use Flash ROM memory for program and system information storage. The Handheld programmer may still be used for storing and uploading DL105 programs. The DL105 may require the Initialize Scratchpad operation to be performed, before changing to Run mode, after (HPP->CPU) EEPROM program has been loaded.

Checking the EEPROM Type

The **AUX 76** function may be used to check the EEPROM size installed in the DL205 CPU or the D2-HPP programmer. The display will indicate both the CPU EEPROM size and the Handheld programmer EEPROM size if installed. If the EEPROM is not installed in the Handheld programer, then dashes (-) will be displayed below the HPP header.

Press these Keystrokes

1. Use AUX 76 to Check EEPROM Type

H 7 | G 6 | AUX

2. To select EEPROM checking

AUX

3. If HPP has a EEPROM installed, use the arrow right key to scroll the display.

→

- Press CLR key to exit EEPROM check function

D2-HPP Display Results

```
AUX  7*  EEPROM
AUX  76  SHOW TYPE
```

```
   CPU        HPP
EEPROM  03K  EEPR
```

```
      HPP  (D2-240)
03K  EEPROM   03K
```

Checking for a Blank EEPROM

Before copying your program to a EEPROM make sure the EEPROM does not contain any information which will be overwritten. You can check for a blank EEPROM by using AUX function 74, BLANK CHK.

Press these Keystrokes

1. Use AUX 74 to Check for a blank EEPROM

H 7 | E 4 | AUX

2. To select EEPROM blank check

ENT

3. To execute EEPROM blank check

ENT

4. The Handheld programer will respond with one of these three display messages.

- Press CLR to exit EEPROM blank check

D2-HPP Display Results

```
AUX  7*  EEPROM
AUX  74  BLANK CHK
```

```
AUX  74  BLANK CHK
EEPROM BLANK CHK
```

```
EEPROM IS BLANK
```

```
E621
EEPROM NOT BLANK
```

```
E622
NO HPP EEPROM
```

NOTE: If you copy data to an EEPROM which has existing data stored on it, the new data could overwrite portions of the existing data and leave other portions as they previously existed resulting in a unreliable copy of your data. It is always recommended to clear non-blank memory cartridges prior to copying data to ensure you get a "clean" copy of your new data.

Erasing a EEPROM

The **AUX 75** function will allow you to erase a EEPROM. Use the following example to erase (clear) a EEPROM which is installed in the Handheld programmer.

Press these Keystrokes

1. Use AUX 75 to Erase EEPROM

H 7 | F 5 | AUX

2. To select Erase function

ENT

3. To execute Erase operation

ENT

- Press CLR to exit Erase EEPROM function

D2-HPP Display Results

```
AUX  7*  EEPROM
AUX  75  ERASE
```

```
AUX  75  ERASE
CLEAR EEPROM?
```

```
AUX  75  ERASE
BUSY
```

```
EEPROM CLEARED
```

Copying Programs from the CPU

To save System V-memory (not system parameters), you will need to modify the memory range to include the upper memory locations for the DL230 or DL240 CPUs for option 2: V — V memory. **All System V-memory is not saved when you select either the System or Program + System options.**

Option and Memory Type	DL240 Default Range	DL130 / DL230 Default Range
1:PGM — Program	$00000 - $02559	$00000 - $02047
2:V — V memory	$00000 - $4777	$00000 - $04777
3:SYS — System	Non-selectable copies system parameters	
4:etc (ALL) — Program, System and *non-volatile* V-memory only	Non-selectable	Non-selectable

Depending on the size of your program, a single EEPROM may not store your entire application. If this is the case, use more than one EEPROM, and save *only* V memory on a EEPROM by itself. Some copying options require a blank EEPROM before they will execute. If you receive the error message E621 EEPROM NOT BLANK, use AUX 75 to erase the EEPROM. Then retry the copy function.

WARNING: Do not try to store more than one of the above options in a single EEPROM, portions of data can be overwritten, yielding an unreliable copy.

Selecting Memory to copy from CPU - EEPROM

The **AUX 71** function may be used copy data from the CPU-->HPP and save to EEPROM memory. You may select different portions of CPU data to copy. Three data types may be selected, program, system, and V-memory. The following figure demonstrates how to use the AUX 71 operation to copy the PGM (program data) into the Handheld programmer EEPROM.

Press these Keystrokes	D2-HPP Display Results
1. Use AUX 71 to copy memory from CPU to HPP H 7 | B 1 | AUX	AUX 7* EEPROM AUX 71 CPU-->HPP
2. To select CPU ⇒ HPP ENT	AUX 71 CPU-->HPP PGM/V/SYS/etc
3. To select PGM (program) press enter ENT	AUX 71 CPU-->HPP PGM/V/SYS/etc
4. Else use the arrow key to choose other area types, then press ENT → | ENT	AUX 71 CPU-->HPP PGM / PGM+SYS
5. Enter the starting address to copy, or press enter for default ($00000) ENT	CPU-->EEPROM(PGM 1st $ 00000
6. Enter END program address of press ENT to select the entire range (e.g. DL130/DL230 default $02047) ENT	CPU-->EEPROM(PGM END $ 02047
7. Enter the destination EEPROM address, or press enter for default ($00000) ENT	CPU-->EEPROM(PGM DIST $ 00000
• Press CLR to exit AUX 71 Copy operation.	CPU-->EEPROM(PGM $00000-$02047->$
• This operation may take a few minutes depending on type and amount of data transferred.	CPU-->EEPROM(PGM EEPROM 08K 01 This value will increment
	OK
If you are copying to an EEPROM which is not erased you will receive this message.	CPU-->EEPROM(PGM EEPROM NOT BLANK

WARNING: Use extreme caution to prevent overwriting information during copy procedure.

Writing Programs to the CPU

The **AUX 72** function allows data to be transferred from the Handheld programmer EEPROM to the CPU memory.

The table below shows the different types of information which may be copied.

Option and Memory Type	DL240 Default Range	DL130 / DL230 Default Range
1:PGM — Program	$00000 - $02559	$00000 - $02047
2:V — V memory	$00000 - $4777	$00000 - $04777
3:SYS — System	Non-selectable (copies all system parameters)	
4:etc (ALL) — Program, System and *non-volatile* V-memory only	Non-selectable	Non-selectable

Press these Keystrokes

1. Use AUX 72 to copy memory from HPP to CPU.

H 7 | C 2 | AUX

2. To select HPP ⇒ CPU copy function.

ENT

3. To select PGM press enter.

ENT

4. Else use the arrow key to position cursor and select area desired by pressing ENT.

→ | ENT

5. Enter the starting address area to copy, or press enter for default ($00000).

ENT

6. Enter END address to copy or press enter to select entire range (DL130/DL230 default $02047).

ENT

- Press CLR key to exit AUX 72 function.
- This operation may take a few minutes depending on amount and type of data copied.

D2-HPP Display Results

```
AUX  7*  EEPROM
AUX  72  HPP-->CPU
```

```
AUX  72  HPP-->CPU
PGM/V/SYS/etc
```

```
AUX  72  HPP-->CPU
PGM  /  PGM+SYS
```

```
EEPROM-->CPU(PGM
1st    $  00000
```

```
EEPROM-->CPU(PGM
END    $  02047
```

```
EEPROM-->CPU(PGM
$00000  -  $02047?
```

```
EEPROM-->CPU(PGM
EEPROM   08K   01
```

```
OK
```

Comparing CPU and Handheld Programs

The **AUX 73** function compares the CPU and HPP programs. You may choose which areas of the program to compare, such as; program instruction (PGM), V-memory contents (V), and System (SYS) memory. The figure below demonstrates how to compare a program in the HPP to the CPU.

You can compare different types of information.

Option and Memory Type	DL240 Default Range	DL130 / DL230 Default Range
1:PGM — Program	$00000 - $02559	$00000 - $02047
2:V — V memory	$00000 - $4777	$00000 - $04777
3:SYS — System	Non-selectable copies system parameters	
4:etc (ALL) — Program, System and *non-volatile* V-memory only	Non-selectable	Non-selectable

Press these Keystrokes

1. Use AUX 73 to copy memory from CPU to HPP.
H 7 | D 3 | AUX

2. To select HPP<-> CPU compare operation
ENT

3. To select PGM press enter
ENT

4. Use the arrow keys ( ) to position cursor to other area type desired, then press ENT
→ | ENT

5. Enter the starting address area to copy, or press enter for default ($00000).
ENT

6. Enter END address to prepare copy or press enter to select entire range (DL130/DL230 default $02047).
ENT

- Press CLR key to exit from Copy function
- This Auxiliary function may take a few minutes depending on type and amount of data copied.

D2-HPP Display Results

```
AUX  7*  EEPROM
AUX  73  HPP<->CPU
```

```
AUX  73  HPP<->CPU
PGM/V/SYS/etc
```

```
AUX  73  HPP<->CPU
PGM  /  PGM+SYS
```

```
VERIFY  PGM+SYS
1st     $ 00000
```

```
VERIFY  PGM
END     $ 02047
```

```
VERIFY  PGM+SYS
DIST    $ 00000
```

```
VERIFY  PGM+SYS
$00000-$02047<->
```

```
VERIFY    PGM
EEPROM    08K    01
```

```
VERIFICATION  OK
```

Verification Errors

While running the Verification function the Handheld programmer may display one of the following verification errors. The first display example occurs if the EEPROM System is different than the CPU. If the Handheld programmer and the CPU programs are different, the display message will show the first address number which differs.

- This display appears if the System programs are different between the HPP and CPU.

Example D2-HPP Display

V	E	R	I	F	Y		P	G	M	+	S	Y	S	
			S	Y	S		V	E	R	I	F	Y	E	R

- If a Verification error occurs the display informs which address and instruction are different in the CPU.

Example D2-HPP Display

$	0	0	0	2	1	V	E	R	I	F	Y		E	R	R
S	T	R		S	P	1									

HINT: Running the Verification program is helpful to ensure PLC backups stored on EEPROM, are exact copies of those running in your PLC system(s).

Saving Offline Generated Programs

If you have been programming off-line, you may temporarily save your program in RAM memory on Handheld programmers. To save a program being generated in the Handheld programmer press the **SAVE** key.

As you've seen, entering and storing programs with the Handheld programmer is a pretty simple task. Once you've entered a program and the machine is running, you can use the Handheld programmer to monitor and change machine operations.

6

System Monitoring and Troubleshooting

In This Chapter. . . .

Troubleshooting Suggestions

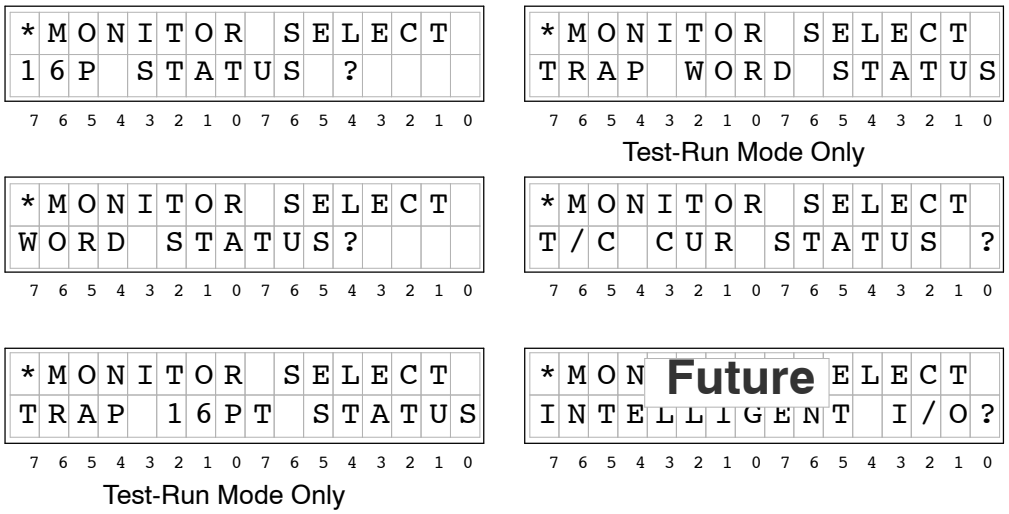
The Handheld programmer is useful for monitoring and troubleshooting your PLC and machine operation. There are several operations and features which help debug and isolate potential PLC problems. Below are some troubleshooting and maintenance features commonly used.

- **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

D2-HPP Display Results

```
*MONITOR SELECT
16P STATUS ?
```

```
16P STATUS
BIT REF C0
```

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

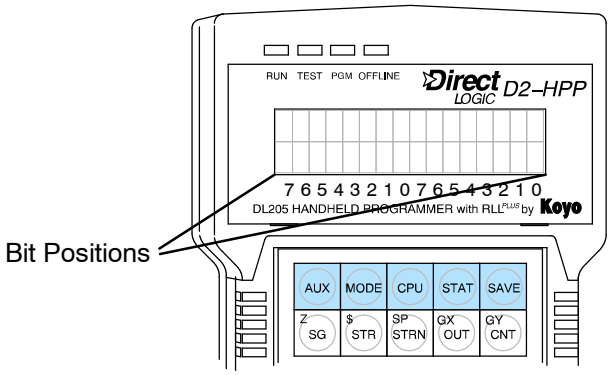
Bit Status for a Range of bits

```
   C  10  C_0
```

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

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.)

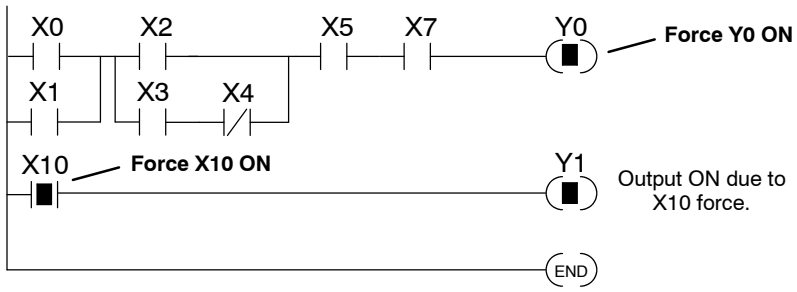
- **Regular Forcing —** This type of forcing can temporarily change the status of a discrete bit. For example, you may want to force an input on, even though it is really off. This allows you to change point status which is in the image register for one scan. **This value will be valid until the image 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

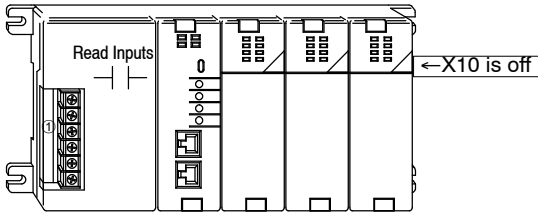


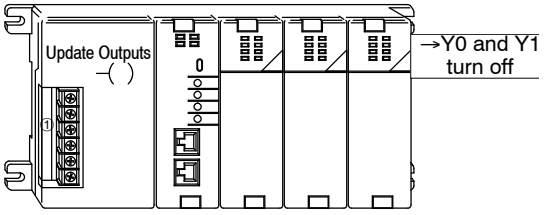
1. The CPU first reads the I/O status from the modules. If discrete input point X10 is off, the CPU overwrites the force command and turns off X10.

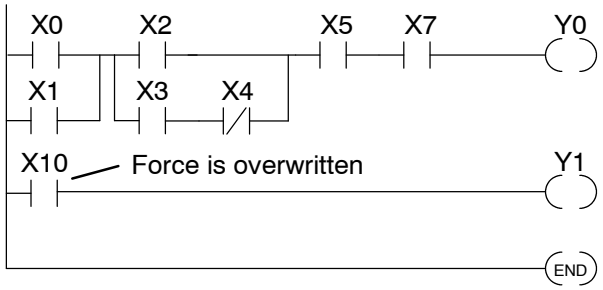
2. While X10 is off, even though previously forced on, Y0 will remain turned off. The CPU will scan and process all program instructions.

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.

CPU Process Update

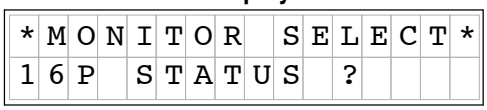
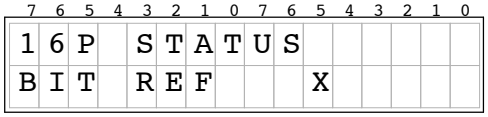
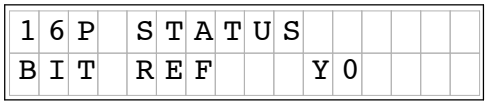
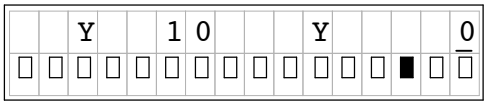
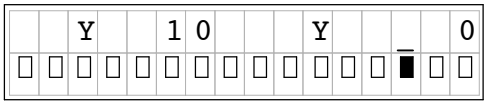
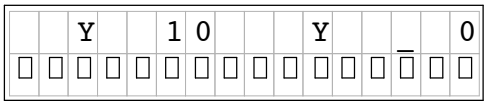






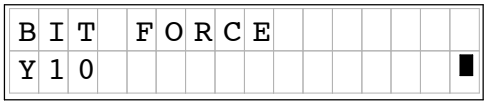
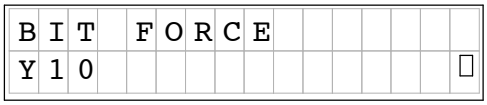
Regular Bit Force using the Status Monitor

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	D2-HPP Display Results
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	

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	D2-HPP Display Results
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	

Bit Override Indicators

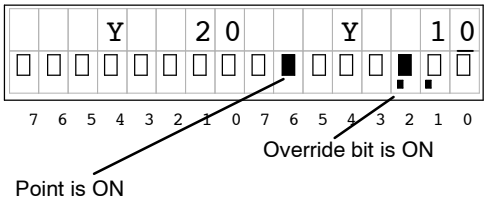
Override bit indicators are also shown on the Handheld programmer status display. Below are the keystrokes to call the status display for Y10 – Y20.

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

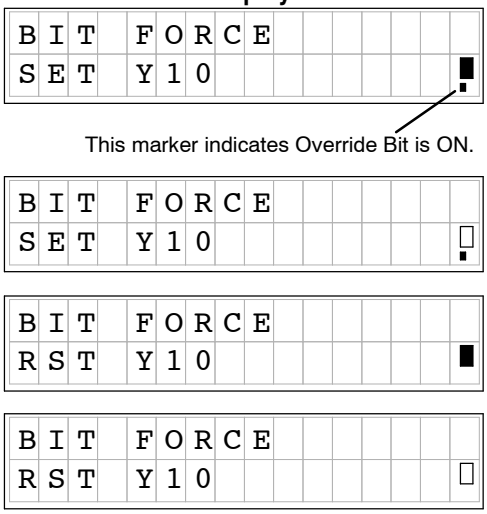
S RST | → | B 1 | A 0 | SHFT | ON INS

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

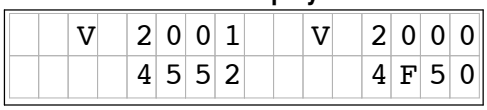
1. Select the location to monitor

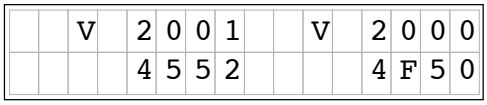
SHFT | V AND | C 2 | A 0 | A 0 | A 0 | STAT

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

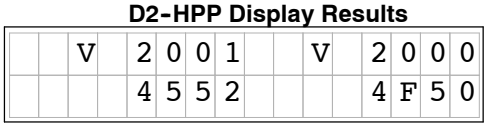
2. Use K (constant) to load a new value in memory location V2000

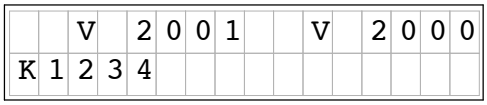
SHFT | K JMP | B 1 | C 2 | D 3 | E 4

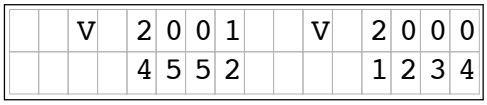
3. Press ENT to enter new value

ENT

D2-HPP Display Results







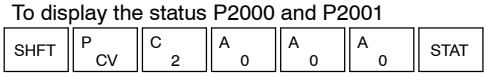
Monitoring Pointer Locations

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.

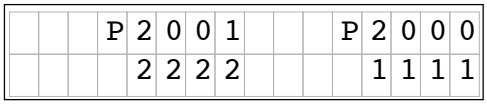
In our example V2000 has the value of 0 and V 2001 has the value of 100 (both values are in octal). At address V0 the value is 1111 and at V100 the value is 2222. When the status display is called with the pointer P2000 the values stored in memory locations V0 and V100 will be displayed, since the addresses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

Press these Keystrokes

1. To display the status P2000 and P2001



D2-HPP Display Results



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

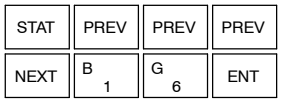
Monitoring Timer/Counter Values

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.

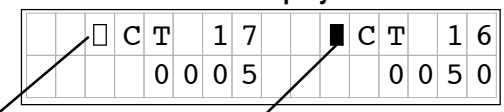
The display for the timer is similar in form to the one shown for the counter.

Press these Keystrokes

1. To display the status of CT16 - CT17



D2-HPP Display Results

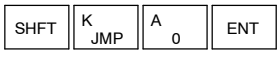


Changing Timer/Counter Current Values

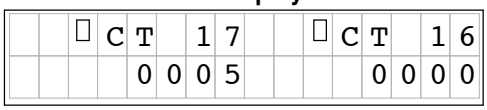
To change Timer and Counter current values is much the same as changing V-memory.

Press these Keystrokes

1. To enter a new counter current value



D2-HPP Display Results



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.

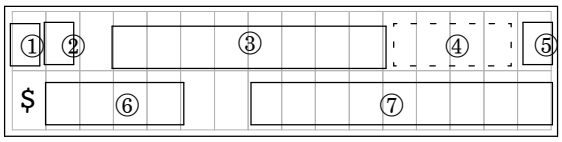
WARNING: The following items should be considered during Run Time Edits.
1. If he program has any instruction syntax errors, the CPU will *not* enter the Run Mode.
2. If you delete an output reference while the output is ON, the output will remain ON until it is forced OFF with a programming device.
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.

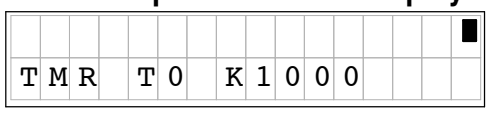
⑤ If the operand is a bit, this field displays the bit status.
■ indicates ON and **S** indicates OFF

⑥ Displays the instruction address.

⑦ Displays the mnemonic instruction and reference number

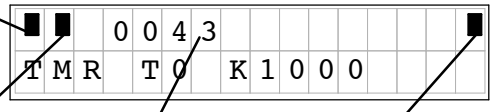
T0 Contact (S is OFF)
(■ is ON)

Example RUN Mode Display



Input to Timer

Example TEST-RUN Mode



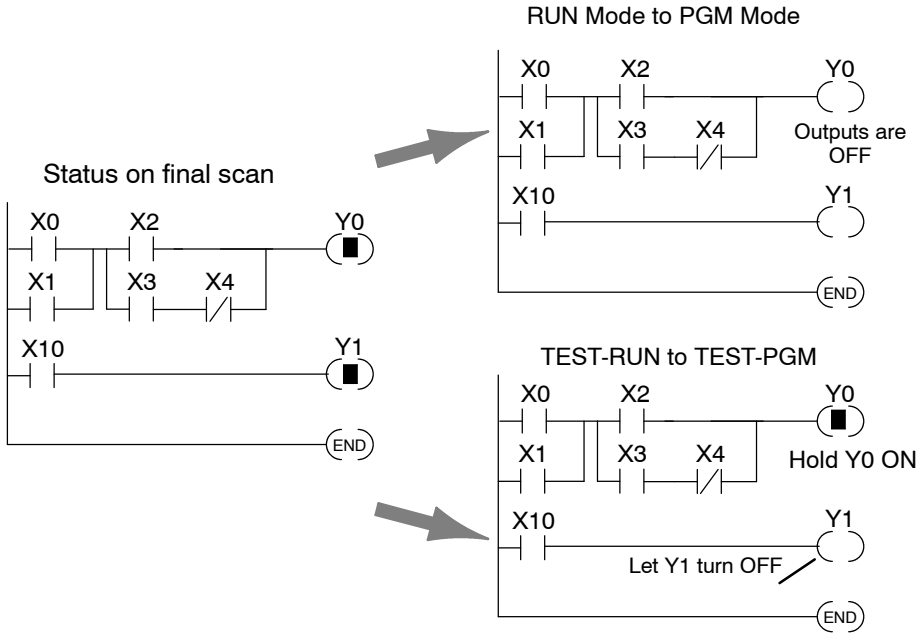
Power flow to rail

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?
```

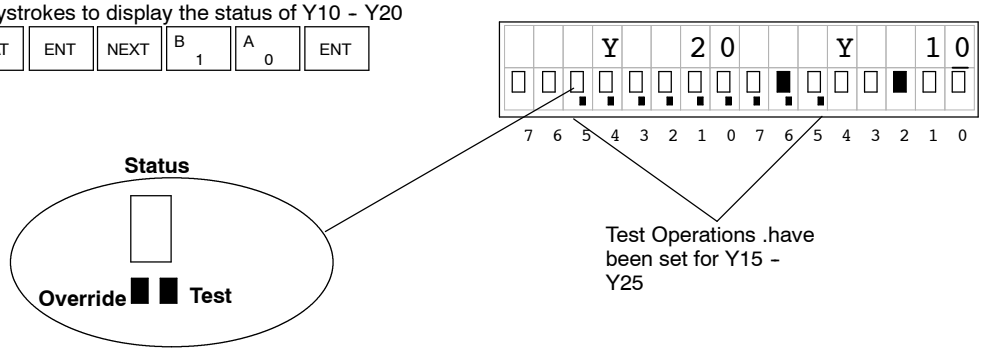
- Press the CLR key to leave AUX 58

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]

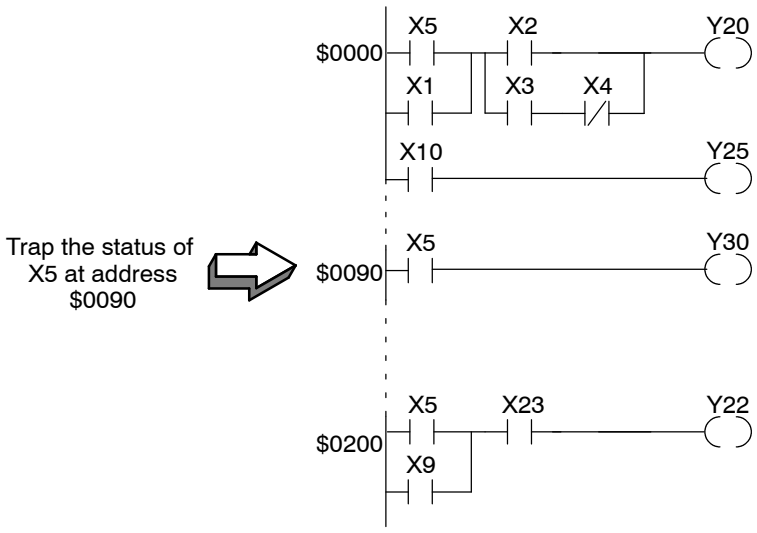


NOTE: Take care not to confuse the Test Operation marker with the marker used for Bit Override. The Test marker is a small box indicated on the right side below the Status point.

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

E 2 5 2												
N E W		I / O		C F G								

D2-HPP Display Results

A U X	4 *	I / O	C F G	
A U X	4 2	I / O	D I A G N	

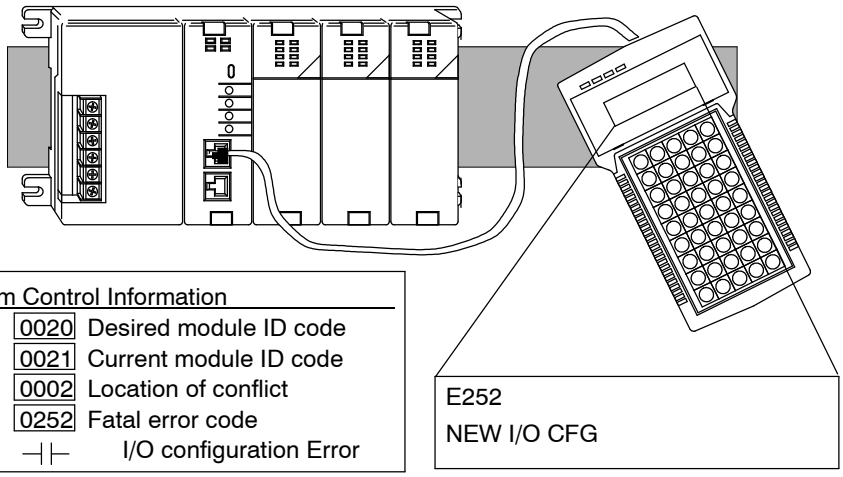
A U X	4 2	I / O	B A S E	
E 2 5 2		I / O	C O N F	

/ O	B A S E	0 / S L O T 3
I / O	C O N F I G	E r r o r

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



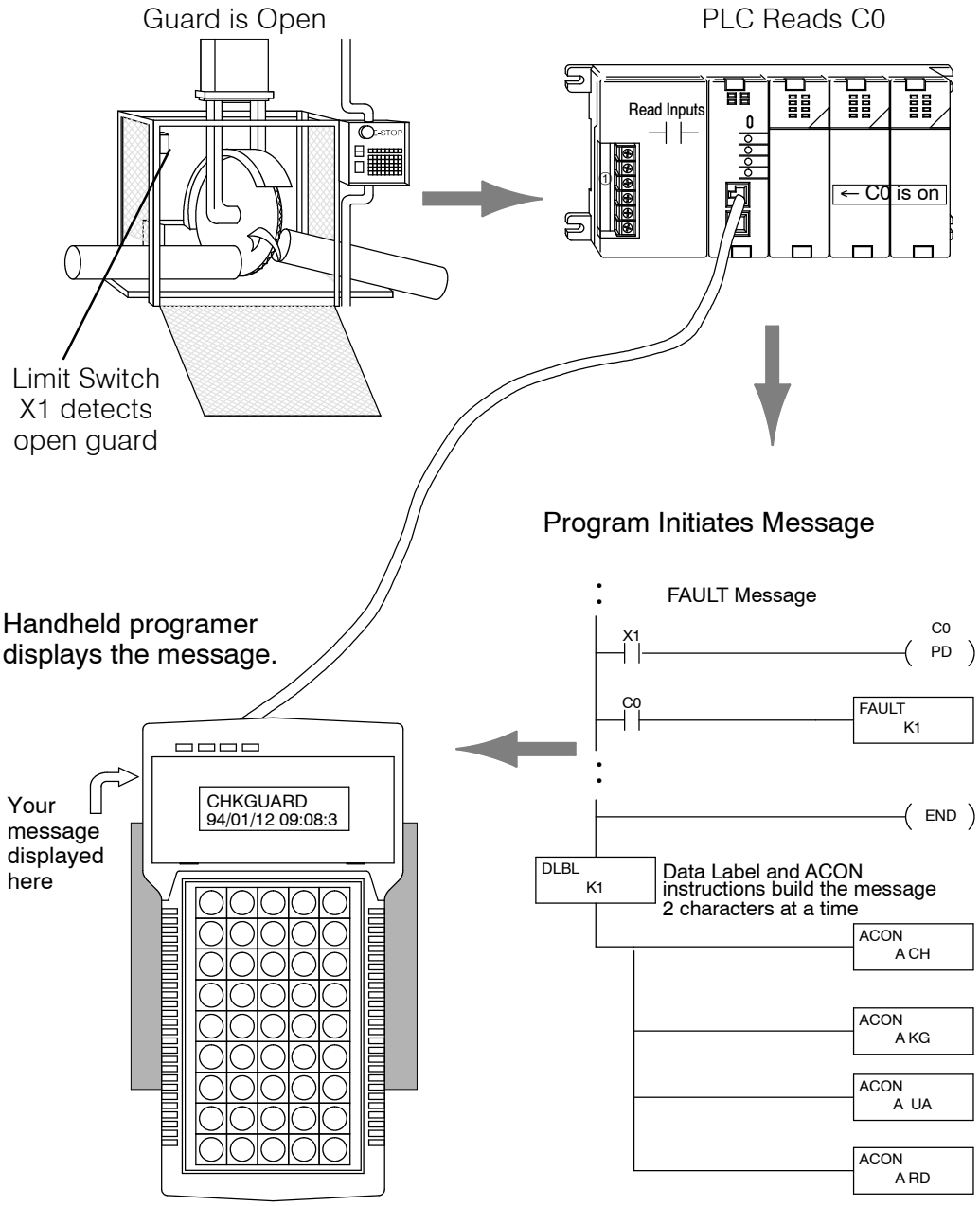
Program Control Information

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

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.

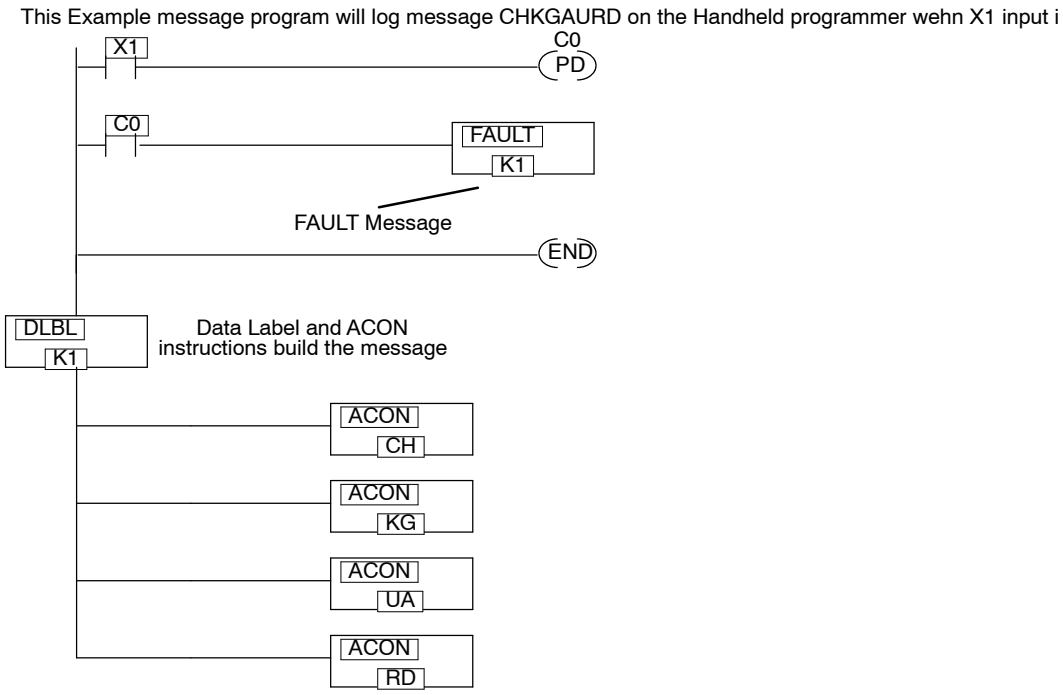


Message Instructions

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]

D2-HPP Display Results

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

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

- 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.

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]

D2-HPP Display Results

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

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

- 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.

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.

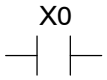
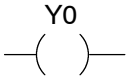
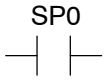
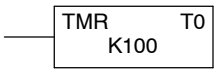
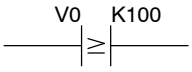
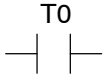
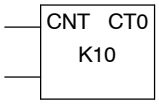
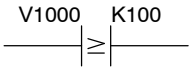
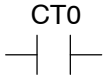
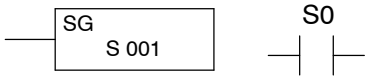
DL105/DL205 Error Code	Description
E627 BAD WRITE	An attempt to write to a write protected or faulty EEPROM was made. Check the write protect jumper and replace the EEPROM if necessary.
E628 EEPROM TYPE ERROR	The wrong size EEPROM is being used. The DL230 and DL240 CPUs use different size EEPROMs.
E640 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.

DL105/DL205 Memory Map

A

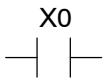
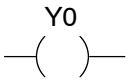
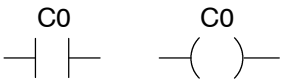
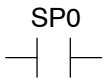
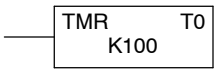
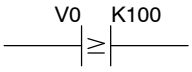
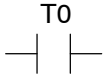
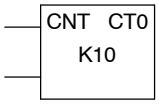
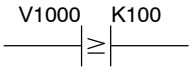
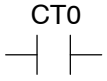
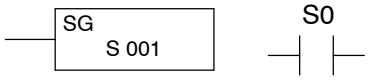
In This Chapter. . . .

DL130 Memory Map Overview

Memory Type	Discrete Memory Reference (octal)	Word Memory Reference (octal)	Qty. Decimal	Symbol
Input Points	X0 – X177	V40400 – V40407	128	
Output Points	Y0 – Y177	V40500 – V40507	128	
Control Relays	C0 – C377	V40600 – V40617	256	
Special Relays	SP0 – SP117 SP540 – SP577	V41200 – V41204 V41226 – V41227	112	
Timers	T0 – T77		64	
Timer Current Values	None	V0 – V77	64	
Timer Status Bits	T0 – T77	V41100 – V41103	64	
Counters	CT0 – CT77		64	
Counter Current Values	None	V1000 – V1077	64	
Counter Status Bits	CT0 – CT77	V41140 – V41143	64	
Data Words	None	V2000 – V2377	256	None specific, used with many instructions
Data Words Non–volatile	None	V4000 – V4177	128	None specific, used with many instructions
Stages	S0 – S377	V41000 – V41017	256	
System V-memory	None	V7620 – V7647 V7750–V7777	48	None specific, used for various purposes

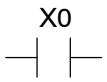
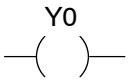
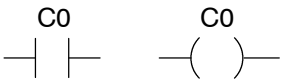
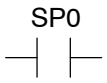
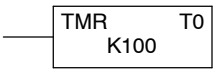
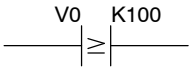
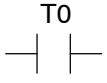
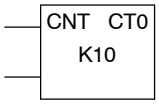
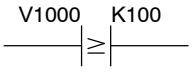
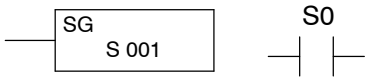
1 – The DL105 systems are limited to 10 discrete Inputs and 8 descrete outputs. There are 8 different DL105 models whcih are configured with various voltage level capabilities. Please refer to the Product Catalog or DL105 User Manual for specific models and specifications.

DL230 Memory Map Overview

Memory Type	Discrete Memory Reference (octal)	Word Memory Reference (octal)	Qty. Decimal	Symbol
Input Points	X0 – X177	V40400 – V40407	128	
Output Points	Y0 – Y177	V40500 – V40507	128	
Control Relays	C0 – C377	V40600 – V40617	256	
Special Relays	SP0 – SP117 SP540 – SP577	V41200 – V41204 V41226 – V41227	112	
Timers	T0 – T77		64	
Timer Current Values	None	V0 – V77	64	
Timer Status Bits	T0 – T77	V41100 – V41103	64	
Counters	CT0 – CT77		64	
Counter Current Values	None	V1000 – V1077	64	
Counter Status Bits	CT0 – CT77	V41140 – V41143	64	
Data Words	None	V2000 – V2377	256	None specific, used with many instructions
Data Words Non–volatile	None	V4000 – V4177	128	None specific, used with many instructions
Stages	S0 – S377	V41000 – V41017	256	
System V-memory	None	V7620 – V7647 V7750–V7777	48	None specific, used for various purposes

1 – The DL205 systems are limited to 128 discrete I/O points (total) with the present system hardware available. These can be mixed between input and output points as necessary.

DL240 Memory Map Overview

Memory Type	Discrete Memory Reference (octal)	Word Memory Reference (octal)	Qty. Decimal	Symbol
Input Points	X0 – X177	V40400 – V40407	128 [1]	
Output Points	Y0 – Y177	V40500 – V40507	128 [1]	
Control Relays	C0 – C377	V40600 – V40617	256	
Special Relays	SP0 – SP137 SP540 – SP617	V41200 – V41205 V41226 – V41230	144	
Timers	T0 – T177		128	
Timer Current Values	None	V0 – V177	128	
Timer Status Bits	T0 – T177	V41100 – V41107	128	
Counters	CT0 – CT177		128	
Counter Current Values	None	V1000 – V1177	128	
Counter Status Bits	CT0 – CT177	V41140 – V41147	128	
Data Words	None	V2000 – V3777	1024	None specific, used with many instructions
Data Words Non–volatile	None	V4000 – V4377	256	None specific, used with many instructions
Stages	S0 – S777	V41000 – V41037	512	
System V-memory	None	V7620 – V7737 V7746–V7777	106	None specific, used for various purposes

1 – The DL205 systems are limited to 128 discrete I/O points (total) with the present system hardware available. These can be mixed between input and output points as necessary.

X Input Bit Map

This table provides a listing of the individual Input points associated with each V-memory address bit for the DL130, 230 and DL240 CPUs.

MSB															LSB	Address
DL130/DL230/DL240 Input (X) Points																
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40400
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40401
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40402
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40403
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40404
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40405
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40406
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40407

Y Output Bit Map

This table provides a listing of the individual output points associated with each V-memory address bit for both the DL130, DL230 and DL240 CPUs.

MSB															LSB	Address
DL130/DL230/DL240 Output (Y) Points																
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40500
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40501
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40502
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40503
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40504
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40505
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40506
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40507

Control Relay Bit Map

This table provides a listing of the individual control relays associated with each V-memory address bit.

MSB															LSB	Address
DL130/DL230/DL240 Control Relays (C)																
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40600
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40601
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40602
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40603
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40604
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40605
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40606
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40607
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V40610
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V40611
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V40612
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V40613
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V40614
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V40615
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V40616
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V40617

Stage Control / Status Bit Map

This table provides a listing of the individual stage control bits associated with each V-memory address bit.

MSB							DL130/DL230/DL240 Stage (S) Control Bits								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41000
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41001
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41002
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41003
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41004
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41005
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41006
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41007
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V41010
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V41011
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V41012
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V41013
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V41014
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V41015
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V41016
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V41017

MSB							DL240 Additional Stage (S) Control Bits								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V41020
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V41021
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V41022
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V41023
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V41024
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V41025
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V41026
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V41027
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V41030
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V41031
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V41032
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V41033
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V41034
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V41035
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V41036
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V41037

Timer Status Bit Map

This table provides a listing of the individual timer contacts associated with each V-memory address bit.

MSB							DL130/DL230/DL240 Timer (T) Contacts								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41100
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41101
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41102
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41103

MSB							Additional DL240 Timer (T) Contacts								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41104
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41105
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41106
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41107

Counter Status Bit Map

This table provides a listing of the individual counter contacts associated with each V-memory address bit.

MSB							DL130/DL230/DL240 Counter (CT) Contacts								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41140
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41141
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41142
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41143

MSB							Additional DL240 Counter (CT) Contacts								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41144
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41145
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41146
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41147

DL130/DL230 System V-memory

The DL205 CPUs reserve several V-memory locations for storing system parameters or certain types of system data. These memory locations store things like the error codes, counter interface module data, and other types of system setup information.

System V-memory	Description of Contents	Default Values / Ranges
V2320–V2377	The default location for multiple preset values for the UP counter.	N/A
V7620–V7627	Locations for DV–1000 operator interface parameters	
V7620	Sets the V-memory location that contains the value.	V0 – V2377
V7621	Sets the V-memory location that contains the message.	V0 – V2377
V7622	Sets the total number (1 – 16) of V-memory locations to be displayed.	1 – 16
V7623	Sets the V-memory location that contains the numbers to be displayed.	V0 – V2377
V7624	Sets the V-memory location that contains the character code to be displayed.	V0 – V2377
V7625	Contains the function number that can be assigned to each key.	V-memory location for X, Y, or C points used.
V7626	Reserved for future use.	
V7627	Reserved for future use.	
V7630	Starting location for the multi–step presets for channel 1. The default value is 2320, which indicates the first value should be obtained from V2320. Since there are 24 presets available, the default range is V2320 – V2377. You can change the starting point if necessary.	Default: V2320 Range: V0 – V2320
V7631–V7632	Not used	N/A
V7633	Sets the desired function code for the high speed counter, interrupt, pulse catch, pulse train, and input filter. Location is also used for setting the with/without battery option, enable/disable CPU mode change, and power-up in Run Mode option.	Default: 0000 Lower Byte Range: Range: 0 – None 10 – Up 40 – Interrupt 50 – Pulse Catch 60 – Filtered discrete In. Upper Byte Range: Bits 8 – 11, 14,15: Unused Bit 12: With/Without Batt. Bit 13: Power-up in Run
V7634	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X0 (when D2–CNTINT is installed).	Default: 0000
V7635	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X1 (when D2–CNTINT is installed).	Default: 0000
V7636	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X2 (when D2–CNTINT is installed).	Default: 0000
V7637	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X3 (when D2–CNTINT is installed).	Default: 0000

System V-memory	Description of Contents	Default Values / Ranges
V7640–V7647	Not used	N/A
V7751	Fault Message Error Code — stores the 4-digit code used with the FAULT instruction when the instruction is executed.	N/A
V7752	I/O Configuration Error — stores the module ID code for the module that does not match the current configuration.	N/A
V7753	I/O Configuration Error — stores the correct module ID code.	
V7754	I/O Configuration Error — identifies the base and slot number.	
V7755	Error code — stores the fatal error code.	
V7756	Error code — stores the major error code.	
V7757	Error code — stores the minor error code.	
V7760–V7764	Module Error — stores the slot number and error code where an I/O error occurs.	
V7765	Scan — stores the total number of scan cycles that have occurred since the last Program Mode to Run Mode transition.	
V7666–V7774	Not used	N/A
V7775	Scan — stores the current scan time (milliseconds).	N/A
V7776	Scan — stores the minimum scan time that has occurred since the last Program Mode to Run Mode transition (milliseconds).	N/A
V7777	Scan — stores the maximum scan time that has occurred since the last Program Mode to Run Mode transition (milliseconds).	N/A

DL240 System* V-memory

The DL205 CPUs reserve several V-memory locations for storing system parameters or certain types of system data. These memory locations store things like the clock / calendar information, analog potentiometer current values, error codes, and other types of system setup information.

System V-memory	Description of Contents	Default Values / Ranges
V3630–V3707	The default location for multiple preset values for UP/DWN and UP counter 1 or pulse catch function.	N/A
V3710–V3767	The default location for multiple preset values for UP/DWN and UP counter 2.	N/A
V3770–V3773	Not used	N/A
V3774–V3777	Default locations for analog potentiometer data (channels 1–4, respectively).	Range: 0 – 9999
V7620–V7627	Locations for DV–1000 operator interface parameters	
V7620	Sets the V-memory location that contains the value.	V0 – V3760
V7621	Sets the V-memory location that contains the message.	V0 – V3760
V7622	Sets the total number (1 – 16) of V-memory locations to be displayed.	1 – 16
V7623	Sets the V-memory location that contains the numbers to be displayed.	V0 – V3760
V7624	Sets the V-memory location that contains the character code to be displayed.	V0 – V3760
V7625	Contains the function number that can be assigned to each key.	V-memory location for X, Y, or C points used.
V7626	Reserved for future use.	
V7627	Reserved for future use.	
V7630	Starting location for the multi–step presets for channel 1. Since there are 24 presets available, the default range is V3630 – V3707. You can change the starting point if necessary.	Default: V3630 Range: V0 – V3710
V7631	Starting location for the multi–step presets for channel 1. Since there are 24 presets available, the default range is V3710– 3767. You can change the starting point if necessary.	Default: V3710 Range: V0 – V3710
V7632	Contains the baud rate setting for Port 2. You can use AUX 56 (from the Handheld Programmer) or, use ***Direct***SOFT™ to set the port parameters if 9600 baud is unacceptable.	Default: 2 – 9600 baud Range: 0 = 300 1 = 1200 2 = 9600 3 = 19.2K
V7633	Sets the desired function code for the high speed counter, interrupt, pulse catch, pulse train, and input filter. Location is also used for setting the with/without battery option, enable/disable CPU mode change, and power-up in Run Mode option.	Default: 0000 Lower Byte Range: Range: 0 – None 10 – Up 20 – Up/Dwn. 30 – Pulse Out 40 – Interrupt 50 – Pulse Catch 60 – Filtered Dis. Upper Byte Range: Bits 8 – 11, 13, 15 Unused Bit 12: With/Without Batt. Bit 14: Mode chg. enable
V7634	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X0 (when D2–CNTINT is installed).	Default: 0000

System V-memory	Description of Contents	Default Values / Ranges
V7635	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X1 (when D2-CNTINT is installed).	Default: 0000
V7636	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X2 (when D2-CNTINT is installed).	Default: 0000
V7637	Contains set up information for high speed counter, interrupt, pulse catch, pulse train output, and input filter for X3 (when D2-CNTINT is installed).	Default: 0000
V7640-V7641	Location for setting the lower and upper limits for the CH1 analog pot.	Default: 0000 Range: 0 - 9999
V7642-V7643	Location for setting the lower and upper limits for the CH2 analog pot.	Default: 0000 Range: 0 - 9999
V7644-V7645	Location for setting the lower and upper limits for the CH3 analog pot.	Default: 0000 Range: 0 - 9999
V7646-V7647	Location for setting the lower and upper limits for the CH4 analog pot.	Default: 0000 Range: 0 - 9999
V7650-V7737	Locations reserved for set up information used with future options (such as remote I/O and data communications.)	
V7746	Location contains the battery voltage, accurate to 0.1V. For example, a value of 32 indicates 3.2 volts.	
V7747	Location contains a 10ms counter. This location increments once every 10ms.	
V7751	Fault Message Error Code — stores the 4-digit code used with the FAULT instruction when the instruction is executed. If you've used ASCII messages (DL240 only) then the data label (DLBL) reference number for that message is stored here.	
V7752	I/O configuration Error — stores the module ID code for the module that does not match the current configuration.	
V7753	I/O Configuration Error — stores the correct module ID code.	
V7754	I/O Configuration Error — identifies the base and slot number.	
V7755	Error code — stores the fatal error code.	
V7756	Error code — stores the major error code.	
V7757	Error code — stores the minor error code.	
V7760-V7764	Module Error — stores the slot number and error code where an I/O error occurs.	
V7765	Scan — stores the total number of scan cycles that have occurred since the last Program Mode to Run Mode transition.	
V7766	Contains the number of seconds on the clock. (00 to 59).	
V7767	Contains the number of minutes on the clock. (00 to 59).	
V7770	Contains the number of hours on the clock. (00 to 23).	
V7771	Contains the day of the week. (Mon, Tue, etc.).	
V7772	Contains the day of the month (1st, 2nd, etc.).	
V7773	Contains the month. (01 to 12)	
V7774	Contains the year. (00 to 99)	
V7775	Scan — stores the current scan time (milliseconds).	
V7776	Scan — stores the minimum scan time that has occurred since the last Program Mode to Run Mode transition (milliseconds).	
V7777	Scan — stores the maximum scan time that has occurred since the last Program Mode to Run Mode transition (milliseconds).	

Special Relays

B

In This Chapter. . . .

DL130/DL230 CPU Special Relays

Startup and Real-Time Relays

SP0	First scan	on for the first scan after a power cycle or program to run transition only. The relay is reset to off on the second scan. It is useful where a function needs to be performed only on program startup.
SP1	Always ON	provides a contact to insure an instruction is executed every scan.
SP3	1 minute clock	on for 30 seconds and off for 30 seconds.
SP4	1 second clock	on for 0.5 second and off for 0.5 second.
SP5	100 ms clock	on for 50 ms. and off for 50 ms.
SP6	50 ms clock	on for 25 ms. and off for 25 ms.
SP7	Alternate scan	on every other scan.

CPU Status Relays

SP12	Terminal run mode	on when the CPU is in the run mode.
SP16	Terminal program mode	on when the CPU is in the program mode.
SP20	Forced stop mode	on when the STOP instruction is executed.
SP22	Interrupt enabled	on when interrupts have been enabled using the ENI instruction.

System Monitoring

SP40	Critical error	on when a critical error such as I/O communication loss has occurred.
SP41	Warning	on when a non critical error such as a low battery has occurred.
SP43	Battery low	on when the CPU battery voltage is low.
SP44	Program memory error	on when a memory error such as a memory parity error has occurred.
SP45	I/O error	on when an I/O error occurs. For example, an I/O module is withdrawn from the base, or an I/O bus error is detected.
SP47	I/O configuration error	on if an I/O configuration error has occurred. The CPU power-up I/O configuration check must be enabled before this relay will be functional.
SP50	Fault instruction	on when a Fault Instruction is executed.
SP51	Watch Dog timeout	on if the CPU Watch Dog timer times out.
SP52	Grammatical error	on if a grammatical error has occurred either while the CPU is running or if the syntax check is run. V7755 will hold the exact error code.
SP53	Solve logic error	on if CPU cannot solve the logic.

Accumulator Status

SP60	Value less than	on when the accumulator value is less than the instruction value.
SP61	Value equal to	on when the accumulator value is equal to the instruction value.
SP62	Greater than	on when the accumulator value is greater than the instruction value.
SP63	Zero	on when the result of the instruction is zero (in the accumulator.)
SP64	Half borrow	on when the 16 bit subtraction instruction results in a borrow.
SP65	Borrow	on when the 32 bit subtraction instruction results in a borrow.
SP66	Half carry	on when the 16 bit addition instruction results in a carry.
SP67	Carry	when the 32 bit addition instruction results in a carry.
SP70	Sign	on anytime the value in the accumulator is negative.
SP71	Invalid octal number	on when an Invalid octal number was entered. This also occurs when the V-memory specified by a pointer (P) is not valid.
SP73	Overflow	on if overflow occurs in the accumulator when a signed addition or subtraction results in an incorrect sign bit.
SP75	Data error	on if a BCD number is expected and a non–BCD number is encountered.
SP76	Load zero	on when any instruction loads a value of zero into the accumulator.

Counter Interface Module Relays

SP100	X0 is on	X0 — on when corresponding input is on.

Equal Relays for Multi-step Presets with Up/Down Counter #1 (for use with a Counter Interface Module)

SP540	Current = target value	on when the counter current value equals the value in V3640.
SP541	Current = target value	on when the counter current value equals the value in V3642.
SP542	Current = target value	on when the counter current value equals the value in V3644.
SP543	Current = target value	on when the counter current value equals the value in V3646.
SP544	Current = target value	on when the counter current value equals the value in V3650.
SP545	Current = target value	on when the counter current value equals the value in V3652.
SP546	Current = target value	on when the counter current value equals the value in V3654.
SP547	Current = target value	on when the counter current value equals the value in V3656.
SP550	Current = target value	on when the counter current value equals the value in V3660.
SP551	Current = target value	on when the counter current value equals the value in V3662.
SP552	Current = target value	on when the counter current value equals the value in V3664.
SP553	Current = target value	on when the counter current value equals the value in V3666.
SP554	Current = target value	on when the counter current value equals the value in V3670.
SP555	Current = target value	on when the counter current value equals the value in V3672.
SP556	Current = target value	on when the counter current value equals the value in V3674.
SP557	Current = target value	on when the counter current value equals the value in V3676.
SP560	Current = target value	on when the counter current value equals the value in V3700.
SP561	Current = target value	on when the counter current value equals the value in V3702.
SP562	Current = target value	on when the counter current value equals the value in V3704.
SP563	Current = target value	on when the counter current value equals the value in V3706.
SP564	Current = target value	on when the counter current value equals the value in V3710.
SP565	Current = target value	on when the counter current value equals the value in V3712.
SP566	Current = target value	on when the counter current value equals the value in V3714.
SP567	Current = target value	on when the counter current value equals the value in V3716.

DL240 CPU Special Relays

Startup and Real-Time Relays

SP0	First scan	on for the first scan after a power cycle or program to run transition only. The relay is reset to off on the second scan. It is useful where a function needs to be performed only on program startup.
SP1	Always ON	provides a contact to insure an instruction is executed every scan.
SP3	1 minute clock	on for 30 seconds and off for 30 seconds.
SP4	1 second clock	on for 0.5 second and off for 0.5 second.
SP5	100 ms clock	on for 50 ms. and off for 50 ms.
SP6	50 ms clock	on for 25 ms. and off for 25 ms.
SP7	Alternate scan	on every other scan.

CPU Status Relays

SP11	Forced run mode	on anytime the CPU switch is in the RUN position.
SP12	Terminal run mode	on when the CPU switch is in the TERM position and the CPU is in the RUN mode.
SP13	Test run mode	on when the CPU switch is in the TERM position and the CPU is in the test RUN mode.
SP15	Test program mode	on when the CPU is in the TERM position and the CPU is in the TEST PROGRAM MODE.
SP16	Terminal program mode	on when the CPU switch is in the TERM position and the CPU is in the PROGRAM MODE.
SP20	Forced stop mode	on when the STOP instruction is executed.
SP22	Interrupt enabled	on when interrupts have been enabled using the ENI instruction.

System Monitoring Relays

SP40	Critical error	on when a critical error such as I/O communication loss has occurred.
SP41	Warning	on when a non-critical error such as a low battery has occurred.
SP43	Battery low	on when the CPU battery voltage is low.
SP44	Program memory error	on when a memory error such as a memory parity error has occurred.
SP45	I/O error	on when an I/O error occurs. For example, an I/O module is withdrawn from the base, or an I/O bus error is detected.
SP46	Communications error	on when a communications error has occurred on any of the CPU ports.
SP47	I/O configuration error	on if an I/O configuration error has occurred. The CPU power-up I/O configuration check must be enabled before this relay will be functional.
SP50	Fault instruction	on when a Fault Instruction is executed.
SP51	Watch Dog timeout	on if the CPU Watch Dog timer times out.
SP52	Grammatical error	on if a grammatical error has occurred either while the CPU is running or if the syntax check is run. V7755 contains the exact error code.
SP53	Solve logic error	on if CPU cannot solve the logic.
SP54	Intelligent I/O error	on when communications with an intelligent module has occurred.

Accumulator Status Relays

SP60	Value less than	on when the accumulator value is less than the instruction value.
SP61	Value equal to	on when the accumulator value is equal to the instruction value.
SP62	Greater than	on when the accumulator value is greater than the instruction value.
SP63	Zero	on when the result of the instruction is zero (in the accumulator.)
SP64	Half borrow	on when the 16 bit subtraction instruction results in a borrow.
SP65	Borrow	on when the 32 bit subtraction instruction results in a borrow.
SP66	Half carry	on when the 16 bit addition instruction results in a carry.
SP67	Carry	when the 32 bit addition instruction results in a carry.
SP70	Sign	on anytime the value in the accumulator is negative.
SP71	Invalid octal number	on when an Invalid octal number was entered. This also occurs when the V-memory specified by a pointer (P) is not valid.
SP73	Overflow	on if overflow occurs in the accumulator when a signed addition or subtraction results in a incorrect sign bit.
SP75	Data error	on if a BCD number is expected and a non–BCD number is encountered.
SP76	Load zero	on when any instruction loads a value of zero into the accumulator.

Counter Interface Module Relays

SP100	X0 is on	X0 — on when corresponding input is on.
SP101	X1 is on	X1 — on when corresponding input is on.
SP102	X2 is on	X2 — on when corresponding input is on.
SP103	X3 is on	X3 — on when corresponding input is on.

Communications Monitoring Relays

SP116	CPU communication	on when the CPU is communicating with another device
SP120	Module busy Slot 0	on when the communication module in slot 0 is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy .
SP121	Com. error Slot 0	on when the communication module in slot 0 of the local base has encountered a communication error.
SP122	Module busy Slot 1	on when the communication module in slot 1 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP123	Com. error Slot 1	on when the communication module in slot 1 of the local base has encountered a communication error.
SP124	Module busy Slot 2	on when the communication module in slot 2 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP125	Com. error Slot 2	on when the communication module in slot 2 of the local base has encountered a communication error.
SP126	Module busy Slot 3	on when the communication module in slot 3 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP127	Com. error Slot 3	on when the communication module in slot 3 of the local base has encountered a communication error.
SP130	Module busy Slot 4	on when the communication module in slot 4 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP131	Com. error Slot 4	on when the communication module in slot 4 of the local base has encountered a communication error.
SP132	Module busy Slot 5	on when the communication module in slot 5 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP133	Com. error Slot 5	on when the communication module in slot 5 of the local base has encountered a communication error.
SP134	Module busy Slot 6	on when the communication module in slot 6 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP135	Com. error Slot 6	on when the communication module in slot 6 of the local base has encountered a communication error.
SP136	Module busy Slot 7	on when the communication module in slot 7 of the local base is busy transmitting or receiving. You must use this relay with the RX or WX instructions to prevent attempting to execute a RX or WX while the module is busy.
SP137	Com. error Slot 7	on when the communication module in slot 7 of the local base has encountered a communication error.

Equal Relays for Multi-step Presets with Up/Down Counter #1 (for use with a Counter Interface Module)

SP540	Current = target value	on when the counter current value equals the value in V3640.
SP541	Current = target value	on when the counter current value equals the value in V3642.
SP542	Current = target value	on when the counter current value equals the value in V3644.
SP543	Current = target value	on when the counter current value equals the value in V3646.
SP544	Current = target value	on when the counter current value equals the value in V3650.
SP545	Current = target value	on when the counter current value equals the value in V3652.
SP546	Current = target value	on when the counter current value equals the value in V3654.
SP547	Current = target value	on when the counter current value equals the value in V3656.
SP550	Current = target value	on when the counter current value equals the value in V3660.
SP551	Current = target value	on when the counter current value equals the value in V3662.
SP552	Current = target value	on when the counter current value equals the value in V3664.
SP553	Current = target value	on when the counter current value equals the value in V3666.
SP554	Current = target value	on when the counter current value equals the value in V3670.
SP555	Current = target value	on when the counter current value equals the value in V3672.
SP556	Current = target value	on when the counter current value equals the value in V3674.
SP557	Current = target value	on when the counter current value equals the value in V3676.
SP560	Current = target value	on when the counter current value equals the value in V3700.
SP561	Current = target value	on when the counter current value equals the value in V3702.
SP562	Current = target value	on when the counter current value equals the value in V3704.
SP563	Current = target value	on when the counter current value equals the value in V3706.
SP564	Current = target value	on when the counter current value equals the value in V3710.
SP565	Current = target value	on when the counter current value equals the value in V3712.
SP566	Current = target value	on when the counter current value equals the value in V3714.
SP567	Current = target value	on when the counter current value equals the value in V3716.

Equal Relays for Multi-step Presets with Up/Down Counter #2 (for use with a Counter Interface Module)

SP570	Current = target value	on when the counter current value equals the value in V3720.
SP571	Current = target value	on when the counter current value equals the value in V3722.
SP572	Current = target value	on when the counter current value equals the value in V3724.
SP573	Current = target value	on when the counter current value equals the value in V3726.
SP574	Current = target value	on when the counter current value equals the value in V3730.
SP575	Current = target value	on when the counter current value equals the value in V3732.
SP576	Current = target value	on when the counter current value equals the value in V3734.
SP577	Current = target value	on when the counter current value equals the value in V3736.
SP600	Current = target value	on when the counter current value equals the value in V3740.
SP601	Current = target value	on when the counter current value equals the value in V3742.
SP602	Current = target value	on when the counter current value equals the value in V3744.
SP603	Current = target value	on when the counter current value equals the value in V3746.
SP604	Current = target value	on when the counter current value equals the value in V3750.
SP605	Current = target value	on when the counter current value equals the value in V3752.
SP606	Current = target value	on when the counter current value equals the value in V3754.
SP607	Current = target value	on when the counter current value equals the value in V3756.
SP610	Current = target value	on when the counter current value equals the value in V3760.
SP611	Current = target value	on when the counter current value equals the value in V3762.
SP612	Current = target value	on when the counter current value equals the value in V3764.
SP613	Current = target value	on when the counter current value equals the value in V3766.
SP614	Current = target value	on when the counter current value equals the value in V3770.
SP615	Current = target value	on when the counter current value equals the value in V3772.
SP616	Current = target value	on when the counter current value equals the value in V3774.
SP617	Current = target value	on when the counter current value equals the value in V3776.

Index

F

I

K

L

M

N

O

P

R

S

T

V

W