

MAINTENANCE AND TROUBLESHOOTING



In This Chapter...

| | |
|--|------|
| Hardware Maintenance..... | 9-2 |
| Diagnostics | 9-3 |
| CPU Error Indicators..... | 9-10 |
| PWR Indicator | 9-11 |
| Communications Problems | 9-13 |
| I/O Module Troubleshooting..... | 9-14 |
| Noise Troubleshooting..... | 9-17 |
| Machine Startup and Program Troubleshooting..... | 9-18 |

Hardware Maintenance

Standard Maintenance

The DL205 is a low maintenance system requiring only a few periodic checks to help reduce the risk of problems. Routine maintenance checks should be made regarding two key items.

- Air quality (cabinet temperature, airflow, etc), and
- CPU battery.

Air Quality Maintenance

The quality of the air your system is exposed to can affect system performance. If you have placed your system in an enclosure, check to see that the ambient temperature is not exceeding the operating specifications. If there are filters in the enclosure, clean or replace them as necessary to ensure adequate airflow. A good rule of thumb is to check your system environment every one to two months. Make sure the DL205 is operating within the system operating specifications.

Low Battery Indicator

The CPU has a battery LED that indicates the battery voltage is low. You should check this indicator periodically to determine if the battery needs replacing. You can also detect low battery voltage from within the CPU program. SP43 is a special relay that comes on when the battery needs to be replaced. If you are using a DL240 CPU, you can also use a programming device or operator interface to determine the battery voltage. V7746 contains the battery voltage. For example, a value of 32 in V7746 would indicate a battery voltage of 3.2V.

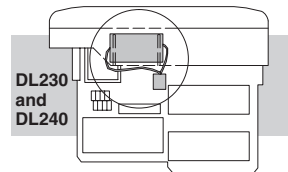
CPU Battery Replacement

The CPU battery is used to retain program V-memory and the system parameters. The life expectancy of this battery is five years.

NOTE: Before installing or replacing your CPU battery, back up your V-memory and system parameters. You can do this by using DirectSOFT to save the program, V-memory, and system parameters to hard/floppy disk on a personal computer.

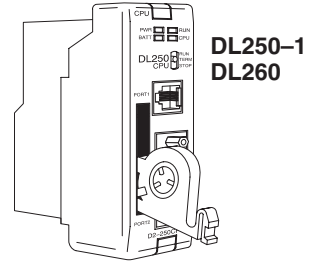
To install the D2-BAT CPU battery in DL230 or DL240 CPUs:

1. Gently push the battery connector onto the circuit board connector.
2. Push the battery into the retaining clip. Don't use excessive force. You may break the retaining clip.
3. Make a note of the date the battery was installed.



To install the D2–BAT–1 CPU battery in the DL250–1 and DL260 CPUs: (#CR2354)

1. Press the retaining clip on the battery door down and swing the battery door open.
2. Remove the old battery and insert the new battery into the coin–type slot with the larger (+) side outwards.
3. Close the battery door, making sure that it locks securely in place.
4. Make a note of the date the battery was installed.



WARNING: Do not attempt to recharge the battery or dispose of an old battery by fire. The battery may explode or release hazardous materials.

Diagnostics

Diagnostics

Your DL205 system performs many pre-defined diagnostic routines with every CPU scan. The diagnostics have been designed to detect various types of failures for the CPU and I/O modules. There are two primary error classes: fatal and non fatal.

Fatal Errors

Fatal errors are errors the CPU has detected that offer a risk of the system not functioning safely or properly. If the CPU is in Run Mode when the fatal error occurs, the CPU will switch to Program Mode. (Remember, in Program Mode all outputs are turned off.) If the fatal error is detected while the CPU is in Program Mode, the CPU will not enter Run Mode until the error has been corrected. Here are some examples of fatal errors.

- Base power supply failure
- Parity error or CPU malfunction
- I/O configuration errors
- Certain programming errors

Non-fatal Errors

Non-fatal errors are errors that are flagged by the CPU as requiring attention. They can neither cause the CPU to change from Run Mode to Program Mode, nor do they prevent the CPU from entering Run Mode. The application program can special relays use to detect if a non-fatal error has occurred. The application program can then be used to take the system to an orderly shutdown or to switch the CPU to Program Mode if necessary.

Some examples of non-fatal errors are:

- Back-up battery voltage low
- All I/O module errors
- Certain programming errors

Finding Diagnostic Information

Diagnostic information can be found in several places with varying levels of message detail.

- The CPU automatically logs error codes and any FAULT messages into two separate tables which can be viewed with the Handheld Programmer or *DirectSOFT*.
- The Handheld Programmer displays error numbers and short descriptions of the error.
- *DirectSOFT* provides the error number and an error message.
- Appendix B in this manual has a complete list of error messages sorted by error number.

Many of these messages point to supplemental memory locations which can be referenced for additional related information. These memory references are in the form of V-memory and SPs (special relays).

The following two tables name the specific memory locations that correspond to certain types of error messages. The special relay table also includes status indicators which can be used in programming. For a more detailed description of each of these special relays, refer to Appendix D.

V-memory Locations Corresponding to Error Codes

| Error Class | Error Category | Diagnostic V-memory |
|------------------------------|---|---------------------|
| Battery Voltage (DL240 only) | Shows battery voltage to tenths (32 is 3.2V) | V7746 |
| User-Defined | Error code used with FAULT instruction | V7751 |
| I/O Configuration | Correct module ID code | V7752 |
| | Incorrect module ID code | V7753 |
| | Base and Slot number where error occurs | V7754 |
| System Error | Fatal Error code | V7755 |
| | Major Error code | V7756 |
| | Minor Error code | V7757 |
| Module Diagnostic | Base and slot number where error occurs | V7760 |
| | Always holds a "0" | V7761 |
| | Error code | V7762 |
| Grammatical | Address where syntax error occurs | V7763 |
| | Error code found during syntax check | V7764 |
| CPU Scan | Number of scans since last Program to Run Mode transition | V7765 |
| | Current scan time (ms) | V7775 |
| | Minimum scan time (ms) | V7776 |
| | Maximum scan time (ms) | V7777 |

Special Relays (SP) Corresponding to Error Codes

| Startup and Real-time Relays | |
|------------------------------|--|
| SP0 | On first scan only |
| SP1 | Always ON |
| SP2 | Always OFF |
| SP3 | 1 minute clock |
| SP4 | 1 second clock |
| SP5 | 100 millisecond clock |
| SP6 | 50 millisecond clock |
| SP7 | On alternate scans |
| CPU Status Relays | |
| SP11 | Forced Run mode (DL240/250-1/260) |
| SP12 | Terminal Run mode |
| SP13 | Test Run mode (DL240/250-1/260) |
| SP15 | Test program mode (DL240/250-1/260) |
| SP16 | Terminal Program mode |
| SP20 | STOP instruction was executed |
| SP22 | Interrupt enabled |
| System Monitoring Relays | |
| SP40 | Critical error |
| SP41 | Non-critical error |
| SP43 | Battery low |
| SP44 | Program memory error |
| SP45 | I/O error |
| SP46 | Communications error |
| SP47 | I/O configuration error |
| SP50 | Fault instruction was executed |
| SP51 | Watchdog timeout |
| SP52 | Syntax error |
| SP53 | Cannot solve the logic |
| SP54 | Intelligent module communication error |

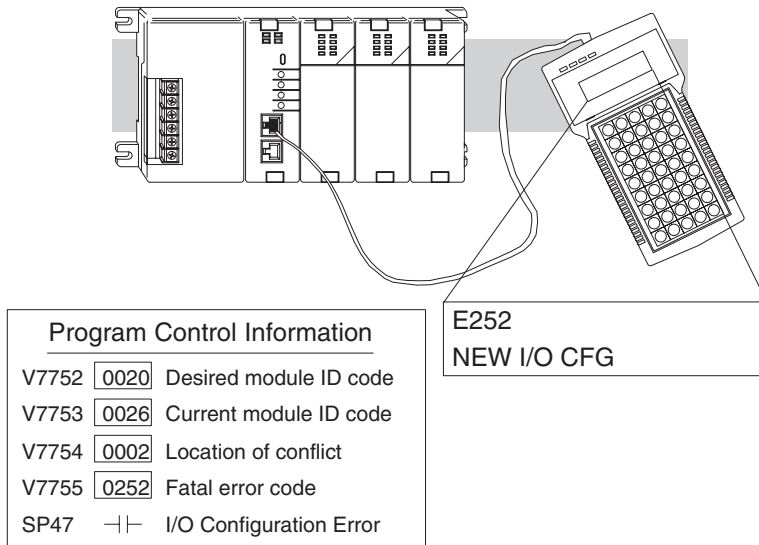
| Accumulator Status Relays | |
|---------------------------------|--|
| SP60 | Acc. is less than value |
| SP61 | Acc. is equal to value |
| SP62 | Acc. is greater than value |
| SP63 | Acc. result is zero |
| SP64 | Half borrow occurred |
| SP65 | Borrow occurred |
| SP66 | Half carry occurred |
| SP67 | Carry occurred |
| SP70 | Result is negative (sign) |
| SP71 | Pointer reference error |
| SP73 | Overflow |
| SP75 | Data is not in BCD |
| SP76 | Load zero |
| Communication Monitoring Relays | |
| SP116 (DL230/DL240) | CPU is communicating with another device |
| SP116 (DL250-1/DL260) | Port 2 is communicating with another device |
| SP117 | Communication error on Port 2 (DL250-1/DL260 only) |
| SP120 | Module busy, Slot 0 |
| SP121 | Communication error Slot 0 |
| SP122 | Module busy, Slot 1 |
| SP123 | Communication error Slot 1 |
| SP124 | Module busy, Slot 2 |
| SP125 | Communication error Slot 2 |
| SP126 | Module busy, Slot 3 |
| SP127 | Communication error Slot 3 |
| SP130 | Module busy, Slot 4 |
| SP131 | Communication error Slot 4 |
| SP132 | Module busy, Slot 5 |
| SP133 | Communication error Slot 5 |
| SP134 | Module busy, Slot 6 |
| SP135 | Communication error Slot 6 |
| SP136 | Module busy, Slot 7 |
| SP137 | Communication error Slot 7 |

I/O Module Codes

Each system component has a code identifier. This code identifier is used in some of the error messages related to the I/O modules. The following table shows these codes.

| Code (Hex) | Component Type | Code (Hex) | Component Type |
|------------|----------------------------|------------|---------------------|
| 04 | CPU | 36 | Analog Input |
| 03 | I/O Base | 2B | 16 pt. Input |
| 20 | 8 pt. Output | 37 | Analog Output |
| 21 | 8 pt. Input | 3D | Analog I/O Combo |
| 24 | 4 input/output combination | 4A | Counter Interface |
| 28 | 12 pt. Output | 7F | Abnormal |
| 28 | 16 pt. Output | FF | No module detected |
| 3F | 32 pt. Input | EE | D2-DCM H2-ECOM |
| 30 | 32 pt. Output | BE | F2-CP128 D2-RMSM |
| 52 | H2-ERM(100) | | |
| 51 | H2-CTRIO(2) | | |

The following diagram shows an example of how the I/O module codes are used:



Error Message Tables

- 230
- 240
- 250-1
- 260

The DL240/250-1/260 CPUs will automatically log any system error codes and any custom messages you have created in your application program with the FAULT instructions. The CPU logs the error code, the date, and the time the error occurred. Two separate tables store this information.

- Error Code Table – the system logs up to 32 errors in the table. When an error occurs, the errors already in the table are pushed down and the most recent error is loaded into the top position. If the table is full when an error occurs, the oldest error is pushed (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 position. If the table is full when an error occurs, the oldest message is pushed (erased) from the table.

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

| Date | Time | Message |
|------------|-------------|---------------------|
| 2008-05-26 | 08:41:51:11 | *Conveyor-2 stopped |
| 2008-04-30 | 17:01:11:56 | *Conveyor-1 stopped |
| 2008-04-30 | 17:01:11:12 | *Limit SW1 failed |
| 2008-04-28 | 03:25:14:31 | *Saw Jam Detect |

You can access the error code table and the message table through *DirectSOFT*'s PLC Diagnostic sub-menus or from the Handheld Programmer. Details on how to access these logs are provided in the *DirectSOFT* manual.

The following examples show you how to use the Handheld Programmer and AUX Function 5C to show the error codes. The most recent error or message is always displayed. You can use the PREV and NXT keys to scroll through the messages.

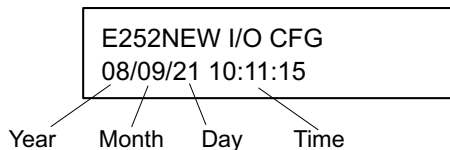
Use AUX 5C to view the tables



Use the arrow key to select Errors or Messages



Example of an error display



System Error Codes

- 230 The System error log contains 32 of the most recent errors that have been detected. The errors that are trapped in the error log are a subset of all the error messages which the DL205 systems generate. These errors can be generated by the CPU or by the Handheld Programmer, depending on the actual error. Appendix B provides more complete descriptions of the error codes.
- 240
- 250-1
- 260

The errors can be detected at various times. However, most of them are detected at power-up, on entry to Run Mode, or when a Handheld Programmer key sequence results in an error or an illegal request.

| Error Code | Description |
|------------|---|
| E003 | Software time-out |
| E004 | Invalid instruction (RAM parity error in the CPU) |
| E041 | CPU battery low |
| E043 | Memory cartridge battery low |
| E099 | Program memory exceeded |
| E101 | CPU memory cartridge missing |
| E104 | Write fail |
| E151 | Invalid command |
| E155 | RAM failure |
| E201 | Terminal block missing |
| E202 | Missing I/O module |
| E203 | Blown fuse |
| E206 | User 24V power supply failure |
| E210 | Power fault |
| E250 | Communication failure in the I/O chain |
| E251 | I/O parity error |
| E252 | New I/O configuration |
| E262 | I/O out of range |
| E312 | Communications error 2 |
| E313 | Communications error 3 |
| E316 | Communications error 6 |
| E320 | Time out |
| E321 | Communications error |
| E499 | Invalid Text entry for Print Instruction |
| E501 | Bad entry |
| E502 | Bad address |
| E503 | Bad command |
| E504 | Bad reference / value |
| E505 | Invalid instruction |

| Error Code | Description |
|------------|--------------------------------------|
| E506 | Invalid operation |
| E520 | Bad operation - CPU in Run |
| E521 | Bad operation - CPU in Test Run |
| E523 | Bad operation - CPU in Test Program |
| E524 | Bad operation - CPU in Program |
| E525 | Mode Switch not in TERM |
| E526 | Unit is offline |
| E527 | Unit is online |
| E528 | CPU mode |
| E540 | CPU locked |
| E541 | Wrong password |
| E542 | Password reset |
| E601 | Memory full |
| E602 | Instruction missing |
| E604 | Reference missing |
| E610 | Bad I/O type |
| E611 | Bad Communications ID |
| E620 | Out of memory |
| E621 | EEPROM Memory not blank |
| E622 | No Handheld Programmer EEPROM |
| E624 | V-memory only |
| E625 | Program only |
| E627 | Bad write operation |
| E628 | Memory type error (should be EEPROM) |
| E640 | Mis-compare |
| E650 | Handheld Programmer system error |
| E651 | Handheld Programmer ROM error |
| E652 | Handheld Programmer RAM error |

Program Error Codes

The following list shows the errors that can occur when there are problems with the program. These errors will be detected when you try to place the CPU into Run Mode or when you use AUX 21 – Check Program. The CPU will also turn on SP52 and store the error code in V7755. Appendix B provides more complete descriptions of the error codes.

| Error Code | Description |
|------------|-----------------------------|
| E4** | No Program in CPU |
| E401 | Missing END statement |
| E402 | Missing LBL |
| E403 | Missing RET |
| E404 | Missing FOR |
| E405 | Missing NEXT |
| E406 | Missing IRT |
| E412 | SBR/LBL >64 |
| E413 | FOR/NEXT >64 |
| E421 | Duplicate stage reference |
| E422 | Duplicate SBR/LBL reference |
| E423 | Nested loops |
| E431 | Invalid ISG/SG address |
| E432 | Invalid jump (GOTO) address |
| E433 | Invalid SBR address |
| E434 | Invalid RTC address |
| E435 | Invalid RT address |
| E436 | Invalid INT address |
| E437 | Invalid IRTC address |
| E438 | Invalid IRT address |
| E440 | Invalid Data address |
| E441 | ACON/NCON |
| E451 | Bad MLS/MLR |
| E452 | X input used as output coil |
| E453 | Missing T/C |
| E454 | Bad TMRA |
| E455 | Bad CNT |
| E456 | Bad SR |

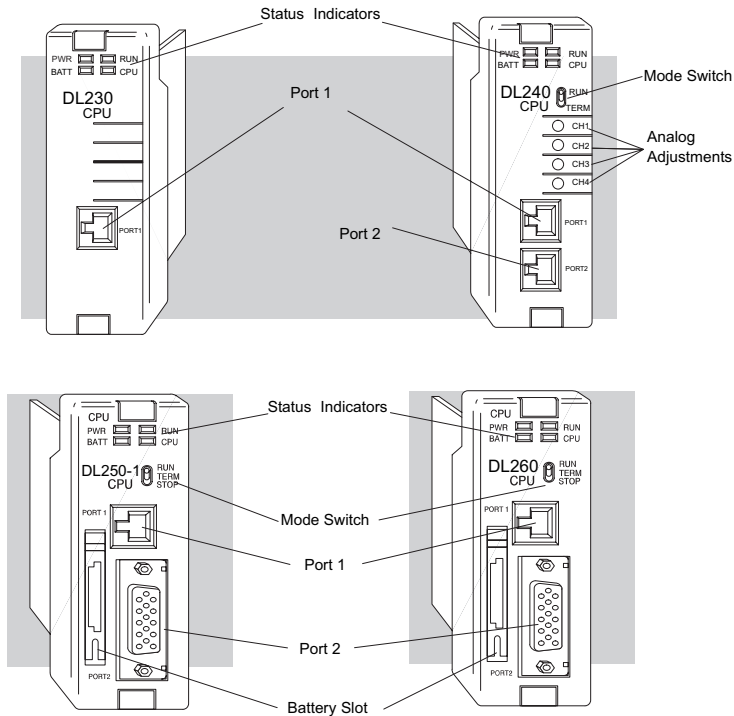
| Error Code | Description |
|------------|---------------------------|
| E461 | Stack Overflow |
| E462 | Stack Underflow |
| E463 | Logic Error |
| E464 | Missing Circuit |
| E471 | Duplicate coil reference |
| E472 | Duplicate TMR reference |
| E473 | Duplicate CNT reference |
| E480 | CV position error |
| E481 | CV not connected |
| E482 | CV exceeded |
| E483 | CVJMP placement error |
| E484 | No CV |
| E485 | No CVJMP |
| E486 | BCALL placement error |
| E487 | No Block defined |
| E488 | Block position error |
| E489 | Block CR identifier error |
| E490 | No Block stage |
| E491 | ISG position error |
| E492 | BEND position error |
| E493 | BEND I error |
| E494 | No BEND |

CPU Error Indicators

The DL205 CPUs have indicators on the front to help you diagnose problems with the system. The table below gives a quick reference of potential problems associated with each status indicator. Following the table will be a detailed analysis of each of these indicator problems.

| Indicator Status | Potential Problems |
|------------------------|---|
| PWR (off) | <ol style="list-style-type: none"> 1. System voltage incorrect 2. Power supply/CPU is faulty 3. Other component such as an I/O module has power supply shorted 4. Power budget exceeded for the base being used |
| RUN (will not come on) | <ol style="list-style-type: none"> 1. CPU programming error 2. Switch in TERM position 3. Switch in STOP position (DL250-1, DL260 only) |
| RUN (flashing) | Firmware upgrade mode |
| CPU (on) | <ol style="list-style-type: none"> 1. Electrical noise interference 2. CPU defective |
| BATT (on) | <ol style="list-style-type: none"> 1. CPU battery low 2. CPU battery missing, or disconnected |

9



PWR Indicator

There are four general reasons for the CPU power status LED (PWR) to be OFF:

- Power to the base is incorrect or is not applied.
- Base power supply is faulty.
- Other component(s) have the power supply shut down.
- Power budget for the base has been exceeded.

Incorrect Base Power

If the voltage to the power supply is not correct, the CPU and/or base may not operate properly or may not operate at all. Use the following guidelines to correct the problem.



WARNING: To minimize the risk of electrical shock, always disconnect the system power before inspecting the physical wiring.

1. First, disconnect the system power and check all incoming wiring for loose connections.
2. If you are using a separate termination panel, check those connections to make sure the wiring is connected to the proper location.
3. If the connections are acceptable, reconnect the system power and measure the voltage at the base terminal strip to ensure it is within specification. If the voltage is not correct, shut down the system and correct the problem.
4. If all wiring is connected correctly and the incoming power is within the specifications required, the base power supply should be returned for repair.

Faulty CPU

There is not a good check to test for a faulty CPU other than substituting a known good one to see if this corrects the problem. If you have experienced major power surges, it is possible the CPU and power supply have been damaged. If you suspect this is the cause of the power supply damage, a line conditioner that removes damaging voltage spikes should be used in the future.

Device or Module causing the Power Supply to Shutdown

It is possible a faulty module or external device using the system 5V can shut down the power supply. This 5V can be coming from the base or from the CPU communication ports.

To test for a device causing this problem:

1. Turn off power to the CPU.
2. Disconnect all external devices (i.e., communication cables) from the CPU.
3. Reapply power to the system.

If the power supply operates normally, you may have either a shorted device or a shorted cable. If the power supply does not operate normally, then test for a module causing the problem by following the steps below:

If the PWR LED operates normally, the problem could be in one of the modules. To isolate which module is causing the problem, disconnect the system power and remove one module at a time until the PWR LED operates normally. Follow the procedure below:

- Turn off power to the base.
- Remove a module from the base.
- Reapply power to the base.

Bent base connector pins on the module can cause this problem. Check to see the connector is not the problem.

9

Power Budget Exceeded

If the machine had been operating correctly for a considerable amount of time prior to the indicator going off, the power budget is not likely to be the problem. Power budgeting problems usually occur during system startup when the PLC is under operation and the inputs/outputs are requiring more current than the base power supply can provide.



WARNING: The PLC may reset if the power budget is exceeded. If there is any doubt about the system power budget, please check it at this time. Exceeding the power budget can cause unpredictable results which can cause damage and injury. Verify the modules in the base operate within the power budget for the chosen base. You can find these tables in Chapter 4, Bases and I/O Configuration.

Run Indicator

If the CPU will not enter the Run mode (the RUN indicator is off), the problem is usually in the application program, unless the CPU has a fatal error. If a fatal error has occurred, the CPU LED should be on. (You can use a programming device to determine the cause of the error.) If the RUN light is flashing, the PLC is in firmware upgrade mode.

If you are using a DL240, DL250-1 or DL260 and you are trying to change the modes with a programming device, make sure the mode switch is in the TERM position.

Both of the programming devices, Handheld Programmer and *DirectSOFT*, will return an error message describing the problem. Depending on the error, there may also be an AUX function you can use to help diagnose the problem. The most common programming error is “Missing END Statement.” All application programs require an END statement for proper termination. A complete list of error codes can be found in Appendix B.

CPU Indicator

If the CPU indicator is on, a fatal error has occurred in the CPU. Generally, this is not a programming problem but an actual hardware failure. You can power cycle the system to clear the error. If the error clears, you should monitor the system and determine what caused the problem. You will find this problem is sometimes caused by high frequency electrical noise introduced into the CPU from an outside source. Check your system grounding and install electrical noise filters if the grounding is suspected. If power cycling the system does not reset the error, or if the problem returns, you should replace the CPU.

BATT Indicator

If the BATT indicator is on, the CPU battery is either disconnected or needs replacing. The battery voltage is continuously monitored while the system voltage is being supplied.

Communications Problems

If you cannot establish communications with the CPU, check these items.

- The cable is disconnected.
- The cable has a broken wire or has been wired incorrectly.
- The cable is improperly terminated or grounded.
- The device connected is not operating at the correct baud rate (9600 baud for the top port. Use AUX 56 to select the baud rate for the bottom port on a DL240, DL250-1 and DL260).
- The device connected to the port is sending data incorrectly.
- A grounding difference exists between the two devices.
- Electrical noise is causing intermittent errors.
- The CPU has a bad communication port and the CPU should be replaced.

If an error occurs, the indicator will come on and stay on until a successful communication has been completed.

I/O Module Troubleshooting

Things to Check

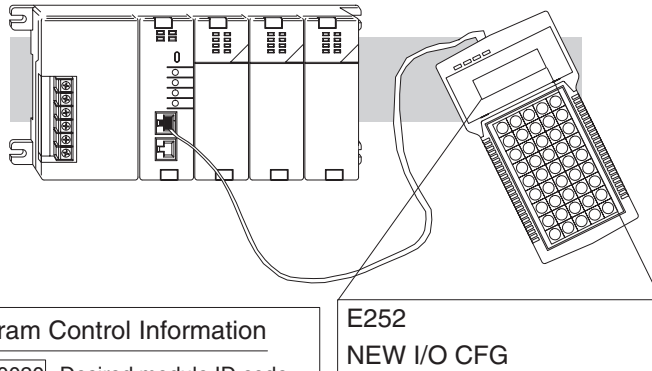
If you suspect an I/O error, there are several things that could be causing the problem.

- A blown fuse.
- A loose terminal block.
- The 24VDC supply has failed.
- The module has failed.
- The I/O configuration check detects a change in the I/O configuration.

I/O Diagnostics

If the modules are not providing any clues to the problem, run AUX 42 from the handheld programmer or I/O diagnostics in *DirectSOFT*. Both options will provide the base number, the slot number and the problem with the module. Once the problem is corrected, the indicators will reset.

An I/O error will not cause the CPU to switch from the run to program mode; however there are special relays (SPs) available in the CPU which will allow this error to be read in ladder logic. The application program can then take the required action, such as entering the program mode or initiating an orderly shutdown. The figure below shows an example of the failure indicators.



| 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 |

Some Quick Steps

When troubleshooting the DL205 series I/O modules, you should be aware of a few facts you that may assist you in quickly correcting an I/O problem:

- The output modules cannot detect shorted or open output points. If you suspect one or more points on an output module to be faulty, you should measure the voltage drop from the common to the suspect point. Remember, when using a Digital Volt Meter, leakage current from an output device, such as a triac or a transistor, must be considered. A point which is off may appear to be on if no load is connected to the point.
- The I/O point status indicators on the modules are logic side indicators. This means the LED which indicates the on or off status reflects the status of the point with respect to the CPU. On an output module, the status indicators could be operating normally, while the actual output device (transistor, triac etc) could be damaged. With an input module, if the indicator LED is on, the input circuitry should be operating properly. To verify proper functionality, check to see that the LED goes off when the input signal is removed.
- Leakage current can be a problem when connecting field devices to I/O modules. False input signals can be generated when the leakage current of an output device is great enough to turn on the connected input device. To correct this, install a resistor in parallel with the input or output of the circuit. The value of this resistor will depend on the amount of leakage current and the voltage applied but usually a 10K Ω to 20K Ω resistor will work. Ensure the wattage rating of the resistor is correct for your application.
- The easiest method to determine if a module has failed is to replace it, if you have a spare. However, if you suspect another device to have caused the failure in the module, that device may cause the same failure in the replacement module as well. As a point of caution, you may want to check devices or power supplies connected to the failed module before replacing it with a spare module.

Testing Output Points

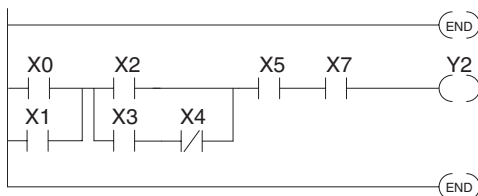
Output points can be set on or off in the DL205 series CPUs. In the DL240 and DL250-1, you can use AUX 59, Bit Override, to force a point even while the program is running. However, this is not a recommended method to test the output points. If you want to do an I/O check independent of the application program for either the DL230, DL240, DL250-1 or DL260, follow the procedure below:

| Step | Action |
|------|---|
| 1 | Use a Handheld Programmer or <i>DirectSOFT</i> to communicate online to the PLC. |
| 2 | Change to Program Mode. |
| 3 | Go to address 0. |
| 4 | Insert an "END" statement at address 0. (This will cause program execution to occur only at address 0 and prevent the application program from turning the I/O points on or off.) |
| 5 | Change to Run Mode. |
| 6 | Use the programming device to set (turn) on or off the points you wish to test. |
| 7 | When you finish testing I/O points, delete the "END" statement at address 0. |



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. Make sure you have taken all appropriate safety precautions prior to testing any I/O points.

Handheld Programmer Keystrokes Used to Test an Output Point



Insert an END statement at the beginning of the program. This disables the remainder of the program.

From a clear display, use the following keystrokes



```
16P STATUS
BIT REF X
```

Use the PREV or NEXT keys to select the Y data type



```
Y 10 Y 0
□□□□□□□□□□□□□□□□
```

Use arrow keys to select point, then use ON and OFF to change the status



Y2 is now on

```
Y 10 Y 0
□□□□□□□□□□□■□□
```


Noise Troubleshooting

Electrical Noise Problems

Noise is one of the most difficult problems to diagnose. Electrical noise can enter a system in many different ways and falls into one of two categories: conducted or radiated. It may be difficult to determine how the noise is entering the system, but the corrective actions for either type of noise problem are similar.

- Conducted noise is when the electrical interference is introduced into the system by way of an attached wire, panel connection, etc. It may enter through an I/O module, a power supply connection, the communication ground connection, or the chassis ground connection.
- Radiated noise is when the electrical interference is introduced into the system without a direct electrical connection, much in the same manner as radio waves.

Reducing Electrical Noise

While electrical noise cannot be eliminated, it can be reduced to a level that will not affect the system.

- Most noise problems result from improper grounding of the system. A good earth ground can be the single most effective way to correct noise problems. If a ground is not available, install a ground rod as close to the system as possible. Ensure all ground wires are single-point grounds and are not daisy chained from one device to another. Ground metal enclosures around the system. A loose wire is no more than a large antenna waiting to introduce noise into the system; therefore, you should tighten all connections in your system. Loose ground wires are more susceptible to noise than the other wires in your system. Review Chapter 2, Installation, Wiring, and Specifications, if you have questions regarding how to ground your system.
- Electrical noise can enter the system through the power source for the CPU and I/O. Installing an isolation transformer for all AC sources can correct this problem. DC power sources should be well grounded, good quality power supplies. Switching DC power supplies commonly generate more noise than linear supplies.
- Separate input wiring from output wiring. Never run I/O wiring close to high voltage wiring.

Machine Startup and Program Troubleshooting

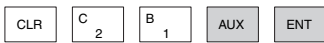
The DL205 CPUs provide several features to help you debug your program before and during machine startup. This section discusses the following topics which can be very helpful.

- Program Syntax Check
- Duplicate Reference Check
- Test Modes
- Special Instructions
- Run Time Edits
- Forcing I/O Points

Program Syntax Check


Even though the Handheld Programmer and *DirectSOFT* provide error checking during program entry, you may want to check a modified program. Both programming devices offer a way to check the program syntax. For example, you can use AUX 21, CHECK PROGRAM to check the program syntax from a Handheld Programmer, or you can use the PLC Diagnostics menu option within *DirectSOFT*. This check will find a wide variety of programming errors. The following example shows how to use the syntax check with a Handheld Programmer.

Use AUX 21 to perform syntax check



AUX 21 CHECK PRO
1:SYN 2:DUP REF

Select syntax check (default selection)

 (You may not get the busy display if the program is not very long.)

BUSY

One of two displays will appear

Error Display (example)

\$00050 E401
MISSING END
(shows location in question)

Syntax OK display

NO SYNTAX ERROR
?

See the Error Codes Section for a complete listing of programming error codes. If you get an error, press CLR and the Handheld Programmer will display the instruction where the error occurred. Correct the problem and continue running the Syntax check until the NO SYNTAX ERROR message appears.

Duplicate Reference Check

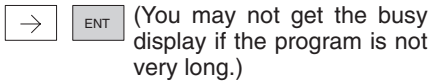
You can also check for multiple uses of the same output coil. Both programming devices offer a way to check for this condition. For example, you can AUX 21, CHECK PROGRAM to check for duplicate references from a Handheld Programmer, or you can use the PLC Diagnostics menu option within *DirectSOFT*. The following example shows how to perform the duplicate reference check with a Handheld Programmer.

Use AUX 21 to perform syntax check



AUX 21 CHECK PRO
1:SYN 2:DUP REF

Select duplicate reference check



BUSY

One of two displays will appear

Error Display (example)

\$00024 E471
DUP COIL REF

(shows location in question)

Syntax OK display

NO DUP REFS
?

If you get an error, press CLR and the Handheld Programmer will display the instruction where the error occurred. Correct the problem and continue running the Duplicate Reference check until no duplicate references are found.



NOTE: You can use the same coil in more than one location, especially in programs using the Stage instructions and/or the OROUT instructions. The Duplicate Reference check will find these outputs even though they may be used in an acceptable fashion.

TEST-PGM and TEST-RUN Modes

Test Mode allows the CPU to start in TEST-PGM mode, enter TEST-RUN mode, run a fixed number of scans, and return to TEST-PGM mode. You can select from 1 to 65,525 scans. Test Mode also allows you to maintain output status while you switch between Test-Program and Test-Run Modes. You can select Test Modes from either the Handheld Programmer (by using the MODE key) or from *DirectSOFT* via a PLC Modes menu option.

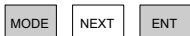
The primary benefit of using the TEST mode is to maintain certain outputs and other parameters when the CPU transitions back to Test-Program mode. For example, you can use AUX 58 from the DL205 Handheld Programmer to configure the individual outputs, CRs, etc., to hold their output state. Also, the CPU will maintain timer and counter current values when it switches to TEST-PGM mode.



NOTE: You can only use *DirectSOFT* to specify the number of scans. This feature is not supported on the Handheld Programmer. However, you can use the Handheld Programmer to switch between Test Program and Test Run Modes.

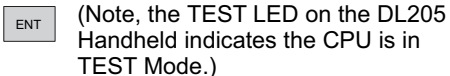
With the Handheld Programmer, the actual mode entered when you first select Test Mode depends on the mode of operation at the time you make the request. If the CPU is in Run Mode, then TEST-RUN is available. If the mode is Program, then TEST-PGM is available. Once you've selected TEST Mode, you can easily switch between TEST-RUN and TEST-PGM. *DirectSOFT* provides more flexibility in selecting the various modes with different menu options. The following example shows how you can use the Handheld Programmer to select the Test Modes.

Use the MODE key to select TEST Modes (example assumes Run Mode)



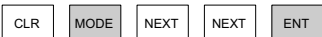
MODE CHANGE
GO TO T-RUN MODE

Press ENT to confirm TEST-RUN Mode



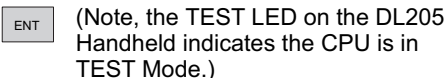
MODE CHANGE
CPU T-RUN

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



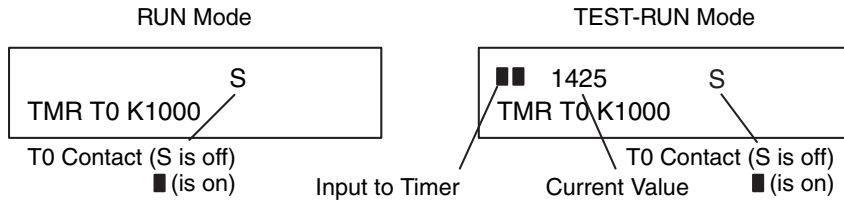
MODE CHANGE
GO TO T-PGM MODE

Press ENT to confirm TEST-PGM Mode

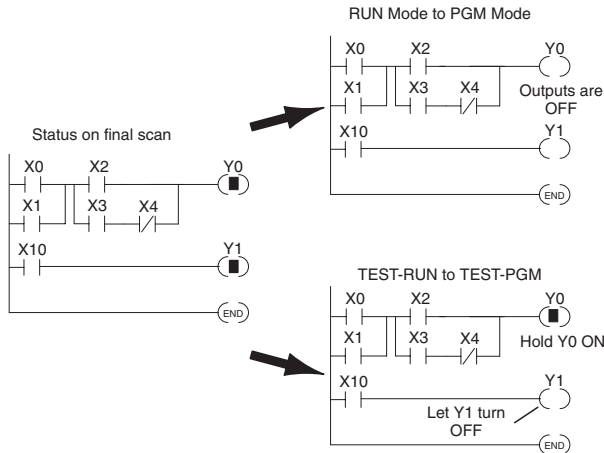


MODE CHANGE
CPU T-PGM

Test Displays: With the Handheld Programmer you also have a more detailed display when you use TEST Mode. For some instructions, the TEST-RUN mode display is more detailed than the status displays shown in RUN mode. The following diagram shows an example of a Timer instruction display during TEST-RUN mode.



Holding Output States: The ability to hold output states is very useful because it allows you to maintain key system I/O points. In some cases, you may need to modify the program, but you do not want certain operations to stop. In normal Run Mode, the outputs are turned off when you return to Program Mode. In TEST-RUN mode, you can set each individual output to either turn off, or to hold its last output state on the transition to TEST-PGM mode. You can use AUX 58 on the Handheld Programmer to select the action for each individual output. This feature is also available via a menu option within *DirectSOFT*. The following diagram shows the differences between RUN and TEST-RUN modes.



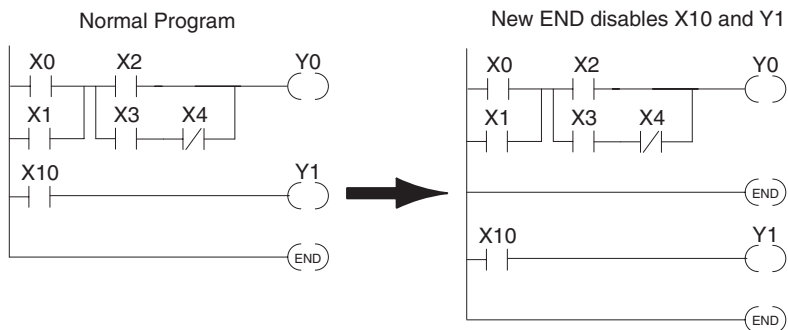
Before you decide that Test Mode is the perfect choice, remember that the DL205 CPUs also allow you to edit the program during Run Mode. The primary difference between the Test Modes and the Run Time Edit feature is you do not have to configure each individual I/O point to hold the output status. When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

Special Instructions

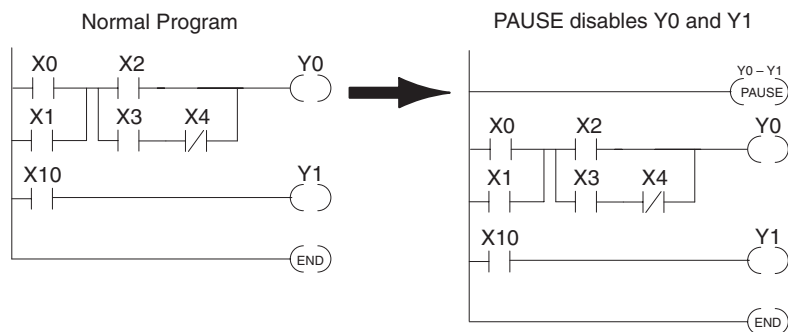
Several instructions can be used to help you debug your program during machine start-up operations.

- END
- PAUSE
- STOP

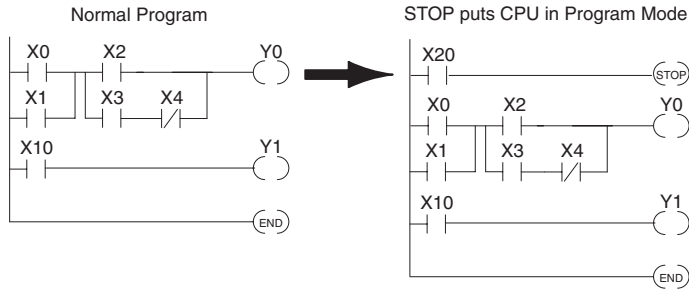
END Instruction: If you need a way to quickly disable part of the program, insert an END statement prior to the portion that should be disabled. When the CPU encounters the END statement, it assumes it is the end of the program. The following diagram shows an example.



PAUSE Instruction: This instruction provides a quick way to allow the inputs (or other logic) to operate while disabling selected outputs. The output image register is still updated, but the output status is not written to the modules. For example, you could make this conditional by adding an input contact or CR to control the instruction with a switch or a programming device. Or, you could add the instruction without any conditions so the selected outputs would be disabled at all times.



STOP Instruction: Sometimes during machine startup you need a way to quickly turn off all the outputs and return to Program Mode. In addition to using the Test Modes and AUX 58 (to configure each individual point), you can also use the STOP instruction. When this instruction is executed, the CPU automatically exits Run Mode and enters Program Mode. Remember, all outputs are turned off during Program Mode. The following diagram shows an example of a condition that returns the CPU to Program Mode.



In the example shown above, you could trigger X20, which would execute the STOP instruction. The CPU would enter Program Mode and all outputs would be turned off.

Run Time Edits

The DL205 CPUs allow you to make changes to the application program during Run Mode. These edits are not “bumpless.” Instead, CPU scan is momentarily interrupted (and the outputs are maintained in their current state) until the program change is complete. This means if the output is off, it will remain off until the program change is complete. If the output is on, it will remain on.



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

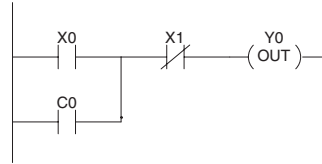
1. If there is a syntax error in the new instruction, the CPU will not enter the Run Mode.
2. If you delete an output coil reference and the output was on at the time, the output will remain on until it is forced off with a programming device.
3. Input point changes are not acknowledged during Run Time Edits. So, if you're using a high-speed operation and a critical input comes on, the CPU may not see the change.

Not all instructions can be edited during a Run Time Edit session. The following list shows the instructions that can be edited.

| Mnemonic | Description |
|-------------|--|
| TMR | Timer |
| TMRF | Fast timer |
| TMRA | Accumulating timer |
| TMRAF | Accumulating fast timer |
| CNT | Counter |
| UDC | Up / Down counter |
| SGCNT | Stage counter |
| STR, STRN | Store, Store not |
| AND, ANDN | And, And not |
| OR, ORN | Or, Or not |
| STRE, STRNE | Store equal, Store not equal |
| ANDE, ANDNE | And equal, And not equal |
| ORE, ORNE | Or equal, Or not equal |
| STR, STRN | Store greater than or equal Store less than |
| AND, ANDN | And greater than or equal And less than |

| Mnemonic | Description |
|----------|--|
| OR, ORN | Or greater than or equal, Or less than |
| LD | Load data (constant) |
| LDD | Load data double (constant) |
| ADDD | Add data double (constant) |
| SUBD | Subtract data double (constant) |
| MUL | Multiply (constant) |
| DIV | Divide (constant) |
| CMPD | Compare accumulator (constant) |
| ANDD | And accumulator (constant) |
| ORD | Or accumulator (constant) |
| XORD | Exclusive or accumulator (constant) |
| LDF | Load discrete points to accumulator |
| OUTF | Output accumulator to discrete points |
| SHFR | Shift accumulator right |
| SHFL | Shift accumulator left |
| NCON | Numeric constant |

Use the program logic shown to describe how this process works. In the example, change X0 to C10. Note, the example assumes you have already placed the CPU in Run Mode.



Use the MODE key to select Run Time Edits



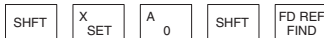
MODE CHANGE
RUN TIME EDIT?

Press ENT to confirm the Run Time Edits

ENT (Note, the RUN LED on the DL205 Handheld starts flashing to indicate Run Time Edits are enabled.)

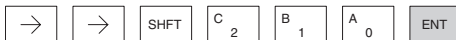
MODE CHANGE
RUNTIME EDITS

Find the instruction you want to change (X0)



\$00000 STR X0

Press the arrow key to move to the X. Then enter the new contact (C10).



RUNTIME EDIT?
STR C10

Press ENT to confirm the change

ENT (Note, once you press ENT, the next address is displayed.)

OR C0

Forcing I/O Points

There are many times, especially during machine startup and troubleshooting, where you need the capability to force an I/O point to be either on or off. Before you use a programming device to force any data type, it is important to understand how the DL205 CPUs process the forcing requests.



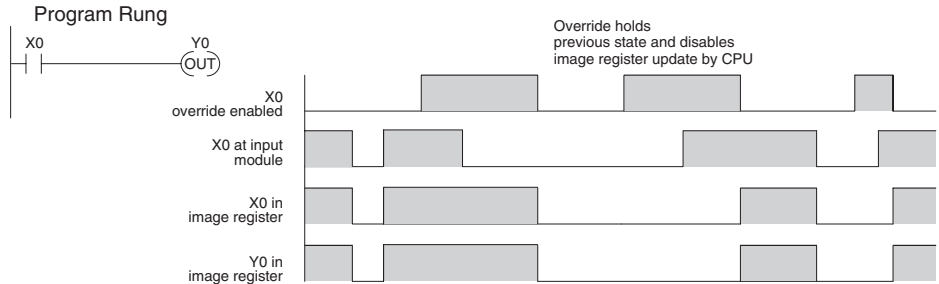
WARNING: Only authorized personnel fully familiar with all aspects of the application should make changes to the program. Make sure you thoroughly consider the impact of any changes to minimize the risk of personal injury or damage to equipment.

Two types of forcing are available with the DL205 CPUs. (Chapter 3 provides a detailed description of how the CPU processes each type of forcing request.)

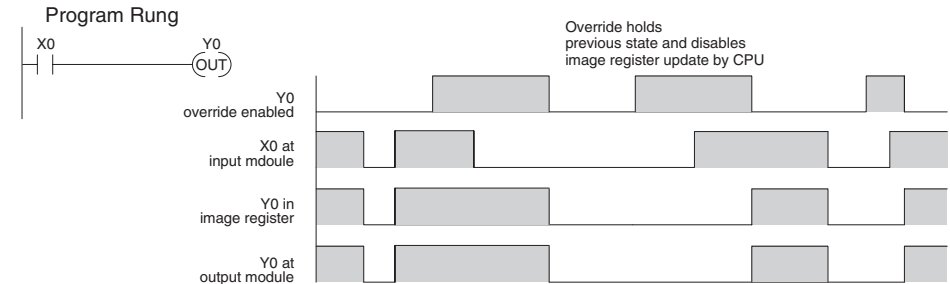
- **Regular Forcing** — This type of forcing can temporarily change the status of a discrete bit. For example, you may want to force an input on, even though it is really off. This allows you to change the point status that was stored in the image register. This value will be valid until the image register location is written to during the next scan. This is primarily useful during testing situations when you need to force a bit on to trigger another event.
- **Bit Override** — (DL240, DL250–1 or DL260) Bit override can be enabled on a point-by-point basis by using AUX 59 from the Handheld Programmer or by a menu option in *DirectSOFT*. 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 X1, and X1 is off at the time, the CPU will not change the state of X1. This means that even if X1 comes on, the CPU will not acknowledge the change. Therefore, if you used X1 in the program, it would always be evaluated as “off” in this case. If X1 was on when the bit override was enabled, then X1 would always be evaluated as “on.”

There is an advantage available when you use the bit override feature. The regular forcing is not disabled because the bit override is enabled. For example, if you enabled the Bit Override for Y0 and it was off at the time, the CPU would not change the state of Y0. However, you can still use a programming device to change the status. If you use the programming device to force Y0 on, it will remain on and the CPU will not change the state of Y0. If you then force Y0 off, the CPU will maintain Y0 as off. 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.

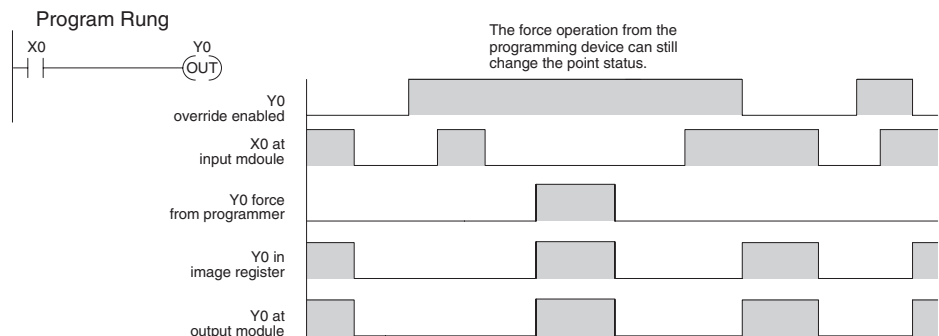
The following diagrams show how the bit override works for both input and output points. The example uses a simple rung, but the concepts are similar for any type of bit memory.



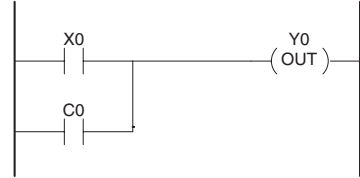
The following diagram shows how the bit override works for an output point. Notice the bit override maintains the output in the current state. If the output is on when the bit override is enabled, then the output stays on. If it is off, then the output stays off.



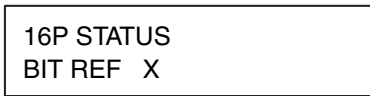
The following diagram shows how you can use a programming device in combination with the bit override to change the status of the point. Remember, bit override only disables CPU changes. You can still use a programming device to force the status of the point. Plus, since bit override maintains the current status, this enables true forcing. The example shown is for an output point, but you can also use the other bit data types.



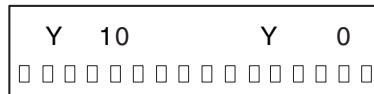
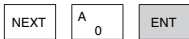
The following diagrams show a brief example of how you could use the DL205 Handheld Programmer to force an I/O point. Remember, if you are using the Bit Override feature, the CPU will retain the forced value until you disable the Bit Override or until you remove the force. The image register will not be updated with the status from the input module. Also, the solution from the application program will not be used to update the output image register. The example assumes you have already placed the CPU into Run Mode.



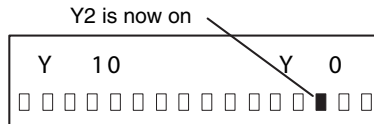
From a clear display, use the following keystrokes



Use the PREV or NEXT keys to select the Y data type. (Once the Y appears, press 0 to start at Y0.)

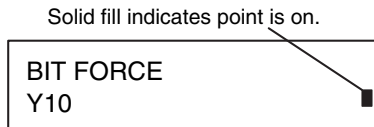


Use arrow keys to select point, then use ON and OFF to change the status

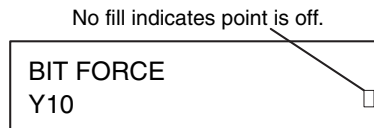


Regular Forcing with Direct Access

From a clear display, use the following keystrokes to force Y10 ON

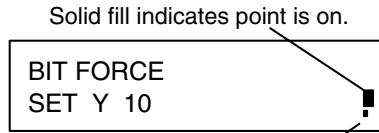


From a clear display, use the following keystrokes to force Y10 OFF



Bit Override Forcing

- ✗ 230 From a clear display, use the following keystrokes to turn on the override bit for Y10.
- ✓ 240
- ✓ 250-1
- ✓ 260

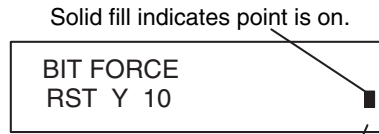


Small box indicates override bit is on.



NOTE: At this point you can use the PREV and NEXT keys to move to adjacent memory locations and use the SHFT ON keys to set the override bit on.

From a clear display, use the following keystrokes to turn off the override bit for Y10.



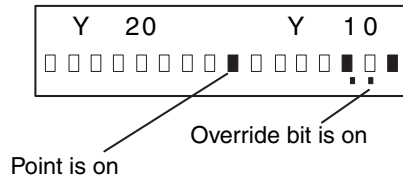
Small box is not present when override bit is off.

Like the example above, you can use the PREV and NEXT keys to move to adjacent memory locations and use the SHFT OFF keys to set the override bit off.

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.

From a clear display, use the following keystrokes to display the status of Y10 – Y20.



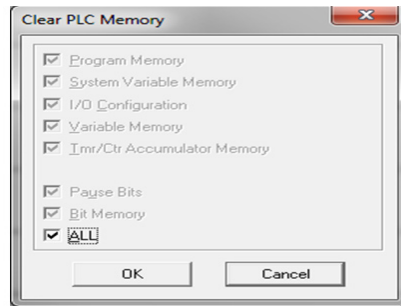
Reset the PLC to Factory Defaults



NOTE: Resetting to factory defaults will not clear any password stored in the PLC.

Resetting a DirectLogic PLC to Factory Defaults is a two-step process. Be sure to have a verified backup of your program using “Save Project to Disk” from the File menu before performing this procedure. Please be aware that the program as well as any settings will be erased and not all settings are stored in the project. In particular you will need to write down any settings for Secondary Communications Ports and manually set the ports up after resetting the PLC to factory defaults.

Step One – While connected to the PLC with DirectSoft, go to the PLC menu and select; “Clear PLC Memory”. Check the “ALL” box at the bottom of the list and press “OK”.



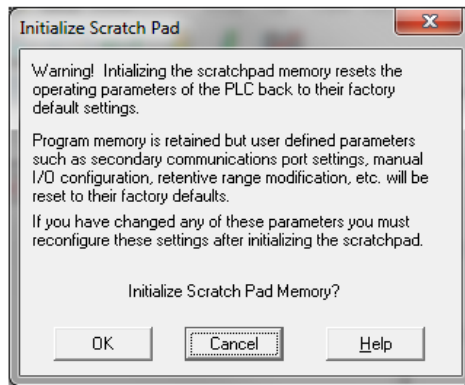
Step Two – While connected with DirectSoft, go the PLC menu and then to the “Setup” submenu and select; “Initialize Scratch Pad” and press “Ok”

NOTE: All configurable communications ports will be reset to factory default state. If you are connected via Port 2 or another configurable port, you may be disconnected when this operation is complete.



NOTE: Retentive ranges will be reset to the factory settings.

NOTE: Manually addressed IO will be reset to factory default settings.



The PLC has now been reset to factory defaults and you can proceed to program the PLC

Notes