ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Koyo Electronics / PLCDirect D2-09BDC-1* at our website: *Click HERE*

AUTOMATIONDIRECT.com

DL205 PLC User Manual

Volume 2 of 2

Manual Number: D2-USER-M



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

WARNING

Thank you for purchasing automation equipment from **Automationdirect.com®**, doing business as, **AutomationDirect**. We want your new 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.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). **AutomationDirect** specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.

For additional warranty and safety information, see the Terms and Conditions section of our catalog. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At **AutomationDirect** 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. **AutomationDirect** disclaims any proprietary interest in the marks and names of others.

Copyright 2010, Automationdirect.com Incorporated
All Rights Reserved

No part of this manual shall be copied, reproduced, or transmitted in any way without the prior, written consent of **Automationdirect.com Incorporated**. **AutomationDirect** retains the exclusive rights to all information included in this document.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AVERTISSEMENT

Nous vous remercions d'avoir acheté l'équipement d'automatisation de **Automationdirect.comMC**, en faisant des affaires comme, **AutomationDirect**. Nous tenons à ce que votre nouvel équipement d'automatisation fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux («activités à risque élevé»). La société **AutomationDirect** nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770-844-4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société **AutomationDirect**, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerce

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. **AutomationDirect** nie tout intérêt dans les autres marques et désignations.

Copyright 2010, Automationdirect.com Incorporated
Tous droits réservés

Nulle partie de ce manuel ne doit être copiée, reproduite ou transmise de quelque façon que ce soit sans le consentement préalable écrit de la société **Automationdirect.com Incorporated**. **AutomationDirect** conserve les droits exclusifs à l'égard de tous les renseignements contenus dans le présent document.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Volume One: Table of Contents



Volume One: Table of Contents i

Volume Two: Table of Contents xi

Chapter 1: Getting Started 1–1

Introduction 1–2
The Purpose of this Manual 1–2
Where to Begin 1–2
Supplemental Manuals 1–2
Technical Support 1–2

Conventions Used 1–3
Key Topics for Each Chapter 1–3

DL205 System Components 1–4
CPUs 1–4
Bases 1–4
I/O Configuration 1–4
I/O Modules 1–4
DL205 System Diagrams 1–5

Programming Methods 1–7
DirectSOFT Programming for Windows. 1–7
Handheld Programmer 1–7

***Direct*LOGIC™ Part Numbering System** 1–8

Quick Start for PLC Validation and Programming 1–10

Steps to Designing a Successful System 1–13

Chapter 2: Installation, Wiring and Specifications 2–1

Safety Guidelines 2–2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Plan for Safety 2–2
Three Levels of Protection 2–3
Emergency Stops 2–3
Emergency Power Disconnect 2–4
Orderly System Shutdown 2–4
Class 1, Division 2, Approval 2–4

Mounting Guidelines 2–5
Base Dimensions 2–5
Panel Mounting and Layout 2–6
Enclosures 2–7
Environmental Specifications 2–8
Power 2–8
Marine Use 2–9
Agency Approvals 2–9
24 VDC Power Bases 2–9

Installing DL205 Bases 2–10
Choosing the Base Type 2–10
Mounting the Base 2–10
Using Mounting Rails 2–11

Installing Components in the Base 2–12

Base Wiring Guidelines 2–13
Base Wiring 2–13

I/O Wiring Strategies 2–14
PLC Isolation Boundaries 2–14
Powering I/O Circuits with the Auxiliary Supply 2–15
Powering I/O Circuits Using Separate Supplies 2–16
Sinking / Sourcing Concepts 2–17
I/O “Common” Terminal Concepts 2–18
Connecting DC I/O to “Solid State” Field Devices 2–19
Solid State Input Sensors 2–19
Solid State Output Loads 2–19
Relay Output Guidelines 2–21
Surge Suppression For Inductive Loads 2–21

I/O Modules Position, Wiring, and Specification 2–25
Slot Numbering 2–25
Module Placement Restrictions 2–25

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Special Placement Considerations for Analog Modules 2–26
Discrete Input Module Status Indicators 2–26
Color Coding of I/O Modules 2–26
Wiring the Different Module Connectors 2–27
I/O Wiring Checklist 2–28
D2-08ND3, DC Input 2–29
D2-16ND3-2, DC Input 2–29
D2–32ND3, DC Input 2–30
D2–32ND3–2, DC Input 2–31
D2-08NA-1, AC Input 2–32
D2-08NA-2, AC Input 2–33
D2-16NA, AC Input 2–34
F2-08SIM, Input Simulator 2–34
D2-04TD1, DC Output 2–35
D2–08TD1, DC Output 2–36
D2–08TD2, DC Output 2–36
D2–16TD1–2, DC Output 2–37
D2–16TD2–2, DC Output 2–37
F2–16TD1(2)P, DC Output With Fault Protection 2–38
F2–16TD1P, DC Output With Fault Protection 2–39
F2–16TD2P, DC Output with Fault Protection 2–40
D2–32TD1, DC Output 2–41
D2–32TD2, DC Output 2–41
F2–08TA, AC Output 2–42
D2–08TA, AC Output 2–42
D2–12TA, AC Output 2–43
D2–04TRS, Relay Output 2–44
D2–08TR, Relay Output 2–45
F2–08TR, Relay Output 2–46
F2–08TRS, Relay Output 2–47

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D2–12TR, Relay Output 2–48
D2–08CDR 4 pt., DC Input / 4pt., Relay Output 2–49
Glossary of Specification Terms 2–50

Chapter 3: CPU Specifications and Operations 3–1

CPU Overview 3–2
General CPU Features 3–2
DL230 CPU Features 3–2
DL240 CPU Features 3–2
DL250–1 CPU Features 3–3
DL260 CPU Features 3–3
CPU General Specifications 3–4
CPU Base Electrical Specifications 3–5
CPU Hardware Setup 3–6
Communication Port Pinout Diagrams 3–6
Port 1 Specifications 3–7
Port 2 Specifications 3–8
Selecting the Program Storage Media 3–9
Built-in EEPROM 3–9
EEPROM Sizes 3–9
EEPROM Operations 3–9
Installing the CPU 3–10
Connecting the Programming Devices 3–10
CPU Setup Information 3–11
Status Indicators 3–12
Mode Switch Functions 3–12
Changing Modes in the DL205 PLC 3–13
Mode of Operation at Power-up 3–13
Using Battery Backup 3–14
DL230 and DL240 3–14
DL250-1 and DL260 3–14
Battery Backup 3–14
Auxiliary Functions 3–15
Clearing an Existing Program 3–16
Initializing System Memory 3–16

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setting the Clock and Calendar 3–16
Setting the CPU Network Address 3–17
Setting Retentive Memory Ranges 3–17
Using a Password 3–18
Setting the Analog Potentiometer Ranges 3–19

CPU Operation 3–21
CPU Operating System 3–21
Program Mode Operation 3–22
Run Mode Operation 3–22
Read Inputs 3–23
Read Inputs from Specialty and Remote I/O 3–23
Service Peripherals and Force I/O 3–23
CPU Bus Communication 3–24
Update Clock, Special Relays and Special Registers 3–24
Solve Application Program 3–25
Solve PID Loop Equations 3–25
Write Outputs 3–25
Write Outputs to Specialty and Remote I/O 3–26
Diagnostics 3–26

I/O Response Time 3–27
Is Timing Important for Your Application? 3–27
Normal Minimum I/O Response 3–27
Normal Maximum I/O Response 3–27
Improving Response Time 3–28

CPU Scan Time Considerations 3–29
Initialization Process 3–30
Reading Inputs 3–30
Reading Inputs from Specialty I/O 3–31
Service Peripherals 3–31
CPU Bus Communication 3–32
Update Clock/Calendar, Special Relays, Special Registers 3–32
Writing Outputs 3–32
Writing Outputs to Specialty I/O 3–33
Diagnostics 3–33
Application Program Execution 3–34

PLC Numbering Systems 3–35

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PLC Resources 3–35
V–Memory 3–36
Binary-Coded Decimal Numbers 3–36
Hexadecimal Numbers 3–36

Memory Map 3–37
Octal Numbering System 3–37
Discrete and Word Locations 3–37
V–Memory Locations for Discrete Memory Areas 3–37
Input Points (X Data Type) 3–38
Output Points (Y Data Type) 3–38
Control Relays (C Data Type) 3–38
Timers and Timer Status Bits (T Data type) 3–38
Timer Current Values (V Data Type) 3–39
Counters and Counter Status Bits (CT Data type) 3–39
Counter Current Values (V Data Type) 3–39
Word Memory (V Data Type) 3–39
Stages (S Data type) 3–40
Special Relays (SP Data Type) 3–40
Remote I/O Points (GX Data Type) 3–40

DL230 System V-memory 3–41

DL240 System V-memory 3–43

DL250–1 System V-memory (DL250 also) 3–46

DL260 System V-memory 3–49

DL205 Aliases 3–52

DL230 Memory Map 3–53

DL240 Memory Map 3–54

DL250–1 Memory Map (DL250 also) 3–55

DL260 Memory Map 3–56

X Input/Y Output Bit Map 3–57

Control Relay Bit Map 3–59

Stage Control/Status Bit Map 3–63

Timer and Counter Status Bit Maps 3–65

Remote I/O Bit Map 3–66

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 4: System Design and Configuration 4–1

DL205 System Design Strategies 4–2
I/O System Configurations 4–2
Networking Configurations 4–2

Module Placement 4–3
Slot Numbering 4–3
Module Placement Restrictions 4–3
Automatic I/O Configuration 4–4
Manual I/O Configuration 4–4
Removing a Manual Configuration 4–5
Power–On I/O Configuration Check 4–5
I/O Points Required for Each Module 4–6

Calculating the Power Budget 4–7
Managing your Power Resource 4–7
CPU Power Specifications 4–7
Module Power Requirements 4–7
Power Budget Calculation Example 4–9
Power Budget Calculation Worksheet 4–10

Local Expansion I/O 4–11
D2–CM Local Expansion Module 4–11
D2–EM Local Expansion Module 4–12
D2–EXCBL–1 Local Expansion Cable 4–12
DL260 Local Expansion System 4–13
DL250–1 Local Expansion System 4–14
Expansion Base Output Hold Option 4–15
Enabling I/O Configuration Check using ***Direct***SOFT 4–16

Expanding DL205 I/O 4–17
I/O Expansion Overview 4–17
Ethernet Remote Master, H2-ERM(-F) 4–17
Ethernet Remote Master Hardware Configuration 4–18
Installing the ERM Module 4–19
Ethernet Base Controller, H2-EBC(100)(-F) 4–22
Install the EBC Module 4–23
Set the Module ID 4–23
Insert the EBC Module 4–23
Network Cabling 4–24

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10BaseFL Network Cabling 4–25
Maximum Cable Length 4–25
Add a Serial Remote I/O Master/Slave Module 4–26
Configuring the CPU's Remote I/O Channel 4–27
Configure Remote I/O Slaves 4–29
Configuring the Remote I/O Table 4–29
Remote I/O Setup Program 4–30
Remote I/O Test Program 4–31

Network Connections to Modbus and *Direct*Net 4–32
Configuring Port 2 For ***Direct***Net 4–32
Configuring Port 2 For Modbus RTU 4–32
Modbus Port Configuration 4–33
DirectNET Port Configuration 4–34

Network Slave Operation 4–35
Modbus Function Codes Supported 4–35
Determining the Modbus Address 4–35
If Your Host Software Requires the Data Type and Address 4–35
If Your Modbus Host Software Requires an Address ONLY 4–38
Example 1: V2100 584/984 Mode 4–40
Example 2: Y20 584/984 Mode 4–40
Example 3: T10 Current Value 484 Mode 4–40
Example 4: C54 584/984 Mode 4–40
Determining the ***Direct***NET Address 4–40
Network Master Operation 4–41
Communications from a Ladder Program 4–44
Multiple Read and Write Interlocks 4–44

Network Modbus RTU Master Operation (DL260 only) 4–45
Modbus Function Codes Supported 4–45
Modbus Port Configuration 4–46
RS–485 Network (Modbus only) 4–47
RS–232 Network 4–47
Modbus Read from Network (MRX) 4–48
MRX Slave Memory Address 4–49
MRX Master Memory Addresses 4–49
MRX Number of Elements 4–49
MRX Exception Response Buffer 4–49
Modbus Write to Network (MWX) 4–50

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MWX Slave Memory Address 4–51
MWX Master Memory Addresses 4–51
MWX Number of Elements 4–51
MWX Exception Response Buffer 4–51
MRX/MWX Example in ***Direct***SOFT 4–52
Multiple Read and Write Interlocks 4–52

Non–Sequence Protocol (ASCII In/Out and PRINT) 4–54
Configure the DL260 Port 2 for Non-Sequence 4–54
RS–485 Network 4–55
RS–232 Network 4–55
Configure the DL250-1 Port 2 for Non-Sequence 4–56
RS–422 Network 4–57
RS–232 Network 4–57

Chapter 5: RLL and Intelligent Box (IBOX) Instructions 5–1

Introduction 5–2

Using Boolean Instructions 5–5
END Statement 5–5
Simple Rungs 5–5
Normally Closed Contact 5–6
Contacts in Series 5–6
Midline Outputs 5–6
Parallel Elements 5–7
Joining Series Branches in Parallel 5–7
Joining Parallel Branches in Series 5–7
Combination Networks 5–7
Comparative Boolean 5–8
Boolean Stack 5–8
Immediate Boolean 5–9

Boolean Instructions 5–10

Comparative Boolean 5–27

Immediate Instructions 5–33

Timer, Counter and Shift Register Instructions 5–41
Using Timers 5–41
Timer Example Using Discrete Status Bits 5–43

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Timer Example Using Comparative Contacts 5–43
Accumulating Timer (TMRA) 5–44
Accumulating Timer Example using Discrete Status Bits 5–45
Accumulator Timer Example Using Comparative Contacts 5–45
Counter Example Using Discrete Status Bits 5–47
Counter Example Using Comparative Contacts 5–47
Stage Counter Example Using Discrete Status Bits 5–49
Stage Counter Example Using Comparative Contacts 5–49
Up/Down Counter Example Using Discrete Status Bits 5–51
Up/Down Counter Example Using Comparative Contacts 5–51

Accumulator/Stack Load and Output Data Instructions 5–53

Logical Instructions (Accumulator) 5–71

Math Instructions 5–88

Transcendental Functions (DL260 only) 5–121

Bit Operation Instructions 5–123

Number Conversion Instructions (Accumulator) 5–130

Table Instructions 5–144

Clock/Calendar Instructions 5–175

CPU Control Instructions 5–177

Program Control Instructions 5–179

Interrupt Instructions 5–187

Intelligent I/O Instructions 5–191

Network Instructions 5–193

Message Instructions 5–197

Modbus RTU Instructions (DL260) 5–205
Modbus Read from Network (MRX) 5–205
Modbus Write to Network (MWX) 5–208

ASCII Instructions (DL260) 5–211

Intelligent Box (IBox) Instructions (DL250-1/DL260) 5-230

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Volume Two: Table of Contents

Chapter 6: Drum Instruction Programming (DL250-1/DL260 only) .6–1

Introduction 6–2
Purpose 6–2
Drum Terminology 6–2
Drum Chart Representation 6–3
Output Sequences 6–3

Step Transitions 6–4
Drum Instruction Types 6–4
Timer-Only Transitions 6–4
Timer and Event Transitions 6–5
Event-Only Transitions 6–6
Counter Assignments 6–6
Last Step Completion 6–7

Overview of Drum Operation 6–8
Drum Instruction Block Diagram 6–8
Powerup State of Drum Registers 6–9

Drum Control Techniques 6–10
Drum Control Inputs 6–10
Self-Resetting Drum 6–11
Initializing Drum Outputs 6–11
Using Complex Event Step Transitions 6–11

Drum Instruction 6–12
Timed Drum with Discrete Outputs (DRUM) 6–12
Event Drum (EDRUM) 6–14
Handheld Programmer Drum Mnemonics 6–16

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Masked Event Drum with Discrete Outputs (MDRMD) 6–19
Masked Event Drum with Word Output (MDRMW) 6–21

Chapter 7: RLL*PLUS* Stage Programming 7–1

Introduction to Stage Programming 7–2
Overcoming "Stage Fright" 7–2

Learning to Draw State Transition Diagrams 7–3
Introduction to Process States 7–3
The Need for State Diagrams 7–3
A 2–State Process 7–3
RLL Equivalent 7–4
Stage Equivalent 7–4
Let's Compare 7–5
Initial Stages 7–5
What Stage Bits Do 7–6
Stage Instruction Characteristics 7–6

Using the Stage Jump Instruction for State Transitions 7–7
Stage Jump, Set, and Reset Instructions 7–7

Stage Program Example: Toggle On/Off Lamp Controller 7–8
A 4–State Process 7–8

Four Steps to Writing a Stage Program 7–9

Stage Program Example: A Garage Door Opener 7–10
Garage Door Opener Example 7–10
Draw the Block Diagram 7–10
Draw the State Diagram 7–11
Add Safety Light Feature 7–12
Modify the Block Diagram and State Diagram 7–12
Using a Timer Inside a Stage 7–13
Add Emergency Stop Feature 7–14
Exclusive Transitions 7–14

Stage Program Design Considerations 7–15
Stage Program Organization 7–15
How Instructions Work Inside Stages 7–16
Using a Stage as a Supervisory Process 7–17
Stage Counter 7–17

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Unconditional Outputs 7–18
Power Flow Transition Technique 7–18

Parallel Processing Concepts 7–19
Parallel Processes 7–19
Converging Processes 7–19
Convergence Stages (CV) 7–19
Convergence Jump (CVJMP) 7–20
Convergence Stage Guidelines 7–20

Managing Large Programs 7–21
Stage Blocks (BLK, BEND) 7–21
Block Call (BCALL) 7–22

RLL*PLUS* (Stage) Instructions 7–23
Stage (SG) 7–23
Initial Stage (ISG) 7–24
Jump (JMP) 7–24
Not Jump (NJMP) 7–24
Converge Stage (CV) and Converge Jump (CVJMP) 7–25
Block Call (BCALL) 7–27
Block (BLK) 7–27
Block End (BEND) 7–27
Stage View in ***Direct***SOFT 7–28

Questions and Answers about Stage Programming 7–29

Chapter 8: PID Loop Operation 8–1

DL250-1 and DL260 PID Loop Features 8–2
Main Features 8–2

Introduction to PID Control 8–4
Why use PID Control? 8–4

Introducing DL205 PID Control 8–6
Process Control Definitions 8–8

PID Loop Operation 8–9
Position Form of the PID Equation 8–9
Reset Windup Protection 8–10
Freeze Bias 8–11
Adjusting the Bias 8–11

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step Bias Proportional to Step Change in SP 8–12
Eliminating Proportional, Integral or Derivative Action 8–12
Velocity Form of the PID Equation 8–12
Bumpless Transfer 8–13
Loop Alarms 8–13
Loop Operating Modes 8–14
Special Loop Calculations 8–14

Ten Steps to Successful Process Control 8–16

PID Loop Setup 8–18
Some Things to Do and Know Before Starting 8–18
PID Error Flags 8–18
Establishing the Loop Table Size and Location 8–18
Loop Table Word Definitions 8–20
PID Mode Setting 1 Bit Descriptions (Addr + 00) 8–21
PID Mode Setting 2 Bit Descriptions (Addr + 01) 8–22
Mode/Alarm Monitoring Word (Addr + 06) 8–23
Ramp/Soak Table Flags (Addr + 33) 8–23
Ramp/Soak Table Location (Addr + 34) 8–24
Ramp/Soak Table Programming Error Flags (Addr + 35) 8–24
PV Auto Transfer (Addr + 36) from I/O Module Base/Slot/Channel Option 8–25
PV Auto Transfer (Addr + 36) from V-memory Option 8–25
Control Output Auto Transfer (Addr + 37) 8–25
Configure the PID Loop 8–26

PID Loop Tuning 8–41
Open-Loop Test 8–41
Manual Tuning Procedure 8–42
Alternative Manual Tuning Procedures by Others 8–44
Tuning PID Controllers 8–44
Auto Tuning Procedure 8–45
Use ***Direct***SOFT Data View with PID View 8–49
Open a New Data View Window 8–49
Open PID View 8–50

Using the Special PID Features 8–52
How to Change Loop Modes 8–52
Operator Panel Control of PID Modes 8–53
PLC Modes Effect on Loop Modes 8–53

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Loop Mode Override 8–53
PV Analog Filter 8–54
Creating an Analog Filter in Ladder Logic 8–55
Use the *Direct*SOFT 5 Filter Intelligent Box (IBOX) Instruction 8–56
FilterB Example 8–56

Ramp/Soak Generator 8–57
Introduction 8–57
Ramp/Soak Table 8–58
Ramp/Soak Table Flags 8–60
Ramp/Soak Generator Enable 8–60
Ramp/Soak Controls 8–60
Ramp/Soak Profile Monitoring 8–61
Ramp/Soak Programming Errors 8–61
Testing Your Ramp/Soak Profile 8–61

***Direct*SOFT Ramp/Soak Example 8–62**
Setup the Profile in PID Setup 8–62
Program the Ramp/Soak Control in Relay Ladder 8–62
Test the Profile 8–63

Cascade Control 8–64
Introduction 8–64
Cascaded Loops in the DL205 CPU 8–65
Tuning Cascaded Loops 8–66

Time-Proportioning Control 8–67
On/Off Control Program Example 8–68

Feedforward Control 8–69
Feedforward Example 8–70

PID Example Program 8–71
Program Setup for the PID Loop 8–71

Troubleshooting Tips 8–74

Glossary of PID Loop Terminology 8–76

Bibliography 8–78

Chapter 9: Maintenance and Troubleshooting 9–1

Hardware Maintenance 9–2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Standard Maintenance 9–2
Air Quality Maintenance 9–2
Low Battery Indicator 9–2
CPU Battery Replacement 9–2

Diagnostics 9–3
Diagnostics 9–3
Fatal Errors 9–3
Non-fatal Errors 9–3
Finding Diagnostic Information 9–4
V-memory Locations Corresponding to Error Codes 9–4
Special Relays (SP) Corresponding to Error Codes 9–5
I/O Module Codes 9–6
Error Message Tables 9–7
System Error Codes 9–8
Program Error Codes 9–9

CPU Error Indicators 9–10

PWR Indicator 9–11
Incorrect Base Power 9–11
Faulty CPU 9–11
Device or Module causing the Power Supply to Shutdown 9–12
Power Budget Exceeded 9–12
Run Indicator 9–13
CPU Indicator 9–13
BATT Indicator 9–13

Communications Problems 9–13

I/O Module Troubleshooting 9–14
Things to Check 9–14
I/O Diagnostics 9–14
Some Quick Steps 9–15
Testing Output Points 9–16
Handheld Programmer Keystrokes Used to Test an Output Point 9–16

Noise Troubleshooting 9–17
Electrical Noise Problems 9–17
Reducing Electrical Noise 9–17

Machine Startup and Program Troubleshooting 9–18

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Syntax Check 9–18
Duplicate Reference Check 9–19
TEST-PGM and TEST-RUN Modes 9–20
Special Instructions 9–22
Run Time Edits 9–24
Forcing I/O Points 9–26
Regular Forcing with Direct Access 9–28
Bit Override Forcing 9–29
Bit Override Indicators 9–29

Appendix A: Auxiliary Functions A–1

Introduction A–2
What are Auxiliary Functions? A–2
Accessing AUX Functions via ***Direct***SOFT A–3
Accessing AUX Functions via the Handheld Programmer A–3

AUX 2* — RLL Operations A–4
AUX 21-24 . . . . . .A–4
AUX 21 Check Program A–4
AUX 22 Change Reference A–4
AUX 23 Clear Ladder Range A–4
AUX 24 Clear Ladders A–4

AUX 3* — V-memory Operations A–5
AUX 31 . . . . . .A–5
AUX 31 Clear V-Memory A–5
AUX 4* — I/O Configuration A–5
AUX 41-46 . . . . . .A–5
AUX 41 Show I/O Configuration A–5
AUX 42 I/O Diagnostics A–5
AUX 44 Power-up Configuration Check A–5
AUX 45 Select Configuration A–6
AUX 46 to I/O Configuration A–6

AUX 5* — CPU Configuration A–7
AUX 51-5C A–7
AUX 51 Modify Program Name A–7
AUX 52 Display/Change Calendar A–7
AUX 53 Display Scan Time A–8

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 54 Initialize Scratchpad A–8
AUX 55 Set Watchdog Timer A–8
AUX 56 CPU Network Address A–8
AUX 57 Set Retentive Ranges A–9
AUX 58 Test Operations A–9
AUX 59 Bit Override A–10
AUX 5B Counter Interface Configuration A–10
AUX 5C Display Error History A–11

AUX 6* — Handheld Programmer Configuration A–12
AUX 61, 62 and 65 . . . . . A–12
AUX 61 Show Revision Numbers A–12
AUX 62 Beeper On/Off A–12
AUX 65 Run Self Diagnostics A–12

AUX 7* - EEPROM Operations A–12
AUX 71 - 76 . . . . . A–12
Transferrable Memory Areas A–13
AUX 71 CPU to HPP EEPROM A–13
AUX 72 HPP EEPROM to CPU A–13
AUX 73 Compare HPP EEPROM to CPU A–13
AUX 74 HPP EEPROM Blank Check A–13
AUX 75 Erase HPP EEPROM A–13
AUX 76 Show EEPROM Type A–13

AUX 8* — Password Operations A–14
AUX 81 - 83 . . . . . A–14
AUX 81 Modify Password A–14
AUX 82 Unlock CPU A–14
AUX 83 Lock CPU A–14

Appendix B: DL205 Error Codes B–1

Appendix C: Instruction Execution Times C–1

Introduction C–2
V-Memory Data Registers C–2
V-Memory Bit Registers C–2
How to Read the Tables C–2

Boolean Instructions C–3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions C–4
Bit of Word Boolean Instructions C–13
Immediate Instructions C–14
Timer, Counter and Shift Register Instructions C–15
Accumulator Data Instructions C–16
Logical Instructions C–18
Math Instructions C–20
Differential Instructions C–23
Bit Instructions C–24
Number Conversion Instructions C–25
Table Instructions C–25
CPU Control Instructions C–27
Program Control Instructions C–27
Interrupt Instructions C–28
Network Instructions C–28
Intelligent I/O Instructions C–28
Message Instructions C–29
RLL*PLUS* Instructions C–29
DRUM Instructions C–29
Clock / Calender Instructions C–30
Modbus Instructions C–30
ASCII Instructions C–30

Appendix D: Special Relays D–1

DL230 CPU Special Relays D–2
Startup and Real-Time Relays D–2
CPU Status Relays D–2
System Monitoring D–2
Accumulator Status D–3
Counter Interface Module Relays D–3
Equal Relays for Multi-step Presets with Up/Down Counter #1 / DL230 (for use with a Counter Interface Module) D–4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL240/DL250-1/DL260 CPU Special Relays D–5
Startup and Real-Time Relays D–5
CPU Status Relays D–5
System Monitoring Relays D–6
Accumulator Status Relays D–6
Counter Interface Module Relays D–7
Communications Monitoring Relays D–8
Equal Relays for Multi-step Presets with Up/Down Counter #1 (for use with a Counter Interface Module) D–9
Equal Relays for Multi-step Presets with Up/Down Counter #2 (for use with a Counter Interface Module) D–10

Appendix E: PLC Memory E-1

DL205 PLC Memory E-2
Non-volatile V-memory in the DL205 . . . . . E-3

Appendix F: DL205 Product Weight Table F-1

DL205 Product Weight Table F-2

Appendix G: ASCII Table G-1

ASCII Conversion Table G-2

Appendix H: Numbering Systems H–1

Introduction H–2
Binary Numbering System H–2
Hexadecimal Numbering System H–3
Octal Numbering System H–4
Binary Coded Decimal (BCD) Numbering System H–5
Real (Floating Point) Numbering System H–5
BCD/Binary/Decimal/Hex/Octal -What is the Difference? H–6
Data Type Mismatch H–7
Signed vs. Unsigned Integers H–8
AutomationDirect.com Products and Data Types H–9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

*Direct*LOGIC PLCs H–9
C-more/C-more Micro-Graphic Panels H–9

Appendix I: European Union Directives (CE) I-1

European Union (EU) Directives I-2
Member Countries I-2
Applicable Directives I-2
Compliance I-2
General Safety I-3
Special Installation Manual I-4
Other Sources of Information I-4

Basic EMC Installation Guidelines I-4
Enclosures I-4
AC Mains Filters I-5
Suppression and Fusing I-5
Internal Enclosure Grounding I-5
Equi–potential Grounding I-6
Communications and Shielded Cables I-6
Analog and RS232 Cables I-7
Shielded Cables within Enclosures I-7
Analog Modules and RF Interference I-8
Network Isolation I-8
DC Powered Versions I-8
Items Specific to the DL205 . . . . . I-9

Index

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum Instruction Programming (DL250-1/DL260 only)



In This Chapter...

Introduction 6–2
Step Transitions 6–4
Overview of Drum Operation 6–8
Drum Control Techniques 6–10
Drum Instruction 6–12

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

Purpose

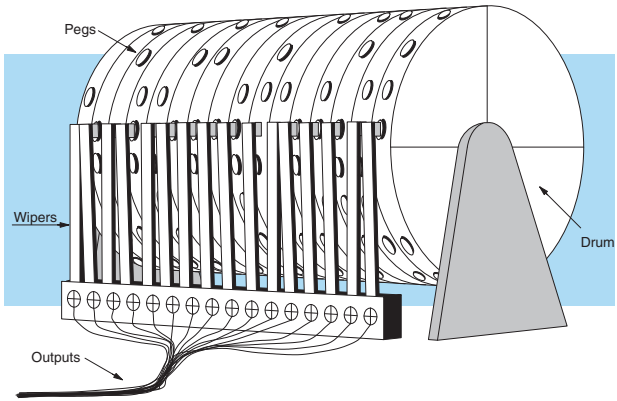
The four types of drum instructions available in the DL250-1 and DL260 CPUs electronically simulate an electro-mechanical drum sequencer. The instructions offer slight variations on the basic principle.

Drum Terminology

Drum instructions are best suited for repetitive processes that consist of a finite number of steps. They can do the work of many rungs of ladder logic with elegant simplicity. Therefore, drums can save a lot of programming and debugging time.

We introduce some terminology associated with the **drum** instruction by describing the original mechanical drum shown below. The mechanical drum generally has pegs on its curved surface. The pegs are populated in a particular **pattern**, representing a set of desired actions for machine control. A motor or solenoid rotates the drum a precise amount at specific times. During rotation, stationary wipers sense the presence of pegs (present = on, absent = off). This interaction makes or breaks electrical contact with the wipers, creating electrical **outputs** from the drum. The outputs are wired to devices on a machine for On/Off control.

Drums usually have a finite number of positions within one rotation, called **steps**. Each step represents some process step. At powerup, the drum **resets** to a particular step. The drum rotates from one step to the next based on a **timer**, or on some external **event**. During special conditions, a machine operator can manually increment the drum step using a **jog** control on the drum's drive mechanism. The contact closure of each wiper generates a unique on/off pattern called a **sequence**, designed for controlling a specific machine. Because the drum is circular, it automatically repeats the sequence once per rotation. Applications vary greatly, and a particular drum may rotate once per second, or as slowly as once per week.



Electronic drums provide the benefits of mechanical drums and more. For example, they have a **preset** feature that is impossible for mechanical drums: The preset function lets you move from the present step directly to any other step on command!

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

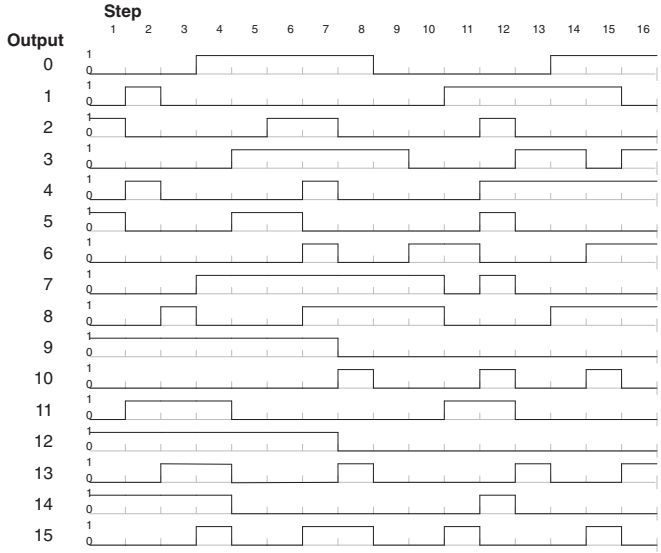
Drum Chart Representation

For editing purposes, the electronic drum is presented in chart form in *Direct*SOFT and in this manual. Imagine slicing the surface of a hollow drum cylinder between two rows of pegs, then pressing it flat. Now you can view the drum as a chart as shown below. Each row represents a step, numbered 1 through 16. Each column represents an output, numbered 0 through 15 (to match word bit numbering). The solid circles in the chart represent pegs (On state) in the mechanical drum, and the open circles are empty peg sites (Off state).

	OUTPUTS															
STEP	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	○	●	○	●	○	○	●	○	○	○	●	○	○	●	○	○
2	○	●	○	●	●	○	●	○	○	○	○	●	○	○	●	○
3	○	●	●	●	●	○	●	●	○	○	○	○	○	○	○	○
4	●	●	●	●	●	○	●	○	●	○	○	○	○	○	○	●
5	○	○	○	●	○	○	●	○	●	○	●	○	●	○	○	●
6	○	○	○	●	○	○	●	○	●	○	●	○	●	●	○	●
7	●	●	○	●	○	○	●	●	●	●	○	●	●	●	○	●
8	●	○	●	○	○	●	○	●	●	○	○	○	●	○	○	●
9	○	○	○	○	○	○	○	●	●	○	○	○	●	○	○	○
10	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○
11	●	○	○	○	●	○	○	○	○	●	○	○	○	○	●	○
12	○	●	○	○	●	●	○	○	●	○	●	●	○	●	●	○
13	○	○	●	○	○	○	○	○	○	○	○	●	●	○	●	○
14	○	○	○	○	○	○	○	●	○	○	○	●	●	○	●	●
15	●	○	○	○	○	●	○	●	○	●	○	●	○	○	●	●
16	○	○	●	○	○	○	○	●	○	●	○	●	●	○	○	●

Output Sequences

The mechanical drum sequencer derives its name from sequences of control changes on its electrical outputs. The following figure shows the sequence of On/Off controls generated by the drum pattern above. Compare the two, and you will find that they are equivalent! If you can see their equivalence, you are well on your way to understanding drum instruction operation.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step Transitions

Drum Instruction Types

There are four types of Drum instructions in the DL250-1 and DL260 CPUs:

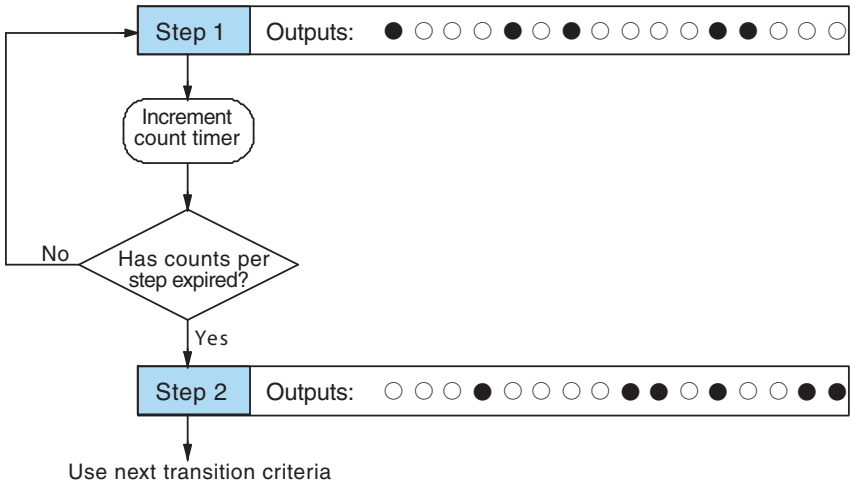
- Timed Drum with Discrete Outputs (DRUM)
- Time and Event Drum with Discrete Outputs (EDRUM)
- Masked Event Drum with Discrete Outputs (MDRMD)
- Masked Event Drum with Word Output (MDRMW)

The four drum instructions include time-based step transitions, and three include event-based transitions as well. Other options include outputs defined as a single word or as individual bits, and an output mask (individual output disable/enable).

Each drum has 16 steps, and each step has 16 outputs. Refering to the figure below, each output can be either a Y or C coil, offering programming flexibility. Step 1 has been assigned an arbitrary unique output pattern (○= Off, ● = On) as shown.

Timer-Only Transitions

Drums move from one step to another based on time and/or an external event (input). Each step has its own transition condition which you assign during the drum instruction entry. The figure below shows how timer-only transitions work.



The drum remains in Step 1 for a specific duration (user-programmable). The timebase of the timer is programmable, from 0.01 seconds to 99.99 seconds. This establishes the resolution, or the duration, of each "tick of the clock". Each step uses the same timebase, but has its own unique counts per step, which you program. The drum spends a specific amount of time in each step, given by the formula:

Time in step = 0.01 seconds X Timebase x Counts per step

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For example, if you program a 5 second time base and 12 counts for Step 1, then the drum will spend 60 seconds in Step 1. The maximum time for any step is given by the formula:

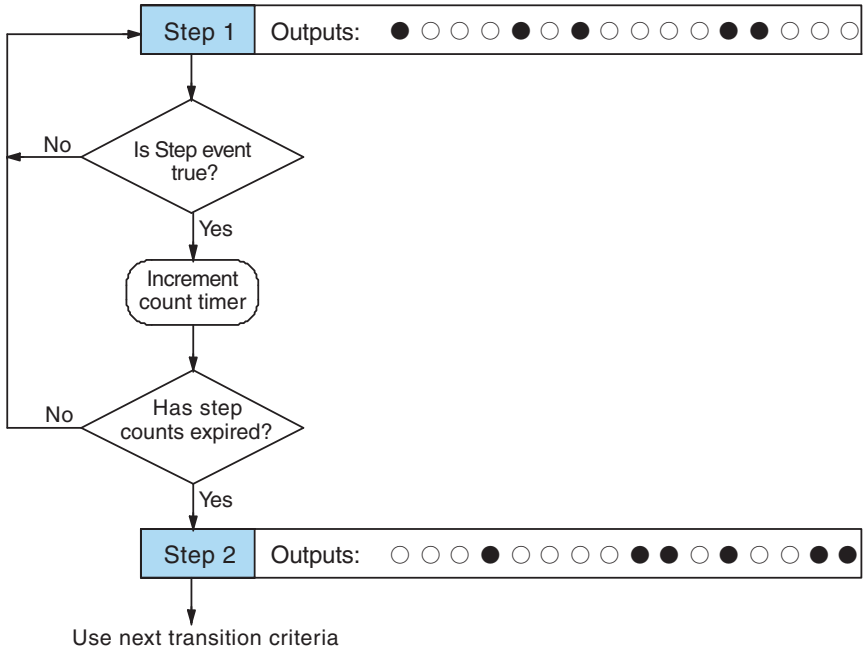
Max Time per step = 0.01 seconds X 9999 X 9999
= 999,800 seconds = 277.7 hours = 11.6 days



***NOTE**: When first choosing the timebase resolution, a good rule of thumb is to make it about 1/10 the duration of the shortest step in your drum. Then you will be able to optimize the duration of that step in 10% increments. Other steps with longer durations allow optimizing by even smaller increments (percentage-wise). Also, note that the drum instruction executes once per CPU scan. Therefore, it is pointless to specify a drum timebase that is much faster than the CPU scan time.*

Timer and Event Transitions

Step transitions may also occur based on time and/or external events. The figure below shows how step transitions work in these cases.

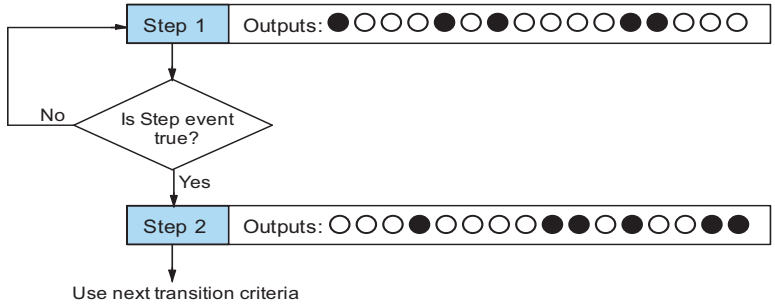


When the drum enters Step 1, it sets the output pattern as shown. Then it begins polling the external input programmed for that step. You can define event inputs as X, Y, or C discrete point types. Suppose we select X0 for the Step 1 event input. If X0 is off, then the drum remains in Step 1. When X0 is On, the event criteria is met and the timer increments. The timer increments as long as the event (X0) remains true. When the counts for Step 1 have expired, then the drum moves to Step 2. The outputs change immediately to match the new pattern for Step 2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Event-Only Transitions

Step transitions do not require both the event and the timer criteria programmed for each step. You have the option of programming just one of the two, and even mixing transition types among all the steps of the drum. For example, you might want Step 1 to transition on an event, Step 2 to transition on time only, and Step 3 to transition on both time and an event. Furthermore, you may elect to use only part of the 16 steps, and only part of the 16 outputs.



Counter Assignments

Each drum instruction uses the resources of four counters in the CPU. When programming the drum instruction, you select the first counter number. The drum also uses the next three counters automatically. The counter bit associated with the first counter turns on when the drum has completed its cycle, going off when the drum is reset. These counter values and the counter bit precisely indicate the progress of the drum instruction, and can be monitored by your ladder program.

Suppose we program a timer drum to have 8 steps, and we select CT10 for the counter number (remember, counter numbering is in octal). Counter usage is shown to the right. The right column holds typical values, interpreted below.

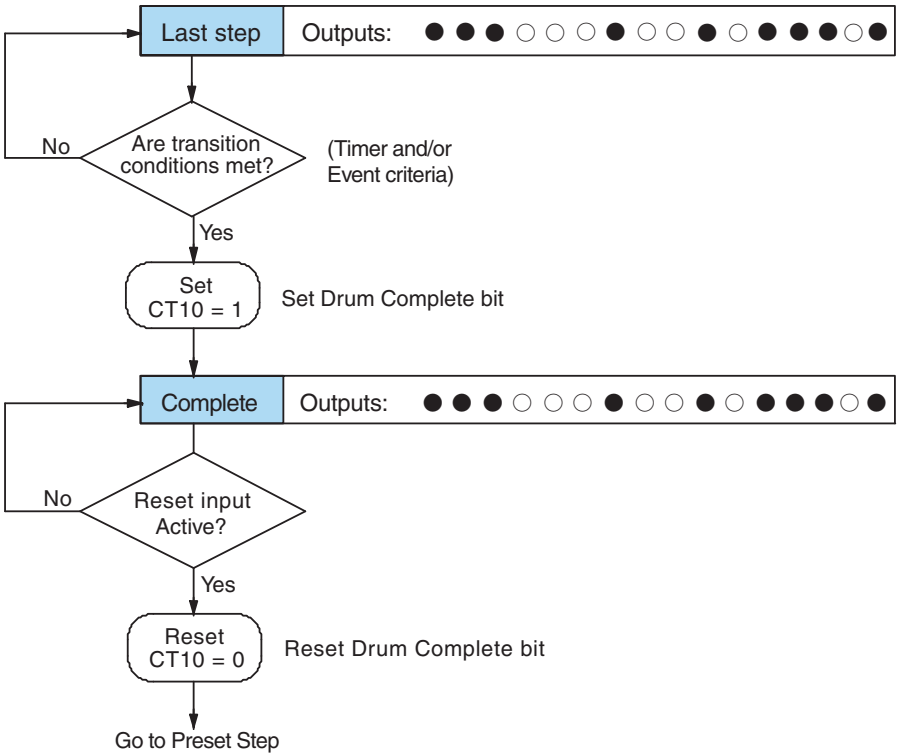
Counter Assignments			
CT10	Counts in step	V1010	1528
CT11	Timer Value	V1011	0200
CT12	Preset Step	V1012	0001
CT13	Current Step	V1013	0004

CT10 shows that we are at the 1528th count in the current step, which is step 4 (shown in CT13). If we have programmed step 4 to have 3000 counts, then the step is just over half completed. CT11 is the count timer, shown in units of 0.01 seconds. So, each least-significant-digit change represents 0.01 seconds. The value of 200 means that we have been in the current count (1528) for 2 seconds (0.01 x 200). Finally, CT12 holds the preset step value which was programmed into the drum instruction. When the drum's Reset input is active, it presets to step 1 in this case. The value of CT12 changes only if the ladder program writes to it, or the drum instruction is edited and the program is restarted. Counter bit CT10 turns on when the drum cycle is complete, and turns off when the drum is reset.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Last Step Completion

The last step in a drum sequence may be any step number, since partial drums are valid. Refer to the following figure. When the transition conditions of the last step are met, the drum sets the counter bit corresponding to the counter named in the drum instruction box (such as CT10). Then it moves to a final "drum complete" state. The drum outputs remain in the pattern defined for the last step. Having finished a drum cycle, the Start and Jog inputs have no effect at this point. The drum leaves the "drum complete" state when the Reset input becomes active (or on a program-to–run mode transition). It resets the drum complete bit (such as CT10), and then goes directly to the appropriate step number defined as the preset step.

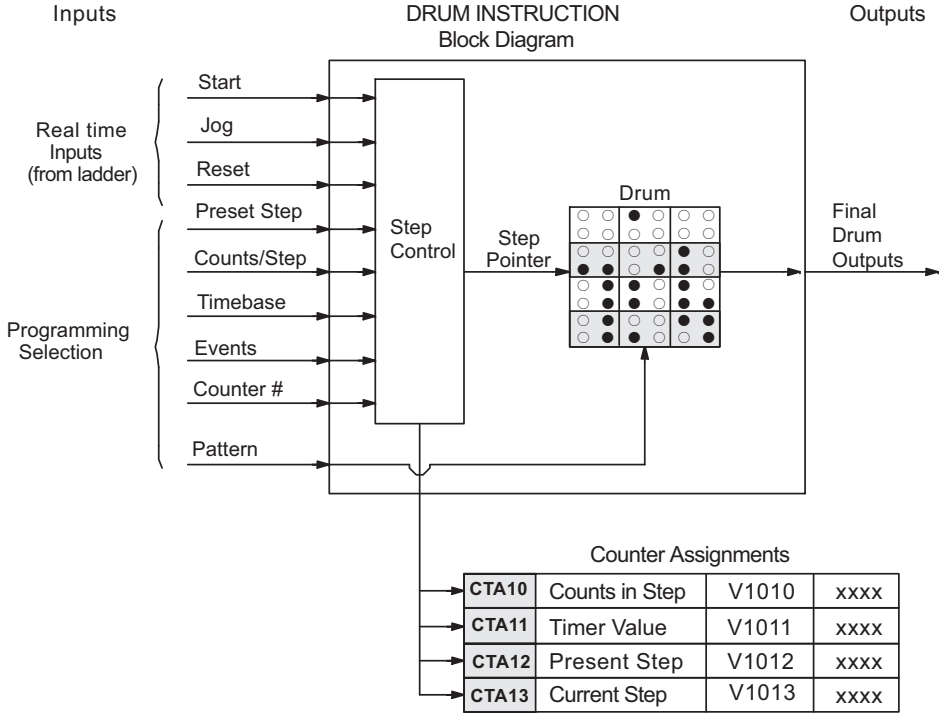


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Overview of Drum Operation

Drum Instruction Block Diagram

The drum instruction utilizes various inputs and outputs in addition to the drum pattern itself. Refer to the figure below.



The drum instruction accepts several inputs for step control, the main control of the drum.The inputs and their functions are:

- **Start** – The Start input is effective only when Reset is off. When Start is on, the drum timer runs if it is in a timed transition, and the drum looks for the input event during event transitions. When Start is off, the drum freezes in its current state (Reset must remain off), and the drum outputs maintain their current on/off pattern.
- **Jog** – The jog input is only effective when Reset is off (Start may be either on or off). The jog input increments the drum to the next step on each off-to-on transition (only EDRUM supports the jog input).
- **Reset** – The Reset input has priority over the Start input. When Reset is on, the drum moves to its preset step. When Reset is off, then the Start input operates normally.
- **Preset Step** – A step number from 1 to 16 that you define (typically is step 1). The drum moves to this step whenever Reset is on, and whenever the CPU first enters run mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Counts/Step** – The number of timer counts the drum spends in each step. Each step has its own counts parameter. However, programming the counts/step is optional.
- **Timer Value** – the current value of the counts/step timer.
- **Counter #** – The counter number specifies the first of four consecutive counters which the drum uses for step control. You can monitor these to determine the drum's progress through its control cycle.
- **Events** – Either an X, Y, C, S, T, CT, or SP type discrete point serves as step transition inputs. Each step has its own event. However, programming the event is optional on Timer/Event Drums.



WARNING: The outputs of a drum are enabled any time the CPU is in Run Mode. The Start Input does not have to be on, and the Reset input does not disable the outputs. Upon entering Run Mode, drum outputs automatically turn on or off according to the pattern of the current step of the drum. This initial step number depends on the counter memory configuration: non-retentive versus retentive.

Powerup State of Drum Registers

The choice of the starting step on powerup and program-to-run mode transitions are important to consider for your application. Please refer to the following chart. If the counter memory is configured as non-retentive, the drum is initialized the same way on every powerup or program-to-run mode transition. However, if the counter memory is configured to be retentive, the drum will stay in its previous state.

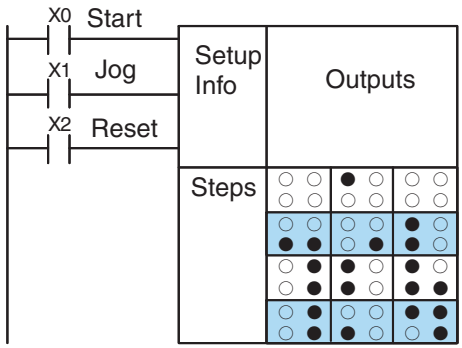
Counter Number	Function	Initialization on Powerup	
		Non-Retentive Case	Retentive Case
CTA(n)	Current Step Count	Initialize = 0	Use Previous (no change)
CTA(n + 1)	Counter Timer Value	Initialize = 0	Use Previous (no change)
CTA(n + 2)	Preset Step	Initialize = Preset Step #	Use Previous (no change)
CTA(n + 3)	Current Step #	Initialize = Preset Step #	Use Previous (no change)

Applications with relatively fast drum cycle times typically will need to be reset on powerup, using the non-retentive option. Applications with relatively long drum cycle times may need to resume at the previous point where operations stopped, using the retentive case. The default option is the retentive case. This means that if you initialize scratchpad V-memory, the memory will be retentive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum Control Techniques

Drum Control Inputs

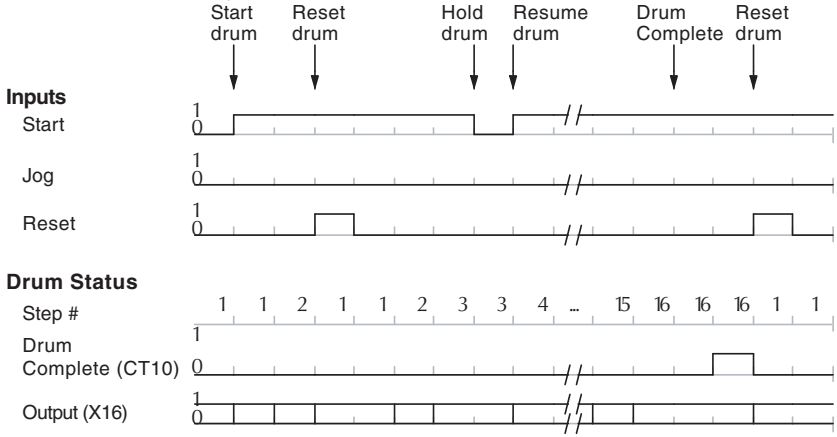


Now we are ready to put together the concepts on the previous pages and demonstrate general control of the drum instruction box. The drawing to the right shows a simplified generic drum instruction. Inputs from ladder logic control the Start, Jog, and Reset Inputs (only the EDRUM instruction supports the Jog Input). The first counter bit of the drum (CT10, for example) indicates the drum cycle is done.

The timing diagram below shows an arbitrary timer drum input sequence and how the drum responds. As the CPU enters Run mode it initializes the step number to the preset step number (typically it is Step 1). When the Start input turns on the drum begins running, waiting for an event and/or running the timer (depends on the setup).

After the drum enters Step 2, Reset turns On while Start is still On. Since Reset has priority over Start, the drum goes to the preset step (Step 1). Note that the drum is *held* in the preset step during Reset, and that step does *not run* (respond to events or run the timer) until Reset turns off.

After the drum has entered step 3, the Start input goes off momentarily, halting the drum's timer until Start turns on again.



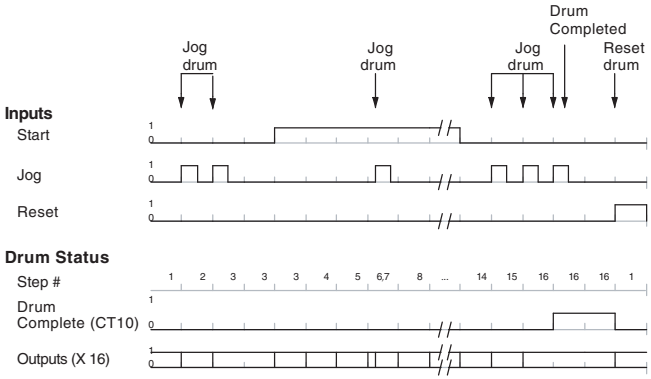
When the drum completes the last step (Step 16 in this example), the Drum Complete bit (CT10) turns on, and the step number remains at 16. When the Reset input turns on, it turns off the Drum Complete bit (CT10), and forces the drum to enter the preset step.

***NOTE**: The timing diagram shows all steps using equal time durations. Step times can vary greatly, depending on the counts/step programmed.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

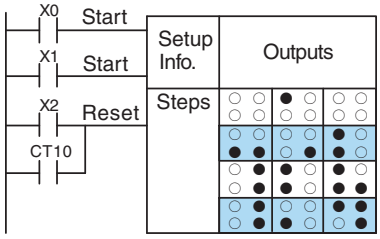
In the figure below, we focus on how the Jog input works on event drums. To the left of the diagram, note that the off-to-on transitions of the Jog input increments the step. Start may be either on or off (however, Reset must be off). Two jogs takes the drum to step three. Next, the Start input turns on, and the drum begins running normally. During step 6 another Jog input signal occurs. This increments the drum to step 7, setting the timer to 0. The drum begins running immediately in step 7, because Start is already on. The drum advances to step 8 normally.

As the drum enters step 14, the Start input turns off. Two more Jog signals moves the drum to step 16. However, note that a third Jog signal is required to move the drum through step 16 to "drum complete". Finally, a Reset input signal arrives which forces the drum into the preset step and turns off the drum complete bit.



Self-Resetting Drum

Applications often require drums that automatically start over once they complete a cycle. This is easily accomplished using the drum complete bit. In the figure to the right, the drum instruction setup is for CT10, so we logically OR the drum complete bit (CT10) with the Reset input. When the last step is done, the drum turns on CT10 which resets itself to the preset step, also resetting CT10. Contact X2 still works as a manual reset.



Initializing Drum Outputs

The outputs of a drum are enabled any time the CPU is in run mode. On program-to-run mode transitions, the drum goes to the preset step, and the outputs energize according to the pattern of that step. If your application requires all outputs to be off at powerup, make the preset step in the drum a "reset step", with all outputs off.

Using Complex Event Step Transitions

Each event-based transition accepts only one contact reference for the event. However, this does not limit events to just one contact. Just use a control relay contact such as C0 for the step transition event. Elsewhere in ladder logic, you may use C0 as an output coil, making it dependent on many other "events" (contacts).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum Instructions

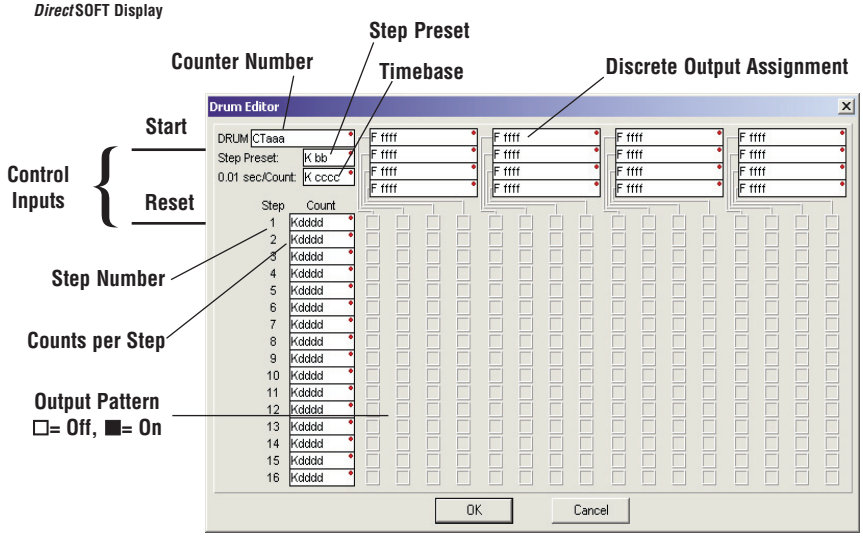
All of the DL250-1 and DL260 drum instructions may be programmed using *Direct*SOFT. The EDRUM is the only drum instruction that can be programmed with a handheld programmer (firmware version v2.21 or later). This section covers entry using *Direct*SOFT for all instructions plus the handheld mnemonics for the EDRUM instruction.

Timed Drum with Discrete Outputs (DRUM)

☒ 230
☒ 240
☑ 250-1
☑ 260

The Timed Drum with Discrete Outputs is the most basic of the DL250-1 and DL260 drum instructions. It operates according to the principles covered on the previous pages. Below is the instruction in chart form as displayed by *Direct*SOFT.

*Direct*SOFT Display



The Timed Drum features 16 steps and 16 outputs. Step transitions occur only on a timed basis, specified in counts per step. Unused steps must be programmed with "counts per step" = 0 (this is the default entry). The discrete output points may be individually assigned as X, Y, or C types, or may be left unused. The output pattern may be edited graphically with *Direct*SOFT.

Whenever the Start input is energized, the drum's timer is enabled. It stops when the last step is complete, or when the Reset input is energized. The drum enters the preset step chosen upon a CPU program-to-run mode transition, and whenever the Reset input is energized.

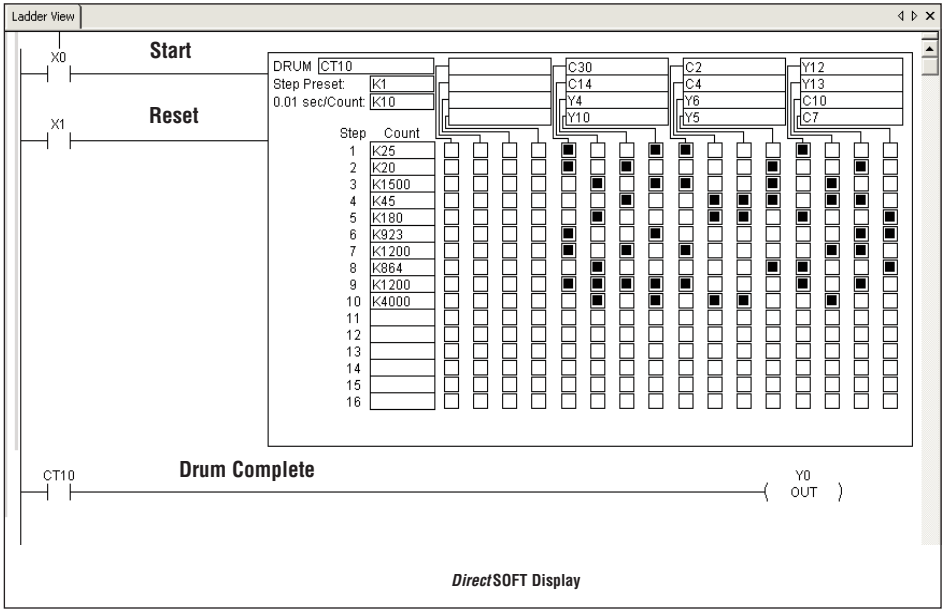
Drum Parameters	Field	Data Types	Ranges
Counter Number	aaa		0 - 174 (DL250-1) 0 - 374 (DL260)
Step Preset	bb	K	1 - 16
Timer base	cccc	K	0 - 99.99 seconds
Counts per step	dddd	K	0 - 9999
Discrete Outputs	F ffff	X, Y, C	see page 3-55 or page 3-56

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum instructions use four counters in the CPU. The ladder program can read the counter values for the drum's status. The ladder program may write a new preset step number to CTA(n+2) at any time. However, the other counters are for monitoring purposes only.

Counter Number	DL250-1 Ranges of (n)	DL260 Ranges of (n)	Function	Counter Bit Function
CTA(n)	0 - 174	0 - 374	Counts in step	CT(n) = Drum Complete
CTA(n+1)	1 - 175	1 - 375	Timer value	CT(n+1) = (not used)
CTA(n+2)	2 - 176	2 - 376	Preset Step	CT(n+2) = (not used)
CTA(n+3)	3 - 177	3 - 377	Current Step	CT(n+3) = (not used)

The following ladder program shows the DRUM instruction in a typical ladder program, as shown by ***Direct***SOFT. Steps 1 through 10 are used, and twelve of the sixteen output points are used. The preset step is step 1. The timebase runs at (K10 x 0.01) = 0.1 second per count. Therefore, the duration of step 1 is (25 x 0.1) = 2.5 seconds. In the last rung, the Drum Complete bit (CT10) turns on output Y0 upon completion of the last step (step 10). A drum reset also resets CT10.



***Direct*SOFT Display**

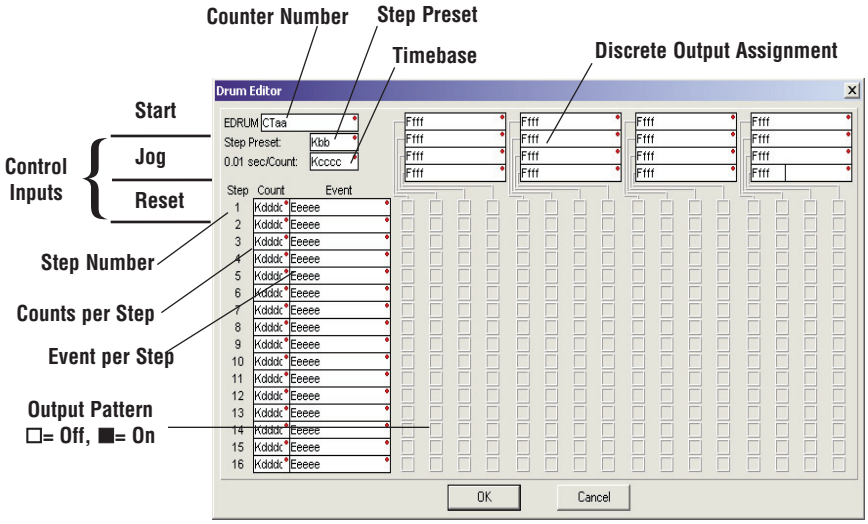
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Event Drum (EDRUM)

☒ 230
☒ 240
☑ 250-1
☑ 260

The Event Drum (EDRUM) features time-based and event-based step transitions. It operates according to the general principles of drum operation covered in the beginning of this chapter. Below is the instruction as displayed by *Direct*SOFT.

*Direct*SOFT 5 Display



The Event Drum features 16 steps and 16 discrete outputs. Step transitions occur on timed and/or event basis. The jog input also advances the step on each off-to-on transition. Time is specified in counts per step, and events are specified as discrete contacts. Unused steps and events must be left blank. The discrete output points may be individually assigned.

Drum Parameters	Field	Data Types	Ranges
Counter Number	aaa	-	0 - 174 (DL250-1) 0 - 374 (DL260)
Step Preset	bb	K	1 - 16
Timer base	cccc	K	0 - 99.99 seconds
Counts per step	dddd	K	0 - 9999
Event	eeee	X, Y, C, S, T, CT, SP	see page 3-55 or page 3-56
Discrete Outputs	F ffff	X, Y, C	

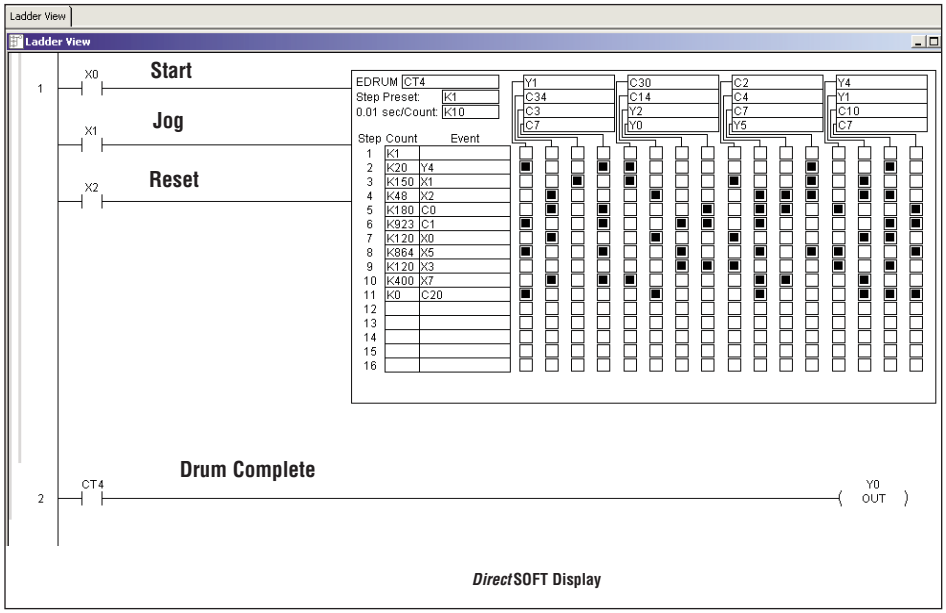
Whenever the Start input is energized, the drum's timer is enabled. As long as the event is true for the current step, the timer runs during that step. When the step count equals the counts per step, the drum transitions to the next step. This process stops when the last step is complete, or when the Reset input is energized. The drum enters the preset step chosen upon a CPU program-to-run mode transition, and whenever the Reset input is energized.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum instructions use four counters in the CPU. The ladder program can read the counter values for the drum's status. The ladder program may write a new preset step number to CTA(n+2) at any time. However, the other counters are for monitoring purposes only.

Counter Number	DL250-1 Ranges of (n)	DL260 Ranges of (n)	Function	Counter Bit Function
CTA(n)	0 - 174	0 - 374	Counts in step	CT(n) = Drum Complete
CTA(n+1)	1 - 175	1 - 375	Timer value	CT(n+1) = (not used)
CTA(n+2)	2 - 176	2 - 376	Preset Step	CT(n+2) = (not used)
CTA(n+3)	3 - 177	3 - 377	Current Step	CT(n+3) = (not used)

The following ladder program shows the EDRUM instruction in a typical ladder program, as shown by *Direct*SOFT. Steps 1 through 11 are used, and all sixteen output points are used. The preset step is step 1. The timebase runs at (K10 x 0.01) = 0.1 second per count. Therefore, the duration of step 1 is (1 x 0.1) = 0.1 second. Note that step 1 is time-based only (event is left blank). And, the output pattern for step 1 programs all outputs off, which is a typically desirable powerup condition. In the last rung, the Drum Complete bit (CT4) turns on output Y0 upon completion of the last step (step 11). A drum reset also resets CT4.



***NOTE**: If all events are true in an event only drum (a drum with 0 counts per step in all steps), the PLC completes one step of the drum per scan; thus, the drum will be complete in 16 scans. However, as the outputs of the drum are enabled any time the CPU is in RUN Mode, the drum discrete outputs will be energized as pulsed outputs for each scan.*

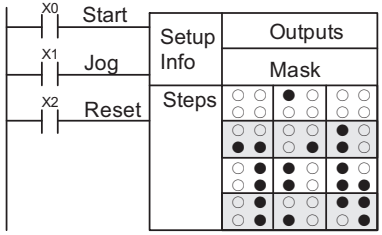
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Handheld Programmer Drum Mnemonics

The EDRUM instruction can also be programmed using a handheld programmer. This section explains entry via the handheld programmer.

First, enter Store instructions for the ladder rungs controlling the drum's ladder inputs. In the example to the right, the timer drum's Start, Jog, and Reset inputs are controlled by X0, X1 and X2 respectively. The required keystrokes are listed beside the mnemonic.

These keystrokes precede the EDRUM instruction mnemonic. Note that the ladder rungs for Start, Jog, and Reset inputs are not limited to being single–contact rungs.



Handheld Programmer Keystrokes

Store X0 | $ STR | → | A 0 | ENT

(Repeat for Store X1 and Store X2)

Handheld Programmer Keystrokes

EDRUM CNT4 | SHFT | E 4 | D 3 | R ORN | U ISG | M ORST | → | E 4 | ENT

After the Store instructions, enter the EDRUM (using Counter CT0) as shown:

After entering the EDRUM mnemonic as above, the handheld programmer creates an input form for all the drum parameters. The input form consists of approximately fifty or more default mnemonic entries containing DEF (define) statements. The default mnemonics are already "input" for you, so they appear automatically. Use the NXT and PREV keys to move forward and backward through the form. Only the editing of default values is required, thus eliminating many keystrokes. The entries required for the basic timer drum are in the chart below.

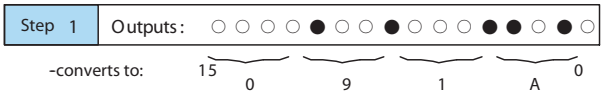


***NOTE**: Default entries for output points and events are "DEF 0000", which means they are unassigned. If you need to go back and change an assigned output as unused again, enter "K0000". The entry will again show as "DEF 0000".*

Drum Parameters	Multiple Entries	Mnemonic / Entry	Default Mnemonic	Valid Data Types	Ranges
Start Input	--	STR (plus input rung)	--	--	--
Jog Input	--	STR (plus input rung)	--	--	--
Reset Input	--	STR (plus input rung)	--	--	--
Drum Mnemonic	--	DRUM CNT aa	--	CT	0 - 174 (DL250-1) 0 - 374 (DL260)
Step Preset	1	bb	DEF K0000	K	1 - 16
Timer base	1	cccc	DEF K0000	K	0 - 9999
Counts per step	16	dddd	DEF K0000	K	0 - 9999
Events	16	eeee	DEF K0000	X, Y, C, S, T, CT, SP	see either page 3-55 or page 3-56
Output points	16	ffff	DEF K0000	X, Y, C	
Output Mask	16	gggg	DEF K0000	K	0 - FFFF

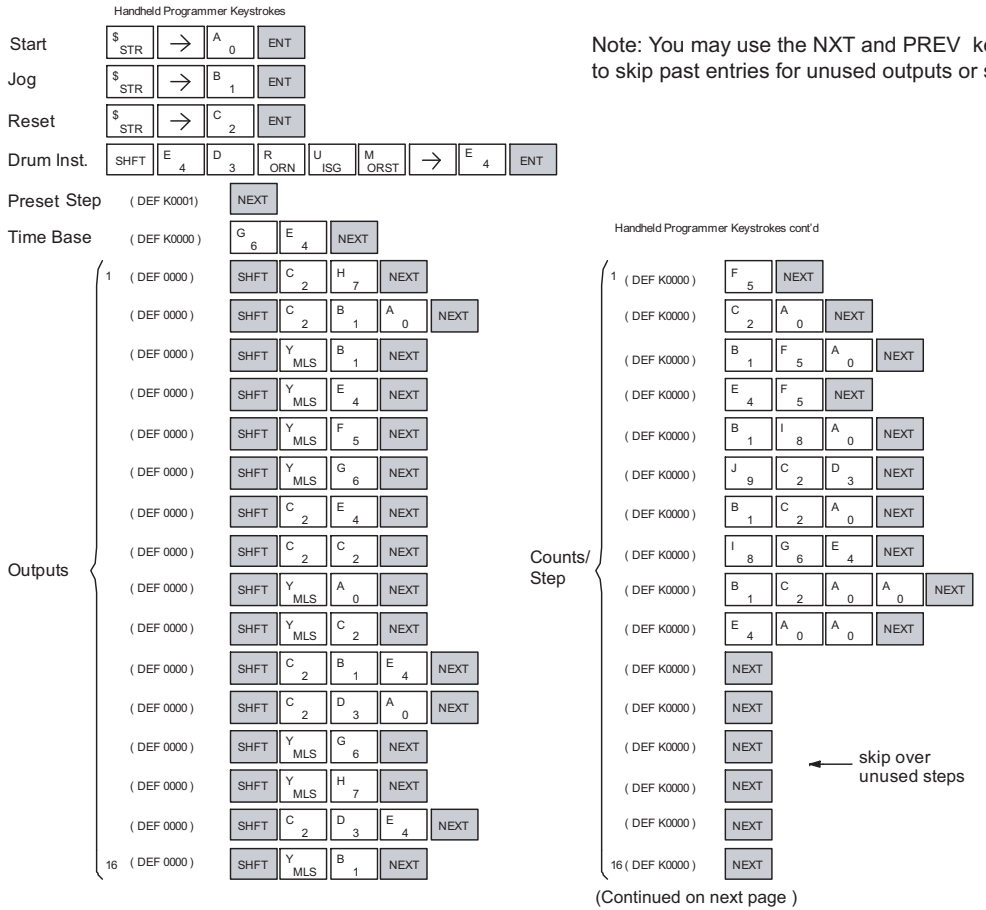
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the DRUM entry chart (two pages before), we show the method of entry for the basic time/event drum instruction. First, we convert the output pattern for each step to the equivalent hex number, as shown in the following example.

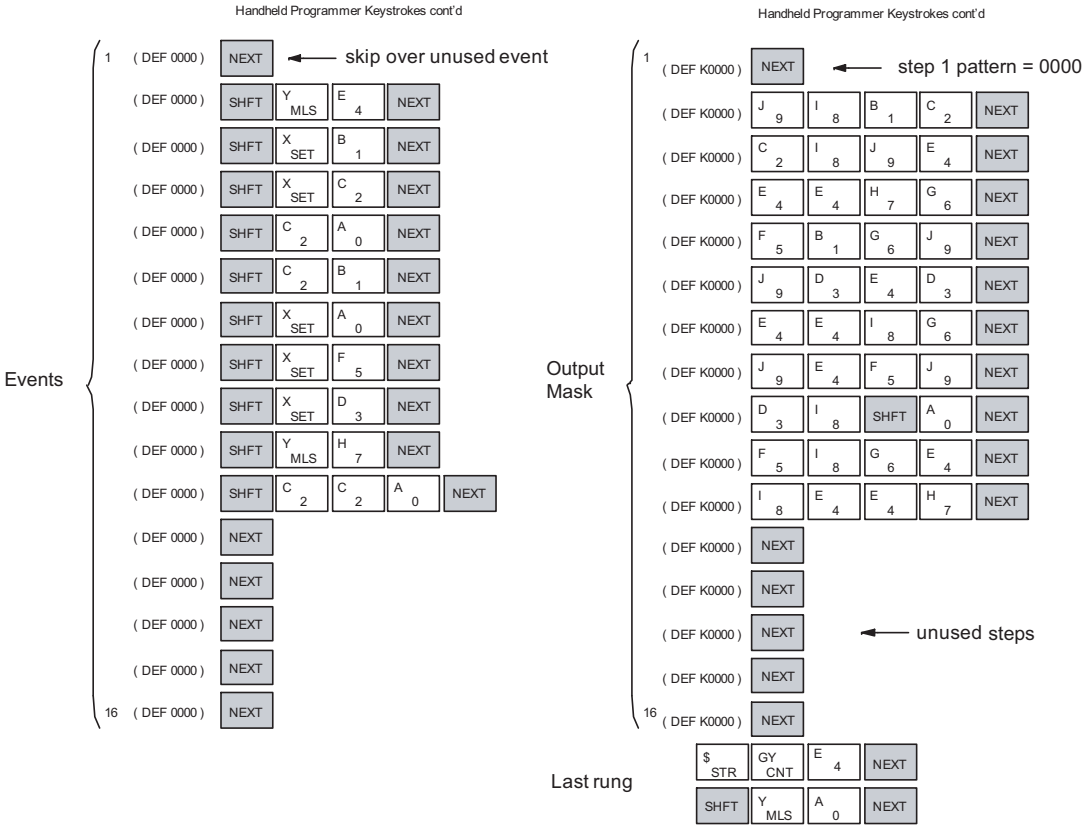


The following diagram shows the method for entering the previous EDRUM example on the HPP. The default entries of the form are in parenthesis. After the drum instruction entry (on the fourth row), the remaining keystrokes over-write the numeric portion of each default DEF statement.

NOTE: Drum editing requires Handheld Programmer firmware version 2.21 or later.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



***NOTE**: Remember, you may use the NXT and PREV keys to skip past entries for unused outputs or steps.*

***NOTE:** For ease of operation when using the EDRUM instruction, we recommend using **Direct**SOFT over the handheld programmer.*

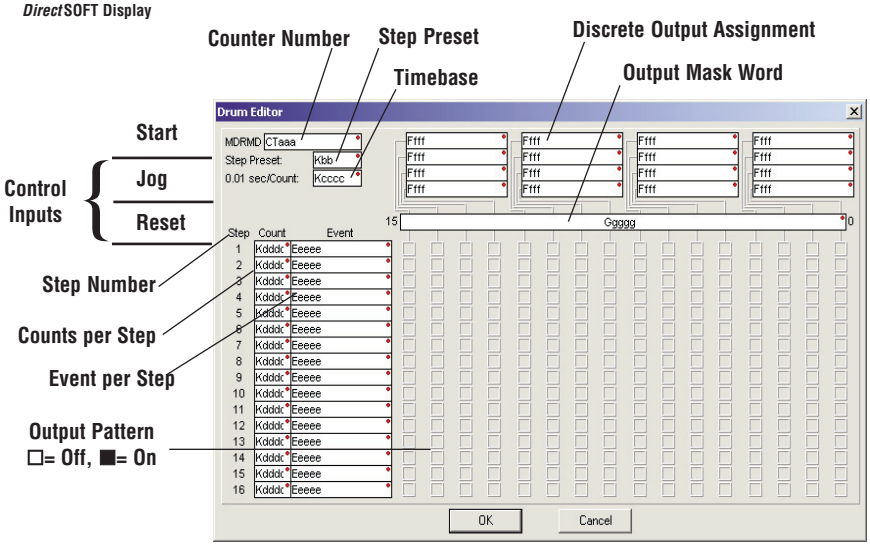
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Masked Event Drum with Discrete Outputs (MDRMD)

☒ 230
☒ 240
☑ 250-1
☑ 260

The Masked Event Drum with Discrete Outputs has all the features of the basic Event Drum plus final output control for each step. It operates according to the general principles of drum operation covered in the beginning of this section. Below is the instruction in chart form as displayed by *Direct*SOFT.

*Direct*SOFT Display



The Masked Event Drum with Discrete Outputs features sixteen steps and sixteen outputs. Drum outputs are logically ANDed bit-by-bit with an output mask word for each step. The Ggggg field specifies the beginning location of the 16 mask words. Step transitions occur on timed and/or event basis. The jog input also advances the step on each off-to-on transition. Time is specified in counts per step, and events are specified as discrete contacts. Unused steps and events can be left blank (this is the default entry). Whenever the Start input is energized, the drum's timer is enabled. As long as the event is true for the current step, the timer runs during that step. When the step count equals the counts per step, the drum transitions to the next step. This process stops when the last step is complete, or when the Reset input is energized. The drum enters the preset step chosen upon a CPU program-to-run mode transition, and whenever the Reset input is energized.

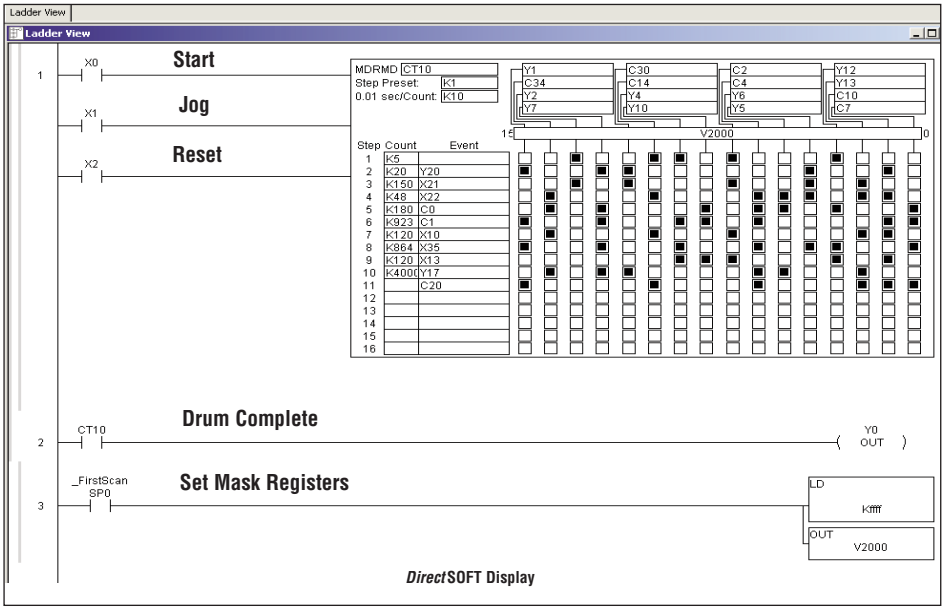
Drum Parameters	Field	Data Types	Ranges
Counter Number	aaa	–	0 - 174 (DL250-1) 0 - 374 (DL260)
Step Preset	bb	K	1 – 16
Timer base	cccc	K	0 – 99.99 seconds
Counts per step	dddd	K	0 – 9999
Event	eeee	X, Y, C, S, T, ST, GX, GY, CT, SP	see either page 3-55 or page 3-56
Discrete Outputs	Ffff	X, Y, C, GX, GY	
Output Mask	Ggggg	V	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum instructions use four counters in the CPU. The ladder program can read the counter values for the drum's status. The ladder program may write a new preset step number to CTA(n+2) at any time. However, the other counters are for monitoring purposes only.

Counter Number	DL250-1 Ranges of (n)	DL260 Ranges of (n)	Function	Counter Bit Function
CTA(n)	0 - 174	0 - 374	Counts in step	CT(n) = Drum Complete
CTA(n+1)	1 - 175	1 - 375	Timer value	CT(n+1) = (not used)
CTA(n+2)	2 - 176	2 - 376	Preset Step	CT(n+2) = (not used)
CTA(n+3)	3 - 177	3 - 377	Current Step	CT(n+3) = (not used)

The following ladder program shows the MDRMD instruction in a typical ladder program, as shown by *Direct*SOFT. Steps 1 through 11 are used, and all 16 output points are used. The output mask word is at V2000. The final drum outputs are shown above the mask word as individual bits. The data bits in V2000 are logically ANDed with the output pattern of the current step in the drum. If you want all drum outputs to be off after powerup, write zeros to V2000 on the first scan. Ladder logic may update the output mask at any time to enable or disable the drum outputs The preset step is step 1. The timebase runs at (K10 x 0.01) = 0.1 second per count. Therefore, the duration of step 1 is (5 x 0.1) = 0.5 seconds. Note that step 1 is time-based only (event is left blank). In the last rung, the Drum Complete bit (CT10) turns on output Y0 upon completion of the last step (step 11). A drum reset also resets CT10.



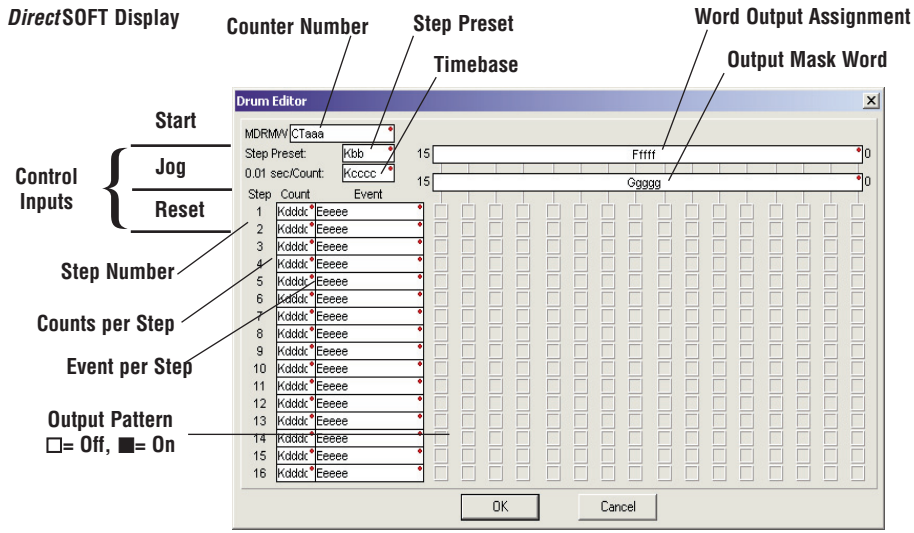
***NOTE**: The ladder program must load constants in V2000 through V2012 to cover all mask registers for the eleven steps used in this drum.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Masked Event Drum with Word Output (MDRMW)

☒ 230
☒ 240
☑ 250-1
☑ 260

The Masked Event Drum with Word Output features outputs organized as bits of a single word, rather than discrete points. It operates according to the general principles of drum operation covered in the beginning of this section. Below is the instruction in chart form as displayed by *Direct*SOFT.



The Masked Event Drum with Word Output features sixteen steps and sixteen outputs. Drum outputs are logically ANDed bit-by-bit with an output mask word for each step. The Ggggg field specifies the beginning location of the 16 mask words, creating the final output (Ffff field). Step transitions occur on timed and/or event basis. The jog input also advances the step on each off-to-on transition. Time is specified in counts per step, and events are specified as discrete contacts. Unused steps and events can be left blank (this is the default entry). Whenever the Start input is energized, the drum's timer is enabled. As long as the event is true for the current step, the timer runs during that step. When the step count equals the counts per step, the drum transitions to the next step. This process stops when the last step is complete, or when the Reset input is energized. The drum enters the preset step chosen upon a CPU program-to-run mode transition, and whenever the Reset input is energized.

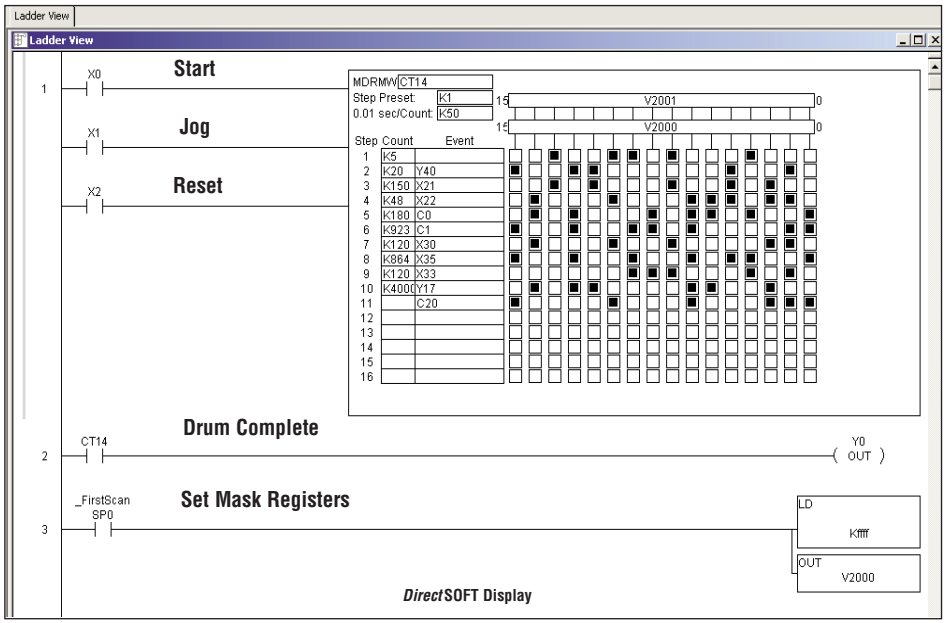
Drum Parameters	Field	Data Types	Ranges
Counter Number	aaa	–	0-174 (DL250-1) 0-374 (DL260)
Preset Step	bb	K	1 – 16
Timer base	cccc	K	0 – 99.99 seconds
Counts per step	dddd	K	0 – 9999
Event	eeee	X, Y, C, S, T, CT, GX, GY, SP	see either page 3-55 or page 3-56
Word Output	Fffff	V	
Output Mask	Ggggg	V	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drum instructions use four counters in the CPU. The ladder program can read the counter values for the drum's status. The ladder program may write a new preset step number to CTA(n+2) at any time. However, the other counters are for monitoring purposes only.

Counter Number	DL250-1 Ranges of (n)	DL260 Ranges of (n)	Function	Counter Bit Function
CTA(n)	0 - 174	0 - 374	Counts in step	CT(n) = Drum Complete
CTA(n+1)	1 - 175	1 - 375	Timer value	CT(n+1) = (not used)
CTA(n+2)	2 - 176	2 - 376	Preset Step	CT(n+2) = (not used)
CTA(n+3)	3 - 177	3 - 377	Current Step	CT(n+3) = (not used)

The following ladder program shows the MDRMW instruction in a typical ladder program, as shown by *Direct*SOFT. Steps 1 through 11 are used, and all sixteen output points are used. The output mask word is at V2000. The final drum outputs are shown above the mask word as a word at V2001. The data bits in V2000 are logically ANDed with the output pattern of the current step in the drum, generating the contents of V2001. If you want all drum outputs to be off after powerup, write zeros to V2000 on the first scan. Ladder logic may update the output mask at any time to enable or disable the drum outputs. The preset step is step 1. The timebase runs at (K50 x 0.01) = 0.5 seconds per count. Therefore, the duration of step 1 is (5 x 0.5) = 2.5 seconds. Note that step 1 is time-based only (event is left blank). In the last rung, the Drum Complete bit (CT14) turns on output Y0 upon completion of the last step (step 11). A drum reset also resets CT14.



***NOTE**: The ladder program must load constants in V2000 through V2012 to cover all mask registers for the eleven steps used in this drum.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RLL*PLUS* Stage Programming



In This Chapter...

Introduction to Stage Programming 7–2
Learning to Draw State Transition Diagrams 7–3
Using the Stage Jump Instruction for State Transitions 7–7
Stage Program Example: Toggle On/Off Lamp Controller 7–8
Four Steps to Writing a Stage Program 7–9
Stage Program Example: A Garage Door Opener 7–10
Stage Program Design Considerations 7–15
Parallel Processing Concepts 7–19
Managing Large Programs 7–21
RLL*PLUS* (Stage) Instructions 7–23
Questions and Answers about Stage Programming 7–29

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

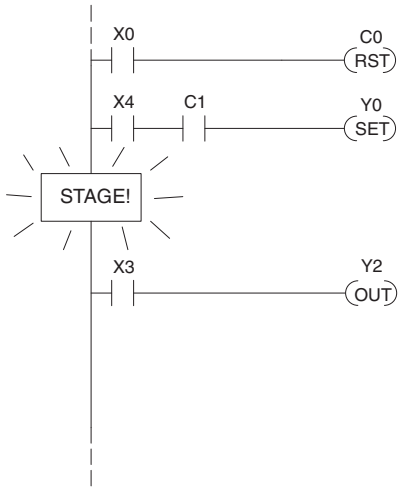
Introduction to Stage Programming

☑ 230
☑ 240
☑ 250-1
☑ 260

Stage Programming (available in all DL205 CPUs) provides a way to organize and program complex applications with relative ease, when compared to purely relay ladder logic (RLL) solutions. Stage programming does not replace or negate the use of traditional boolean ladder programming. This is why Stage Programming is also called RLL^{PLUS}. You will not have to discard any training or experience you already have. Stage programming simply allows you to divide and organize a RLL program into groups of ladder instructions called stages. This allows quicker and more intuitive ladder program development than traditional RLL alone provides.

Overcoming "Stage Fright"

Many PLC programmers in the industry have become comfortable using RLL for every PLC program they write... but often remain skeptical or even fearful of learning new techniques such as stage programming. While RLL is great at solving boolean logic relationships, it has disadvantages as well:



- Large programs can become almost unmanageable, because of a lack of structure.
- In RLL, latches must be tediously created from self-latching relays.
- When a process gets stuck, it is difficult to find the rung where the error occurred.
- Programs become difficult to modify later, because they do not intuitively resemble the application problem they are solving.

It's easy to see that these inefficiencies consume a lot of additional time, and time is money. *Stage programming overcomes these obstacles!* We believe a few moments of studying the stage concept is one of the greatest investments in programming speed and efficiency a PLC programmer can make!

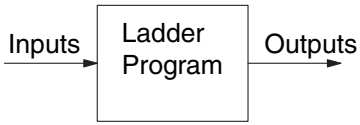
So, we encourage you to study stage programming and add it to your "toolbox" of programming techniques. This chapter is designed as a self-paced tutorial on stage programming. For best results:

- Start at the beginning and do not skip over any sections.
- Study each stage programing concept by working through each example. The examples build progressively on each other.
- Read the Stage Questions and Answers at the end of the chapter for a quick review.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

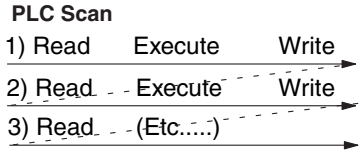
Learning to Draw State Transition Diagrams

Introduction to Process States



Those familiar with ladder program execution know the CPU must scan the ladder program repeatedly, over and over. Its three basic steps are:

1. Read the inputs
2. Execute the ladder program
3. Write the outputs



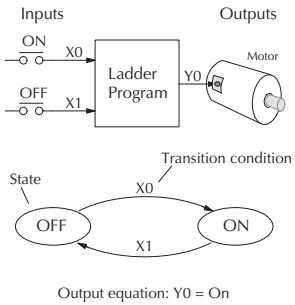
The benefit is that a change at the inputs can affect the outputs in just a few milliseconds.

Most manufacturing processes consist of a series of activities or conditions, each lasting for several seconds, minutes, or even hours. We might call these "process states", which are either active or inactive at any particular time. A challenge for RLL programs is that a particular input event may last for a brief instant. We typically create latching relays in RLL to preserve the input event in order to maintain a process state for the required duration. We can organize and divide ladder logic into sections called "stages", representing process states. But before we describe stages in detail, we will reveal **the secret to understanding stage programming:** state transition diagrams.

The Need for State Diagrams

Sometimes we need to forget about the scan nature of PLCs, and focus our thinking toward the states of the process we need to identify. Clear thinking and concise analysis of an application gives us the best chance at writing efficient, bug-free programs. *State diagrams are tools to help us draw a picture of our process!* You will discover that if we can get the picture right, **our program will also be right!**

A 2–State Process



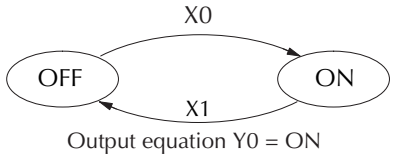
Consider the simple process shown to the right, which controls an industrial motor. We will use a green momentary SPST pushbutton to turn the motor on, and a red one to turn it off. The machine operator will press the appropriate pushbutton for a second or so. The two states of our process are ON and OFF.

The next step is to draw a *state transition diagram*, as shown to the right. It shows the two states OFF and ON, with two transition lines in-between. When the event X0 is true, we transition from OFF to ON. When X1 is true, we transition from ON to OFF.

If you're following along, you are very close to grasping the concept and the problem-solving power of state transition diagrams. The output of our controller is Y0, which is true any time we are in the ON state. In a boolean sense, Y0=ON state.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The state transition diagram to the right is a picture of the solution we need to create. The beauty of it is this: it expresses the problem independently of the programming language we may use to realize it. In other words, *by drawing the diagram we have already solved the control problem!*

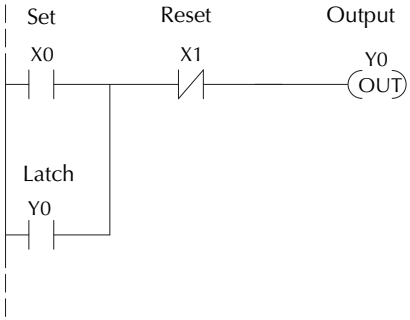


First, we'll translate the state diagram to traditional RLL. Then we'll show how easy it is to translate the diagram into a stage programming solution.

RLL Equivalent

The RLL solution is shown to the right. It consists of a self-latching motor output coil, Y0. When the On pushbutton (X0) is pressed, output coil Y0 turns on and the Y0 contact on the second row latches itself on. So, X0 **sets the latch** Y0 on, and it remains on after the X0 contact opens.

When the Off pushbutton (X1) is pressed, it opens the normally-closed X1 contact, which **resets the latch**. Motor output Y0 turns off.

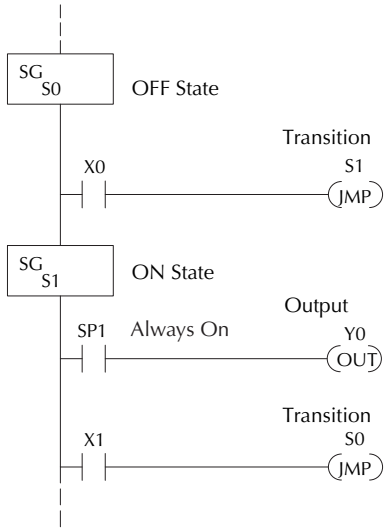


Stage Equivalent

The stage program solution is shown to the right. The two inline stage boxes S0 and S1 correspond to the two states OFF and ON. The ladder rung(s) below each stage box belong to each respective stage. This means that *the PLC only has to scan those rungs when the corresponding stage is active!*

For now, let's assume we begin in the OFF State, so stage S0 is active. When the On pushbutton (X0) is pressed, a stage transition occurs. The JMP S1 instruction executes, which simply turns off the Stage bit S0 and turns on Stage bit S1. So on the next PLC scan, the CPU will not execute Stage S0, but will execute stage S1!

In the On State (Stage S1), we want the motor to always be on. The special relay contact SP1 is defined as always on, so Y0 turns the motor on.



When the Off pushbutton (X1) is pressed, a transition back to the Off State occurs. The JMP S0 instruction executes, which simply turns off the Stage bit S1 and turns on Stage bit S0. On the next PLC scan, the CPU will not execute Stage S1, so the motor output Y0 will turn off. The Off state (Stage 0) will be ready for the next cycle.

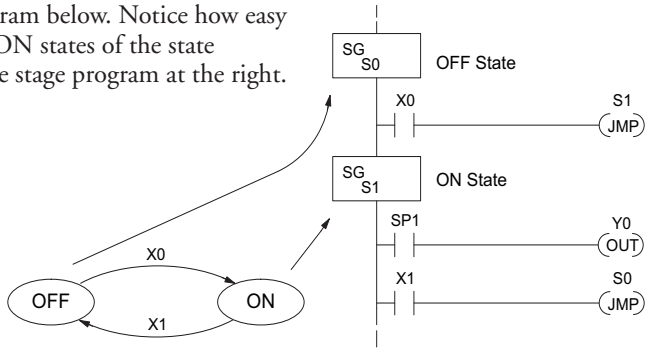
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Let's Compare

Right now, you may be thinking "I don't see the big advantage to Stage Programming... in fact, the stage program is longer than the plain RLL program". Well, now is the time to exercise a bit of faith. As control problems grow in complexity, stage programming quickly out-performs RLL in simplicity, program size, etc.

For example, consider the diagram below. Notice how easy it is to correlate the OFF and ON states of the state transition diagram below to the stage program at the right.

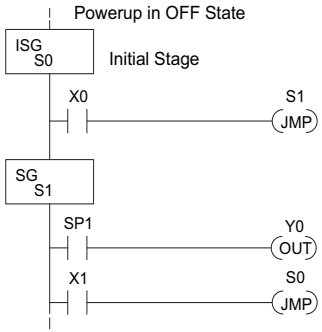
Now, we challenge anyone to easily identify the same states in the RLL program on the previous page!



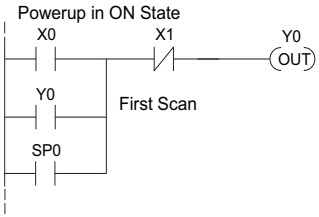
Initial Stages

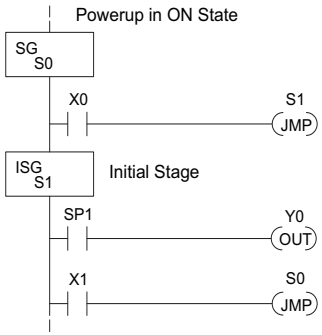
At powerup and Program-to-Run Mode transitions, the PLC always begins with all normal stages (SG) off. So, the stage programs shown so far have actually had no way to get started (because rungs are not scanned unless their stage is active).

Assume that we want to always begin in the Off state (motor off), which is how the RLL program works. The Initial Stage (ISG) is defined to be active at powerup. In the modified program to the right, we have changed stage S0 to the ISG type. This ensures the PLC will scan contact X0 after powerup, because Stage S0 is active. **After powerup, an Initial Stage (ISG) works just like any other stage!**



We can change both programs so that the motor is ON at powerup. In the RLL below, we must add a first scan relay SP0, latching Y0 on. In the stage example to the right, we simply make Stage S1 an initial stage (ISG) instead of S0.

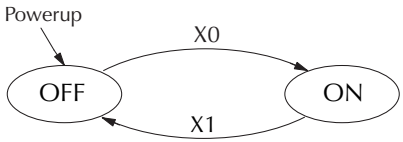




NOTE: If the ISG is within the retentive range for stages, the ISG will remain in the state it was in before power down and will NOT turn itself on during the first scan.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

We can mark our desired powerup state as shown to the right, which helps us remember to use the appropriate Initial Stages when creating a stage program. It is permissible to have as many initial stages as the process requires.

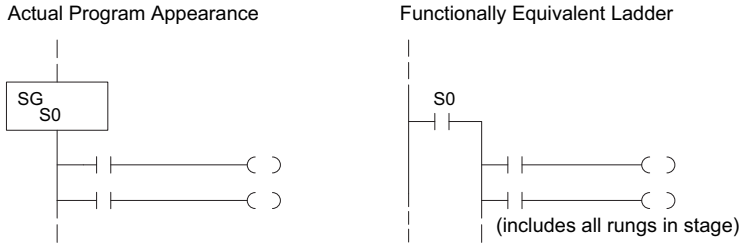


What Stage Bits Do

You may recall that a stage is a section of ladder program which is either active or inactive at a given moment. All stage bits (S0 to Sxxx) reside in the PLC's image register as individual status bits. Each stage bit is either a boolean 0 or 1 at any time.

Program execution always reads ladder rungs from top to bottom, and from left to right. The drawing below shows the effect of stage bit status. The ladder rungs below the stage instruction continuing until the next stage instruction or the end of program belong to stage 0. Its equivalent operation is shown on the right. When S0 is true, the two rungs have power flow.

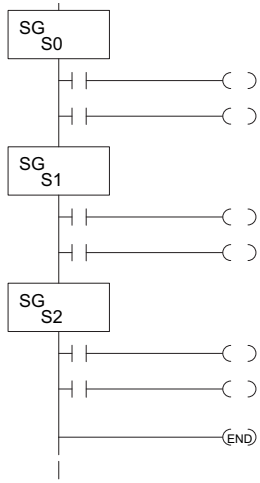
- If Stage bit S0 = 0, its ladder rungs *are not scanned* (executed).
- If Stage bit S0 = 1, its ladder rungs *are scanned* (executed).



Stage Instruction Characteristics

The inline stage boxes on the left power rail divide the ladder program rungs into stages. Some stage rules are:

- **Execution** – Only logic in active stages are executed on any scan.
- **Transitions** – Stage transition instructions take effect on the next occurrence of the stages involved.
- **Octal numbering** – Stages are numbered in octal, like I/O points, etc. So "S8" is not valid.
- **Total Stages** – The maximum number of stages is CPU-dependent.
- **No duplicates** – Each stage number is unique and can be used just once.
- **Any order** – You can skip numbers and sequence the stage numbers in any order.
- **Last Stage** – The last stage in the ladder program includes all rungs from its stage box until the end coil.

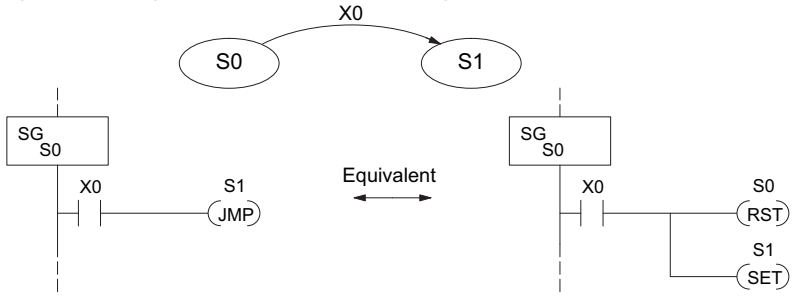


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the Stage Jump Instruction for State Transitions

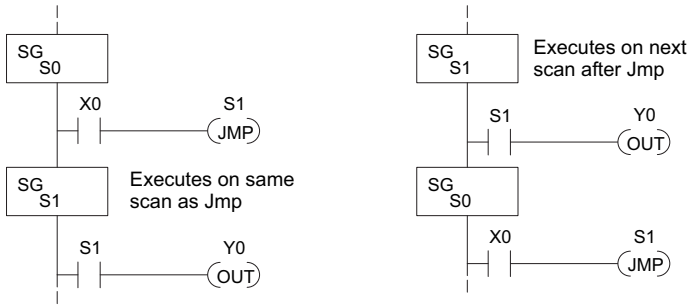
Stage Jump, Set, and Reset Instructions

The Stage JMP instruction we have used deactivates the stage in which the instruction occurs, while activating the stage in the JMP instruction. Refer to the state transition shown below. When contact X0 energizes, the state transition from S0 to S1 occurs. The two stage examples shown below are equivalent. So, the Stage Jump instruction is equal to a Stage Reset of the current stage, plus a Stage Set instruction for the stage to which we want to transition.



Please Read Carefully – The jump instruction is easily misunderstood. The "jump" does not occur immediately like a GOTO or GOSUB program control instruction when executed. Here's how it works:

- The jump instruction resets the stage bit of the stage in which it occurs. All rungs in the stage still finish executing during the current scan, *even if there are other rungs in the stage below the jump instruction!*
- The reset will be in effect on the following scan, so the stage that executed the jump instruction previously will be inactive and bypassed.
- The stage bit of the stage named in the Jump instruction will be set immediately, so the stage will be executed on its next occurrence. In the left program shown below, stage S1 executes during the same scan as the JMP S1 occurs in S0. In the example on the right, Stage S1 executes on the next scan after the JMP S1 executes, because stage S1 is located above stage S0.



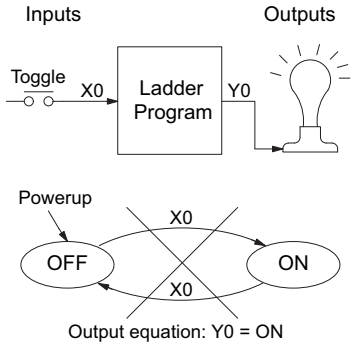
***NOTE**: Assume we start with Stage 0 active and Stage 1 inactive for both examples.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Stage Program Example: Toggle On/Off Lamp Controller

A 4–State Process

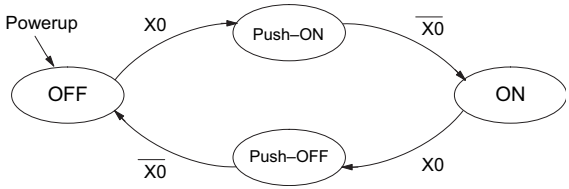
In the process shown to the right, we use an ordinary momentary pushbutton to control a light bulb. The ladder program will latch the switch input, so that we will push and release to turn on the light, push and release again to turn it off (sometimes called toggle function). Sure, we could buy a mechanical switch with the alternate on/off action built in... However, this example is educational and also fun!



Next we draw the state transition diagram. A typical first approach is to use X0 for both transitions (like the example shown to the right). However, *this is incorrect* (please keep reading).

Note that this example differs from the motor example, because now we have only one pushbutton. When we press the pushbutton, both transition conditions are met. We would transition around the state diagram at top speed. If implemented in Stage, this solution would flash the light on or off each scan (obviously undesirable)!

The solution is to make the push and the release of the pushbutton separate events. Refer to the new state transition diagram below. At powerup we enter the OFF state. When switch X0 is pressed, we enter the Press-ON state. When it is released, we enter the ON state. Note that X0 with the bar above it denotes X0 NOT.

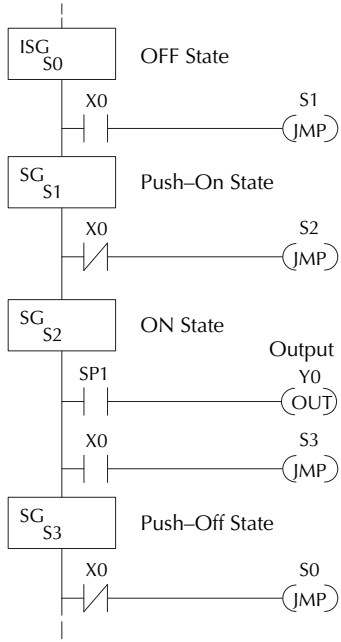


Output equation: Y0 = ON

When in the ON state, another push and release cycle similarly takes us back to the OFF state. Now we have two unique states (OFF and ON) used when the pushbutton is released, which is what was required to solve the control problem.

The equivalent stage program is shown to the right. The desired powerup state is OFF, so we make S0 an initial stage (ISG). In the ON state, we add special relay contact SP1, which is always on.

Note that even as our programs grow more complex, it is still easy to correlate the state transition diagram with the stage program!



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Four Steps to Writing a Stage Program

By now, you've probably noticed that we follow the same steps to solve each example problem. The steps will probably come to you automatically if you work through all the examples in this chapter. It's helpful to have a checklist to guide us through the problem solving. The following steps summarize the stage program design procedure:

1. Write a Word Description of the application.

Describe all functions of the process in your own words. Start by listing what happens first, then next, etc. If you find there are too many things happening at once, try dividing the problem into more than one process. Remember, you can still have the processes communicate with each other to coordinate their overall activity.

2. Draw the Block Diagram.

Inputs represent all the information the process needs for decisions, and outputs connect to all devices controlled by the process.

- Make lists of inputs and outputs for the process.
- Assign I/O point numbers (X and Y) to physical inputs and outputs.

3. Draw the State Transition Diagram.

The state transition diagram describes the central function of the block diagram, reading inputs and generating outputs.

- Identify and name the states of the process.
- Identify the event(s) required for each transition between states.
- Ensure the process has a way to re-start itself, or is cyclical.
- Choose the powerup state for your process.
- Write the output equations.

4. Write the Stage Program.

Translate the state transition diagram into a stage program.

- Make each state a stage. Remember to number stages in octal. Up to 256 total stages are available in the DL230 CPU. Up to 512 total stages are available in the DL240 CPU. Up to 1024 total stages are available in the DL250–1 and DL260 CPUs.
- Put transition logic inside the stage which originates each transition (the stage each arrow points away from).
- Use an initial stage (ISG) for any states that must be active at powerup.
- Place the outputs or actions in the appropriate stages.

You will notice that Steps 1 through 3 prepare us to write the stage program in Step 4. However, the program virtually writes itself because of the preparation beforehand. Soon you will be able to start with a word description of an application and create a stage program in one easy session!

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Stage Program Example: A Garage Door Opener

Garage Door Opener Example

In this next stage programming example we will create a garage door opener controller. Hopefully most readers are familiar with this application, and we can have fun besides!

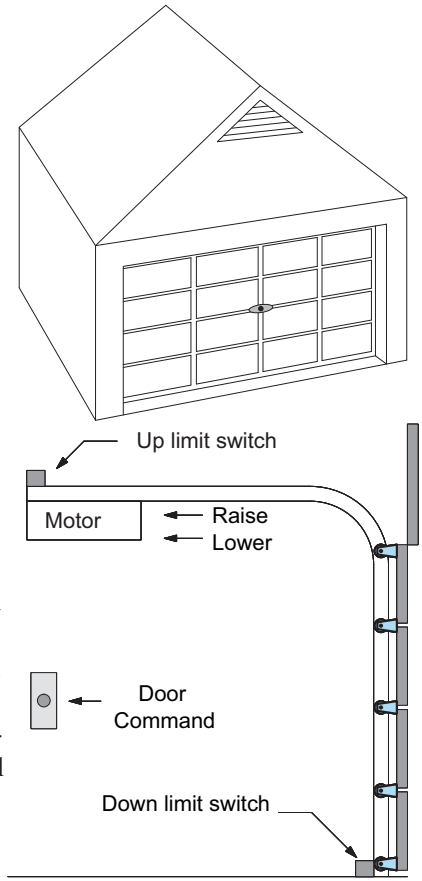
The first step we must take is to describe how the door opener works. We will start by achieving the basic operation, waiting to add extra features later (stage programs are very easy to modify).

Our garage door controller has a motor which raises or lowers the door on command. The garage owner pushes and releases a momentary pushbutton once to raise the door. After the door is up, another push-release cycle will lower the door.

In order to identify the inputs and outputs of the system, it's sometimes helpful to sketch its main components, as shown in the door side view to the right. The door has an up limit and a down limit switch. Each limit switch closes only when the door has reached the end of travel in the corresponding direction. In the middle of travel, neither limit switch is closed.

The motor has two command inputs: raise and lower. When neither input is active, the motor is stopped.

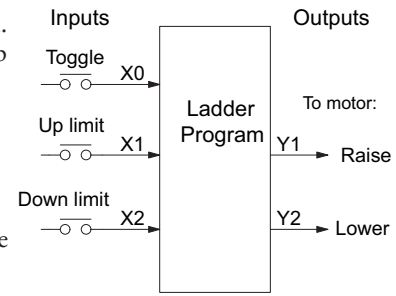
The door command is a simple pushbutton. Whether wall-mounted as shown, or a radio-remote control, all door control commands logically OR together as one pair of switch contacts.



Draw the Block Diagram

The block diagram of the controller is shown to the right. Input X0 is from the pushbutton door control. Input X1 energizes when the door reaches the full up position. Input X2 energizes when the door reaches the full down position. When the door is positioned between fully up or down, both limit switches are open.

The controller has two outputs to drive the motor. Y1 is the up (raise the door) command, and Y2 is the down (lower the door) command.

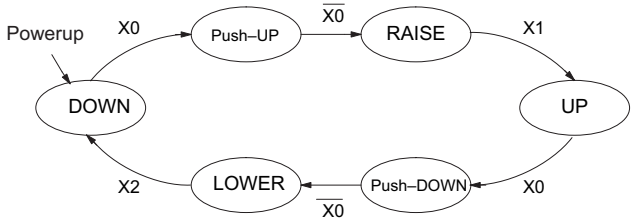


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Draw the State Diagram

Now we are ready to draw the state transition diagram. Like the previous light bulb controller example, this application also has only one switch for the command input. Refer to the figure below.

- When the door is down (DOWN state), nothing happens until X0 energizes. Its push and release brings us to the RAISE state, where output Y1 turns on and causes the motor to raise the door.
- We transition to the UP state when the up limit switch (X1) energizes, and turns off the motor.
- Then nothing happens until another X0 press-release cycle occurs. That takes us to the LOWER state, turning on output Y2 to command the motor to lower the door. We transition back to the DOWN state when the down limit switch (X2) energizes.

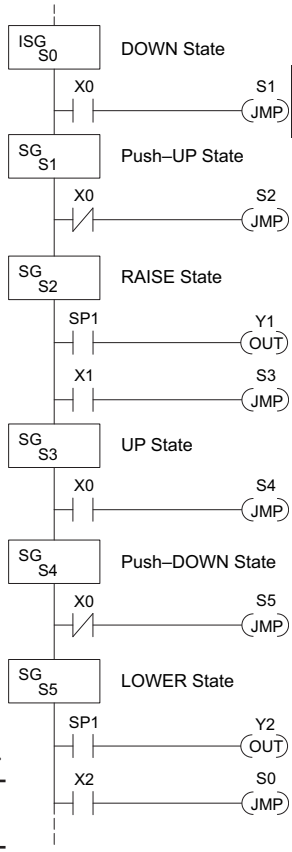


Output equations: Y1 = Raise Y2 = Lower

The equivalent stage program is shown to the right. For now, we will assume the door is down at powerup, so the desired powerup state is DOWN. We make S0 an initial stage (ISG). Stage S0 remains active until the door control pushbutton activates. Then we transition (JMP) to Push-UP stage, S1.

A push-release cycle of the pushbutton takes us through stage S1 to the RAISE stage, S2. We use the always-on contact SP1 to energize the motor's raise command, Y1. When the door reaches the fully-raised position, the up limit switch X1 activates. This takes us to the UP Stage S3, where we wait until another door control command occurs.

In the UP Stage S3, a push-release cycle of the pushbutton will take us to the LOWER Stage S5, where we activate Y2 to command the motor to lower the door. This continues until the door reaches the down limit switch, X2. When X2 closes, we transition from Stage S5 to the DOWN stage S0, where we began.





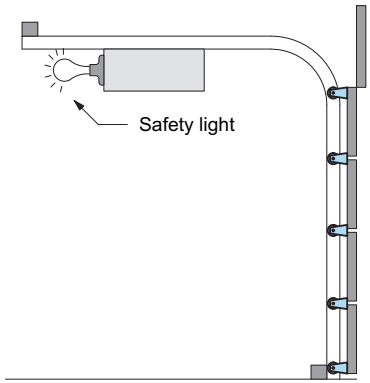
NOTE: *The only thing special about an initial stage (ISG) is that it is automatically active at powerup. Afterwards, it is just like any other.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Add Safety Light Feature

Next we will add a safety light feature to the door opener system. It's best to get the main function working first as we have done, then adding the secondary features.

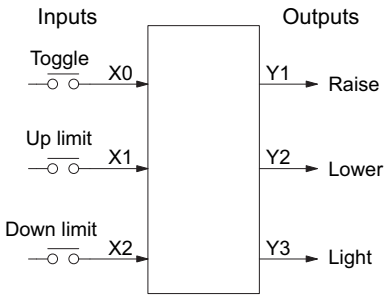
The safety light is standard on many commercially-available garage door openers. It is shown to the right, mounted on the motor housing. The light turns on upon any door activity, remaining on for approximately 3 minutes afterwards.



This part of the exercise will demonstrate the use of parallel states in our state diagram. Instead of using the JMP instruction, we will use the set and reset commands.

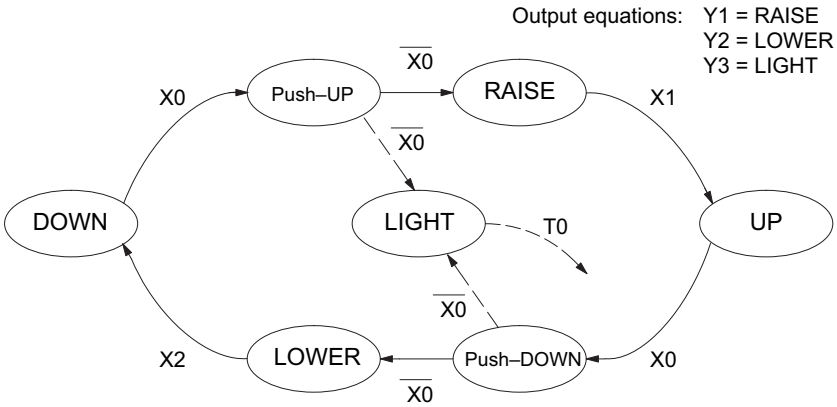
Modify the Block Diagram and State Diagram

To control the light bulb, we add an output to our controller block diagram, shown to the right, Y3 is the light control output.



In the diagram below, we add an additional state called "LIGHT". Whenever the garage owner presses the door control switch and releases, the RAISE or LOWER state is active *and the LIGHT state is simultaneously active*. The line to the Light state is dashed, because it is not the primary path.

We can think of the Light state as a parallel process to the raise and lower state. The paths to the Light state are not a transition (Stage JMP), but a State Set command. In the logic of the Light stage, we will place a three-minute timer. When it expires, timer bit T0 turns on and resets the Light stage. The path out of the Light stage goes nowhere, indicating the Light stage becomes inactive, and the light goes out!



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using a Timer Inside a Stage

The finished modified program is shown to the right. The shaded areas indicate the program additions.

In the Push-UP stage S1, we add the Set Stage Bit S6 instruction. When contact X0 opens, we transition from S1 and go to two new active states: S2 and S6. In the Push-DOWN state S4, we make the same additions. So, any time someone presses the door control pushbutton, the light turns on.

Most new stage programmers would be concerned about where to place the Light Stage in the ladder, and how to number it. The good news is that it doesn't matter!

- Choose an unused Stage number, and use it for the new stage and as the reference from other stages.
- Placement in the program is not critical, so we place it at the end.

You might think that each stage has to be directly under the stage that transitions to it. While it is good practice, it is not required (that's good, because our two locations for the Set S6 instruction make that impossible). Stage numbers and how they are used determines the transition paths.

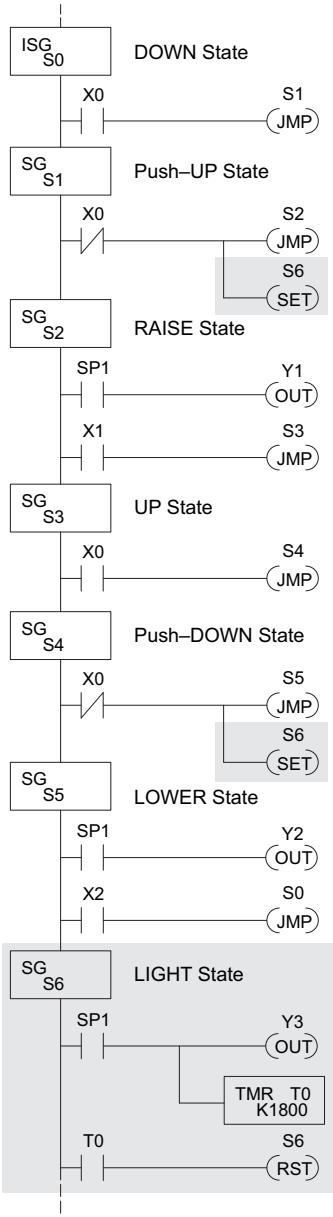
In stage S6, we turn on the safety light by energizing Y3. Special relay contact SP1 is always on. Timer T0 times at 0.1 second per count. To achieve 3 minutes time period, we calculate:

$$K = \frac{3 \text{ min. x } 60 \text{ sec/min}}{0.1 \text{ sec/count}}$$

$$K = 1800 \text{ counts}$$

The timer has power flow whenever stage S6 is active. The corresponding timer bit T0 is set when the timer expires. So three minutes later, T0=1 and the instruction Reset S6 causes the stage to be inactive.

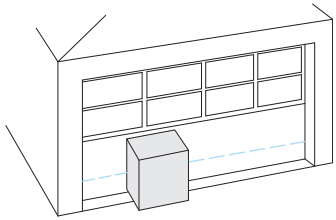
While Stage S6 is active and the light is on, stage transitions in the primary path continue normally and independently of Stage 6. That is, the door can go up, down, or whatever, but the light will be on for precisely 3 minutes.



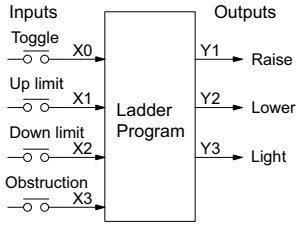
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

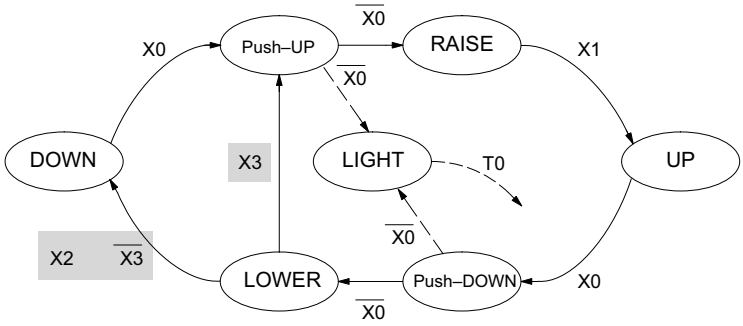
Add Emergency Stop Feature

Some garage door openers today will detect an object under the door. This halts further lowering of the door. Usually implemented with a photocell ("electric-eye"), a door in the process of being lowered will halt and begin raising. We will define our safety feature to work in this way, adding the input from the photocell to the block diagram as shown to the right. X3 will be on if an object is in the path of the door.



Next, we make a simple addition to the state transition diagram, shown in shaded areas in the figure below. Note the new transition path at the top of the LOWER state. If we are lowering the door and detect an obstruction (X3), we then jump to the Push-UP State. We do this instead of jumping directly to the RAISE state, to give the Lower output Y2 one scan to turn off, before the Raise output Y1 energizes.

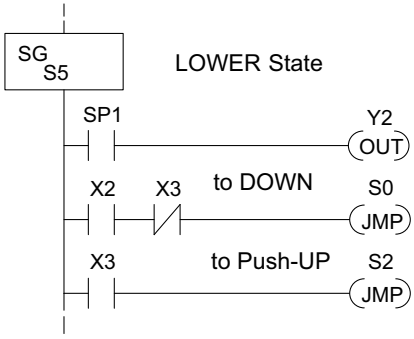




Exclusive Transitions

It is theoretically possible the down limit (X2) and the obstruction input (X3) could energize at the same moment. In that case, we would "jump" to the Push-UP and DOWN states simultaneously, which does not make sense.

Instead, we give priority to the obstruction by changing the transition condition to the DOWN state to [X2 AND NOT X3]. This ensures the obstruction event has the priority. The modifications we must make to the LOWER Stage (S5) logic are shown to the right. The first rung remains unchanged. The second and third rungs implement the transitions we need. Note the opposite relay contact usage for X3, which ensures the stage will execute only one of the JMP instructions.



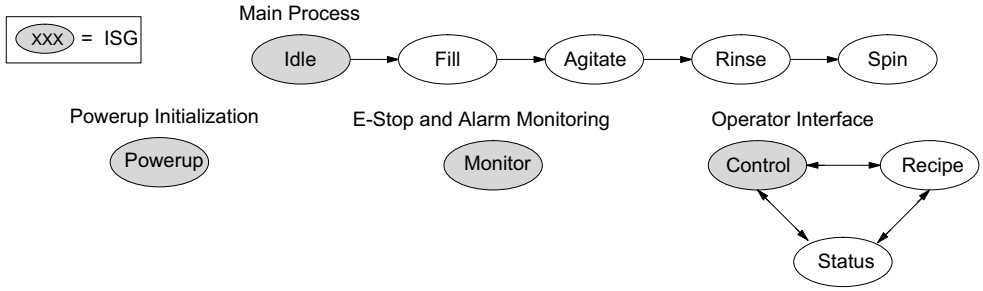
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Stage Program Design Considerations

Stage Program Organization

The examples so far in this chapter used one self-contained state diagram to represent the main process. However, we can have multiple processes implemented in stages, all in the same ladder program. New stage programmers sometimes try to turn a stage on and off each scan, based on the false assumption that only one stage can be on at a time. For ladder rungs that you want to execute each scan, put them in a stage that is always on.

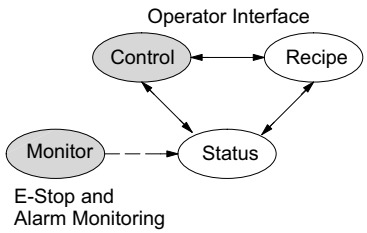
The following figure shows a typical application. During operation, the primary manufacturing activity Main Process, Powerup Initialization, E-Stop and Alarm Monitoring, and Operator Interface are all running. At powerup, four initial stages shown begin operation.



In a typical application, the separate stage sequences above operate as follows:

- **Powerup Initialization** – This stage contains ladder rung tasks performed once at powerup. Its last rung resets the stage, so this stage is only active for one scan (or only as many scans that are required)
- **Main Process** – This stage sequence controls the heart of the process or machine. One pass through the sequence represents one part cycle of the machine, or one batch in the process.
- **E-Stop and Alarm Monitoring** – This stage is always active because it is watching for errors that could indicate an alarm condition or require an emergency stop. It is common for this stage to reset stages in the main process or elsewhere, in order to initialize them after an error condition.
- **Operator Interface** – This is another task that must always be active and ready to respond to an operator. It allows an operator interface to change modes, etc., independently of the current main process step.

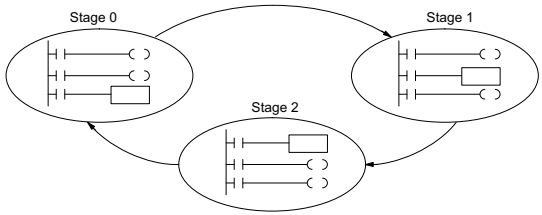
Although we have separate processes, there can be coordination among them. For example, in an error condition, the Status Stage may want to automatically switch the operator interface to the status mode to show error information as shown to the right. The monitor stage could set the stage bit for Status and Reset the stages Control and Recipe.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

How Instructions Work Inside Stages

We can think of states or stages as simply dividing up our ladder program as depicted in the figure below. Each stage contains only the ladder rungs which are needed for the corresponding state of the process. The logic for transitioning out of a stage is contained within that stage. It's easy to choose which ladder rungs are active at powerup by using an "initial" stage type (ISG).



Most instructions work like they do in standard RLL. You can think of a stage like a miniature RLL program which is either active or inactive.

Output Coils – As expected, output coils in active stages will turn on or off outputs according to power flow into the coil. However, note the following:

- Outputs work as usual, provided each output reference (such as "Y3") is used in only one stage.
- Output coils automatically turn off when leaving a stage. However, Set and Reset instructions are not "undone" when leaving a stage.
- An output can be referenced from more than one stage, as long as only one of the stages is active at a time.
- If an output coil is controlled by more than one stage simultaneously, the active stage nearest the bottom of the program determines the final output status during each scan. So, use the OROUT instruction instead when you want multiple stages to have a logical OR control of an output.

One-Shot or PD coils – Use care if you must use a Positive Differential coil in a stage. Remember the input to the coil must make a 0–1 transition. If the coil is already energized on the first scan when the stage becomes active, the PD coil will not work. This is because the 0–1 transition did not occur.

PD coil alternative: If there is a task which you want to do only once (on 1 scan), it can be placed in a stage which transitions to the next stage on the same scan.

Counter – When using a counter inside a stage, the stage must be active for one scan before the input to the counter makes a 0–1 transition. Otherwise, there is no real transition and the counter will not count. The ordinary Counter instruction does have a restriction inside stages: it may not be reset from other stages using the RST instruction for the counter bit. However, the special Stage Counter provides a solution (see next paragraph).

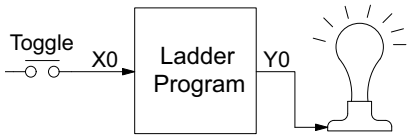
Stage Counter – The Stage Counter has the benefit that its count may be globally reset from other stages by using the RST instruction. It has a count input, but no reset input. This is the only difference from a standard counter instruction.

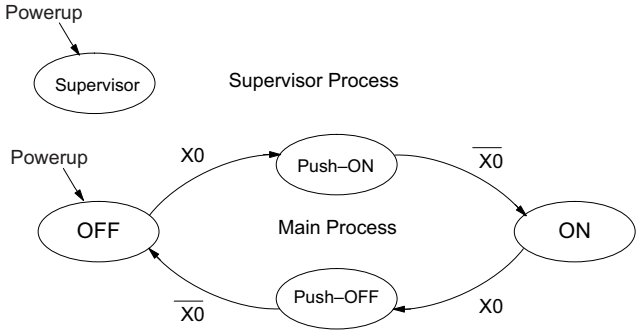
Drum – Realize the drum sequencer is its own process, and is a different programming method than stage programming. If you need to use a drum and stages, be sure to place the drum instruction in an ISG stage that is always active.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

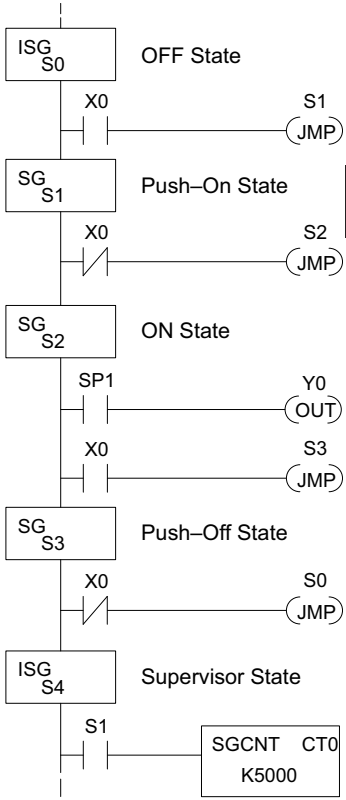
Using a Stage as a Supervisory Process

You may recall the light bulb on-off controller example from earlier in this chapter. For the purpose of illustration, suppose we want to monitor the "productivity" of the lamp process, by counting the number of on-off cycles which occurs. This application will require the addition of a simple counter, but the key decision is in where to put the counter.





New stage programming students will typically try to place the counter inside one of the stages of the process they are trying to monitor. The problem with this approach is that the stage is active only part of the time. In order for the counter to count, the count input must transition from off to on at least one scan after its stage activates. Ensuring this requires extra logic that can be tricky. In this case, we only need to add another supervisory stage as shown above, to "watch" the main process. The counter inside the supervisor stage uses the stage bit S1 of the main process as its count input. *Stage bits used as a contact let us monitor a process!*





***NOTE**: Both the Supervisor stage and the OFF stage are initial stages. The supervisor stage remains active indefinitely.*

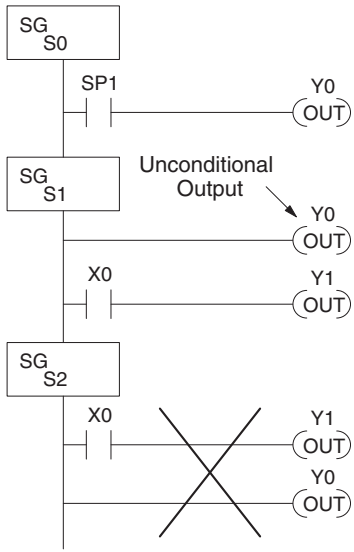
Stage Counter

The counter in the above example is a special Stage Counter. Note that it does not have a reset input. The count is reset by executing a Reset instruction, naming the counter bit (CT0 in this case). The Stage Counter has the benefit that its count may be globally reset from other stages. The standard Counter instruction does not have this global reset capability. You may still use a regular Counter instruction inside a stage... however, the reset input to the counter is the only way to reset it.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Unconditional Outputs

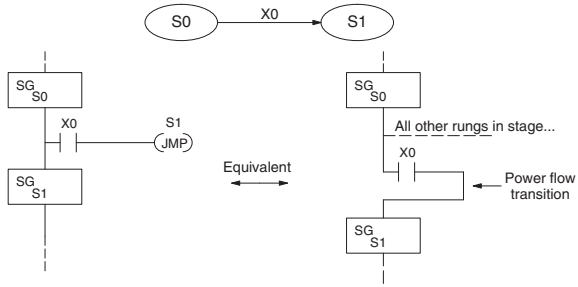
As in most example programs in this chapter and Stage 0 to the right, your application may require a particular output to be ON unconditionally when a particular stage is active. Until now, the examples always use the SP1 special relay contact (always on) in series with the output coils. It's possible to omit the contact, as long as you place any unconditional outputs first (at the top) of a stage section of ladder. The first rung of Stage 1 does this.



WARNING: Unconditional outputs placed elsewhere in a stage do not necessarily remain on when the stage is active. In Stage 2 to the right, Y0 is shown as an unconditional output, but its powerflow comes from the rung above. So, Y0 status will be the same as Y1 is not correct.

Power Flow Transition Technique

Our discussion of state transitions has shown how the Stage JMP instruction makes the current stage inactive and the next stage (named in the JMP) active. As an alternative way to enter this in *Direct*SOFT, you may use the power flow method for stage transitions. The main requirement is the current stage be located directly above the next (jump-to) stage in the ladder program. This arrangement is shown in the diagram below, by stages S0 and S1, respectively.



Recall the Stage JMP instruction may occur anywhere in the current stage, and the result is the same. However, power flow transitions (shown above) must occur as the last rung in a stage. All other rungs in the stage will precede it. The power flow transition method is also achievable on the handheld programmer, by simply following the transition condition with the Stage instruction for the next stage.

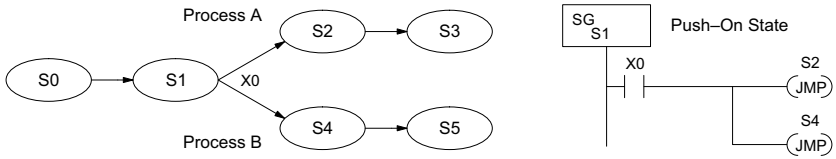
The power flow transition method does eliminate one Stage JMP instruction, its only advantage. However, it is not as easy to make program changes as using the Stage JMP. Therefore, we advise using Stage JMP transitions for most programs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Parallel Processing Concepts

Parallel Processes

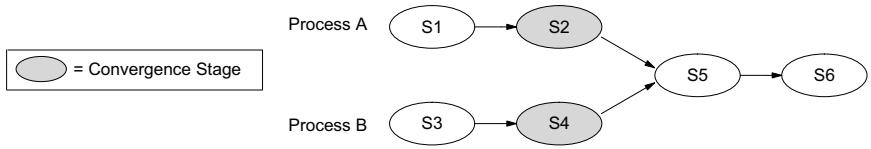
Previously in this chapter we discussed how a state may transition to either one state or another, called an exclusive transition. In other cases, we may need to branch simultaneously to two or more parallel processes, as shown below. It is acceptable to use all JMP instructions as shown, or we could use one JMP and a Set Stage bit instruction(s) (at least one must be a JMP, in order to leave S1). Remember that all instructions in a stage execute, even when it transitions (the JMP is not a GOTO).



Note that if we want Stages S2 and S4 to energize exactly on the same scan, both stages must be located below Stage S1 in the ladder program (see the explanation at the bottom of page 7–7). Overall, parallel branching is easy!

Converging Processes

Now we consider the opposite case of parallel branching, which is *converging processes*. This simply means we stop doing multiple things and continue doing one thing at a time. In the figure below, processes A and B converge when stages S2 and S4 transition to S5 at some point in time. So, S2 and S4 are *Convergence Stages.*

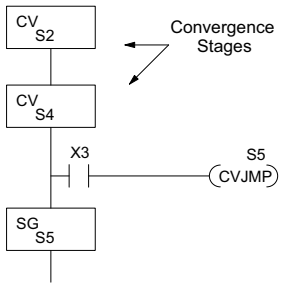


Convergence Stages (CV)

230
240
250-1
260

While the converging principle is simple enough, it brings a new complication. As parallel processing completes, the multiple processes almost never finish at the same time. In other words, how can we know whether Stage S2 or S4 will finish last? This is an important point, because we have to decide how to transition to Stage S5.

The solution is to coordinate the transition condition out of convergence stages. We accomplish this with a stage type designed for this purpose: the Convergence Stage (type CV). In the example to the right, convergence stages S2 and S4 are required to be grouped together as shown. **No logic is permitted between CV stages!** The transition condition (X3 in this case) must be located in the last convergence stage. The transition condition only has power flow when all convergence stages in the group are active.

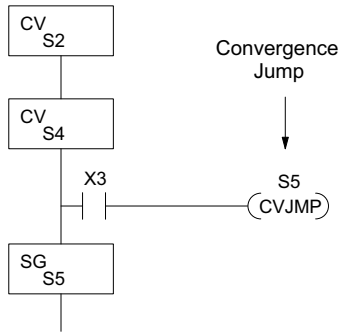


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Convergence Jump (CVJMP)

230
240
250-1
260

Recall the last convergence stage only has power flow when all CV stages in the group are active. To complement the convergence stage, we need a new jump instruction. The Convergence Jump (CVJMP) shown to the right will transition to Stage S5 when X3 is active (as one might expect), but it also *automatically resets all convergence stages in the group*. This makes the CVJMP jump a very powerful instruction. Note that this instruction may only be used with convergence stages.



Convergence Stage Guidelines

The following summarizes the requirements in the use of convergence stages, including some tips for their effective application:

- A convergence stage is to be used as the last stage of a process which is running in parallel to another process or processes. A transition to the convergence stage means that a particular process is through, and represents a waiting point until all other parallel processes also finish.
- The maximum number of convergence stages which make up one group is 17. In other words, a maximum of 17 stages can converge into one stage.
- Convergence stages of the same group must be placed together in the program, connected on the power rail without any other logic in between.
- Within a convergence group, the stages may occur in any order, top to bottom. It does not matter which stage is last in the group, because all convergence stages have to be active before the last stage has power flow.
- The last convergence stage of a group may have ladder logic within the stage. However, this logic will not execute until all convergence stages of the group are active.
- The convergence jump (CVJMP) is the intended method to be used to transition from the convergence group of stages to the next stage. The CVJMP resets all convergence stages of the group, and energizes the stage named in the jump.
- The CVJMP instruction must only be used in a convergence stage, as it is invalid in regular or initial stages.
- Convergence Stages or CVJMP instructions may not be used in subroutines or interrupt routines.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

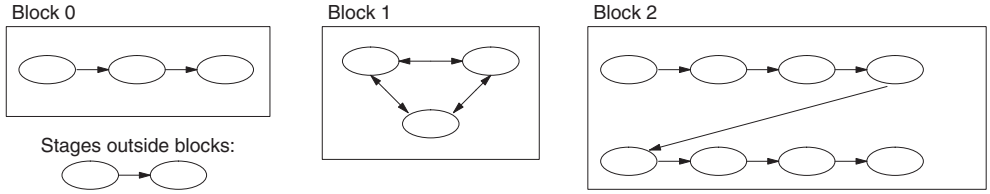
Managing Large Programs

A stage may contain a lot of ladder rungs, or only one or two program rungs. For most applications, good program design will ensure the average number of rungs per stage will be small. However, large application programs will still create a large number of stages. We introduce a new construct which will help us organize related stages into groups called blocks. So, program organization is the main benefit of the use of stage blocks.

Stage Blocks (BLK, BEND)

☒ 230
☑ 240
☑ 250-1
☑ 260

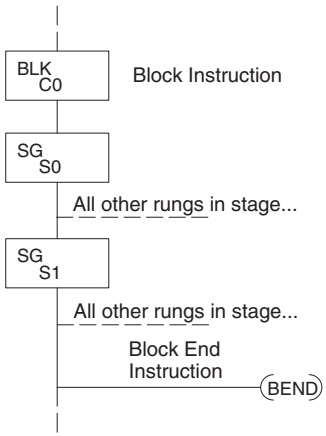
A block is a section of ladder program which contains stages. In the figure below, each block has its own reference number. Like stages, a stage block may be active or inactive. Stages inside a block are not limited in how they may transition from one to another. Note the use of stage blocks does not require each stage in a program to reside inside a block, shown below by the "stages outside blocks".



A program with 20 or more stages may be considered large enough to use block grouping (however, their use is not mandatory). When used, the number of stage blocks should probably be two or higher, because the use of one block provides a negligible advantage.

A block of stages is separated from other ladder logic with special beginning and ending instructions. In the figure to the right, the BLK instruction at the top marks the start of the stage block. At the bottom, the Block End (BEND) marks the end of the block. The stages in between these boundary markers (S0 and S1 in this case) and their associated rungs make up the block.

Note the block instruction has a reference value field (set to "C0" in the example). The block instruction borrows or uses a control relay contact number, so that other parts of the program can control the block. *Any control relay number (such as C0) used in a BLK instruction is not available for use as a control relay.*





***NOTE**: The stages within a block must be regular stages (SG) or convergence stages (CV). So, they cannot be initial stages. The numbering of stages inside stage blocks can be in any order, and is completely independent from the numbering of the blocks.*

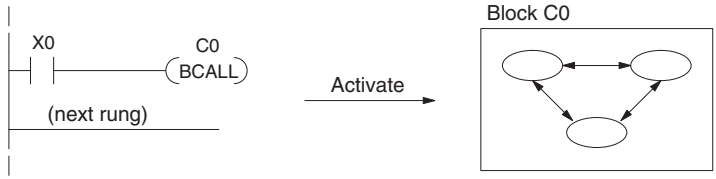
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Block Call (BCALL)

☒ 230
☑ 240
☑ 250-1
☑ 260

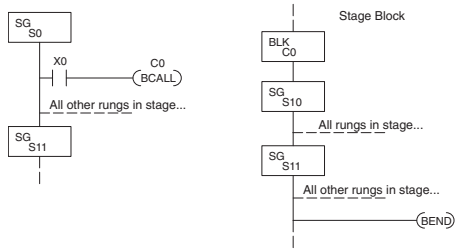
The purpose of the Block Call instruction is to activate a stage block. At powerup or upon Program-to-Run mode transitions, all stage blocks and the stages within them are inactive. Shown in the figure below, the Block Call instruction is a type of output coil. When the X0 contact is closed, the BCALL will cause the stage block referenced in the instruction (C0) to become active. When the BCALL is turned off, the corresponding stage block and the stages within it become inactive.

We must avoid confusing block call operation with how a "subroutine call" works. After a BCALL coil executes, program execution continues with the next program rung. Whenever program execution arrives at the ladder location of the stage block named in the BCALL, then logic within the block executes because the block is now active. Similarly, do not classify the BCALL as type of state transition (is not a JMP).



When a stage block becomes active, the first stage in the block automatically becomes active on the same scan. The "first" stage in a block is the one located immediately under the block (BLK) instruction in the ladder program. So, that stage plays a similar role to the initial type stage we discussed earlier.

The Block Call instruction may be used in several contexts. Obviously, the first execution of a BCALL must occur outside a stage block, since stage blocks are initially inactive. Still, the BCALL may occur on an ordinary ladder rung, or it may occur within an active stage as shown below. Note that either turning off the BCALL or turning off the stage containing the BCALL will deactivate the corresponding stage block. You may also control a stage block with a BCALL in another stage block.



***NOTE**: Stage Block may come before or after the location of the BCALL instruction in the program.*

The BCALL may be used in many ways or contexts, so it can be difficult to find the best usage. Remember the purpose of stage blocks is to help you organize the application problem by grouping related stages together. Remember that initial stages must exist *outside* stage blocks.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RLL*PLUS* (Stage) Instructions

Stage (SG)

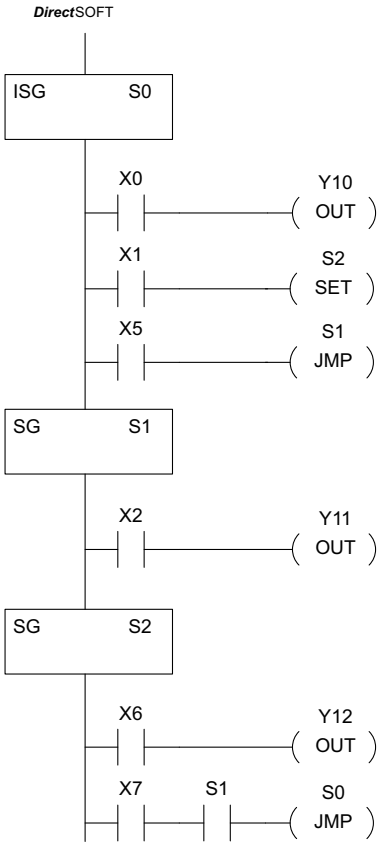
☑ 230
☑ 240
☑ 250-1
☑ 260

The Stage instructions are used to create structured RLL*PLUS* programs. Stages are program segments which can be activated by transitional logic, a jump or a set stage that is executed from an active stage. Stages are deactivated one scan one after transitional logic, a jump, or a reset stage instruction is executed.



Operand Data Type	DL230 Range	DL240 Range	DL250-1 Range	DL260 Range
	aaa	aaa	aaa	aaa
Stage S	0-377	0-777	0-1777	0-1777

The following example is a simple RLL*PLUS* program. This program utilizes the initial stage, stage, and jump instruction to create a structured program.



Handheld Programmer Keystrokes

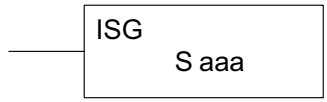
ISG	→	S(SG)	0	ENT	
STR	→	X(IN)	0	ENT	
OUT	→	Y(OUT)	1	0	ENT
STR	→	X(IN)	1	ENT	
SET	→	S(SG)	2	ENT	
STR	→	X(IN)	5	ENT	
JMP	→	S(SG)	1	ENT	
SG	→	S(SG)	1	ENT	
STR	→	X(IN)	2	ENT	
OUT	→	Y(OUT)	1	1	ENT
SG	→	S(SG)	2	ENT	
STR	→	X(IN)	6	ENT	
OUT	→	Y(OUT)	1	2	ENT
STR	→	X(IN)	7	ENT	
AND	→	S(SG)	1	ENT	
JMP	→	S(SG)	0	ENT	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Initial Stage (ISG)

☑ 230
☑ 240
☑ 250-1
☑ 260

The Initial Stage instruction is normally used as the first segment of an RLL*PLUS* program. Initial stages will be active when the CPU enters the run mode allowing for a starting point in the program. Initial Stages are also activated by transitional logic, a jump or a set stage executed from an active stage. Initial Stages are deactivated one scan after transitional logic, a jump, or a reset stage instruction is executed. Multiple Initial Stages are allowed in a program.



Operand Data Type		DL230 Range	DL240 Range	DL250-1 Range	DL260 Range
		aaa	aaa	aaa	aaa
Stage	S	0-377	0-777	0-1777	0-1777



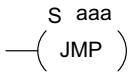
NOTE: *If the ISG is within the retentive range for stages, the ISG will remain in the state it was in before power down and will NOT turn itself on during the first scan.*

7

Jump (JMP)

☑ 230
☑ 240
☑ 250-1
☑ 260

The Jump instruction allows the program to transition from an active stage which contains the jump instruction to another stage which is specified in the instruction. The jump will occur when the input logic is true. The active stage that contains the Jump will be deactivated 1 scan after the Jump instruction is executed.

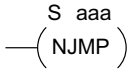


Operand Data Type		DL230 Range	DL240 Range	DL250-1 Range	DL260 Range
		aaa	aaa	aaa	aaa
Stage	S	0-377	0-777	0-1777	0-1777

Not Jump (NJMP)

☑ 230
☑ 240
☑ 250-1
☑ 260

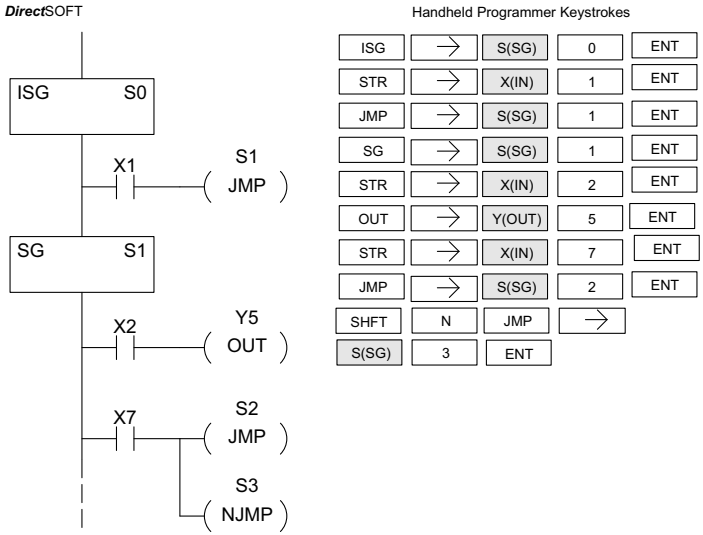
The Not Jump instruction allows the program to transition from an active stage which contains the jump instruction to another which is specified in the instruction. The jump will occur when the input logic is off. The active stage that contains the Not Jump will be deactivated 1 scan after the Not Jump instruction is executed.



Operand Data Type		DL230 Range	DL240 Range	DL250-1 Range	DL260 Range
		aaa	aaa	aaa	aaa
Stage	S	0-377	0-777	0-1777	0-1777

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

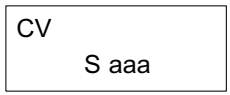
In the following example, when the CPU begins program execution only ISG 0 will be active. When X1 is on, the program execution will jump from Initial Stage 0 to Stage 1. In Stage 1, if X2 is on, output Y5 will be turned on. If X7 is on, program execution will jump from Stage 1 to Stage 2. If X7 is off, program execution will jump from Stage 1 to Stage 3.



Converge Stage (CV) and Converge Jump (CVJMP)

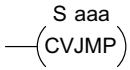
☒ 230
☑ 240
☑ 250-1
☑ 260

The Converge Stage instruction is used to group certain stages together by defining them as Converge Stages.



When all of the Converge Stages within a group become active, the CVJMP instruction (and any additional logic in the final CV stage) will be executed. All preceding CV stages *must* be active before the final CV stage logic can be executed. All Converge Stages are deactivated one scan after the CVJMP instruction is executed.

Additional logic instructions are only allowed following the last Converge Stage instruction and before the CVJMP instruction. Multiple CVJUMP instructions are allowed.



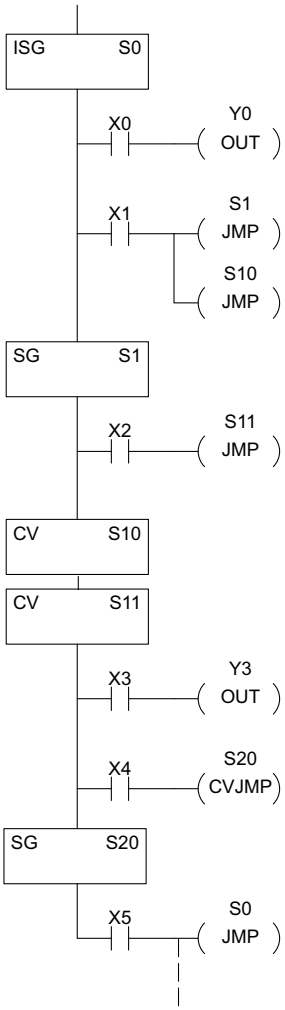
Converge Stages must be programmed in the main body of the application program. This means they cannot be programmed in Subroutines or Interrupt Routines.

Operand Data Type		DL240 Range	DL250-1 Range	DL260 Range
		aaa	aaa	aaa
Stage	S	0-777	0-1777	0-1777

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In the following example, when Converge Stages S10 and S11 are *both* active the CVJMP instruction will be executed when X4 is on. The CVJMP will deactivate S10 and S11, and activate S20. Then, if X5 is on, the program execution will jump back to the initial stage, S0.

DirectSOFT



Handheld Programmer Keystrokes

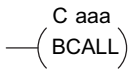
ISG	→	S(SG)	0	ENT				
STR	→	X(IN)	0	ENT				
OUT	→	Y(OUT)	0	ENT				
STR	→	X(IN)	1	ENT				
JMP	→	S(SG)	1	ENT				
JMP	→	S(SG)	1	0	ENT			
SG	→	S(SG)	1	ENT				
STR	→	X(IN)	2	ENT				
JMP	→	S(SG)	1	1	ENT			
SHFT	C	V	→	S(SG)	1	0	ENT	
SHFT	C	V	→	S(SG)	1	1	ENT	
STR	→	X(IN)	3	ENT				
OUT	→	Y(OUT)	3	ENT				
STR	→	X(IN)	4	ENT				
SHFT	C	V	SHFT	JMP	S(SG)	2	0	ENT
SG	→	S(SG)	2	0	ENT			
STR	→	X(IN)	5	ENT				
JMP	→	S(SG)	0	ENT				

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Block Call (BCALL)

☒ 230
☑ 240
☑ 250-1
☑ 260

The stage block instructions are used to activate a block of stages. The Block Call, Block, and Block End instructions must be used together. The BCALL instruction is used to activate a stage block. There are several things you need to know about the BCALL instruction.



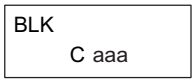
- Uses CR Numbers — The BCALL appears as an output coil, but does not actually refer to a Stage number as you might think. Instead, the block is identified with a Control Relay (Caaa). This control relay cannot be used as an output anywhere else in the program.
- Must Remain Active — The BCALL instruction actually controls all the stages between the BLK and the BEND instructions even after the stages inside the block have started executing. The BCALL must *remain active* or all the stages in the block will automatically be turned off. *If either the BCALL instruction, or the stage that contains the BCALL instruction goes off, then the stages in the defined block will be turned off automatically.*
- Activates First Block Stage — When the BCALL is executed it automatically activates the first stage following the BLK instructions.

Operand Data Type	DL240 Range	DL250-1 Range	DL260 Range
	aaa	aaa	aaa
Control Relay C	0-777	0-1777	0-3777

Block (BLK)

☒ 230
☑ 240
☑ 250-1
☑ 260

The Block instruction is a label which marks the beginning of a block of stages that can be activated as a group. A Stage instruction must immediately follow the Start Block instruction. Initial Stage instructions are not allowed in a block. The control relay (Caaa) specified in Block instruction must not be used as an output anywhere else in the program.



Block End (BEND)

☒ 230
☑ 240
☑ 250-1
☑ 260

The Block End instruction is a label used with the Block instruction. It marks the end of a block of stages. There is no operand with this instruction. Only one Block End is allowed per Block Call.



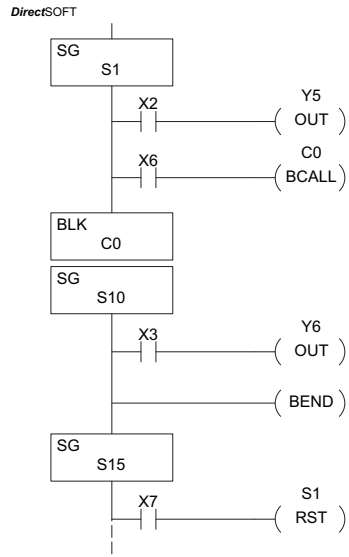
Operand Data Type	DL240 Range	DL250-1 Range	DL260 Range
	aaa	aaa	aaa
Control Relay C	0-777	0-1777	0-3777

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In this example, the Block Call is executed when stage 1 is active and X6 is on. The Block Call then automatically activates stage S10, which immediately follows the Block instruction.

This allows the stages between S10 and the Block End instruction to operate as programmed. If the BCALL instruction is turned off, or if the stage containing the BCALL instruction is turned off, then all stages between the BLK and BEND instructions are automatically turned off.

If you examine S15, you will notice that X7 could reset Stage S1, which would disable the BCALL, thus resetting all stages within the block.

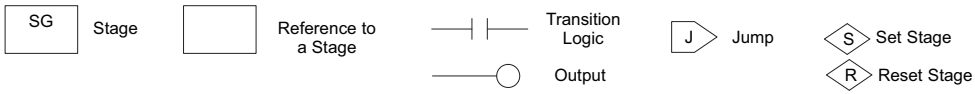


Handheld Programmer Keystrokes

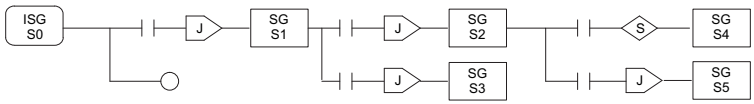
SG	→	S(SG)	1	ENT					
STR	→	X(IN)	2	ENT					
OUT	→	Y(OUT)	5	ENT					
STR	→	X(IN)	6	ENT					
SHFT	B	C	A	L	L	→	C(CR)	0	ENT
SHFT	B	L	K	→	C(CR)	0	ENT		
SG	→	S(SG)	1	0	ENT				
STR	→	X(IN)	3	ENT					
OUT	→	Y(OUT)	6	ENT					
SHFT	B	E	N	D	ENT				
SG	→	S(SG)	1	5	ENT				
STR	→	X(IN)	7	ENT					
RST	→	S(SG)	1	ENT					

Stage View in *Direct*SOFT

The Stage View option in *Direct*SOFT will let you view the ladder program as a flow chart. The figure below shows the symbol convention used in the diagrams. You may find the stage view useful as a tool to verify that your stage program has faithfully reproduced the logic of the state transition diagram you intend to realize.



The following diagram is a typical stage view of a ladder program containing stages. Note the left-to-right direction of the flow chart.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Questions and Answers about Stage Programming

We include the following commonly-asked questions about Stage Programming as an aid to new students. All question topics are covered in more detail in this chapter.

Q. What does stage programming do that I can't do with regular RLL programs?

A. Stages allow you to identify all the states of your process before you begin programming. This approach is more organized, because you divide up a ladder program into sections. As stages, these program sections are active only when they are actually needed by the process. Most processes can be organized into a sequence of stages, connected by event-based transitions.

Q. Isn't a stage really like a software subroutine?

A. No, it is very different. A subroutine is called by a main program when needed, and executes only once before returning to the point from which it was called. A stage, however, is part of the main program. It represents a state of the process, so an active stage executes on every scan of the CPU until it becomes inactive.

Q. What are Stage Bits?

A. A stage bit is a single bit in the CPU's image register, representing the active/inactive status of the stage in real time. For example, the bit for Stage 0 is referenced as "S0". If S0 = 0, then the ladder rungs in Stage 0 are bypassed (not executed) on each CPU scan. If S0 = 1, then the ladder rungs in Stage 0 are executed on each CPU scan. Stage bits, when used as contacts, allow one part of your program to monitor another part by detecting stage active/inactive status.

Q. How does a stage become active?

A. There are three ways:

- If the Stage is an initial stage (ISG), it is automatically active at powerup.
- Another stage can execute a Stage JMP instruction naming this stage, which makes it active upon its next occurrence in the program.
- A program rung can execute a Set Stage Bit instruction (such as SET S0).

Q. How does a stage become inactive?

A. There are three ways:

- Standard Stages (SG) are automatically inactive at powerup.
- A stage can execute a Stage JMP instruction, resetting its Stage Bit to 0.
- Any rung in the program can execute a Reset Stage Bit instruction (such as RST S0).

Q. What about the power flow technique of stage transitions?

A. The power flow method of connecting adjacent stages (directly above or below in the program) actually is the same as the Stage Jump instruction executed in the stage above, naming the stage below. Power flow transitions are more difficult to edit in *Direct*SOFT, we list them separately from two preceding questions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Q. Can I have a stage which is active for only one scan?

A. Yes, but this is not the intended use for a stage. Instead, make a ladder rung active for one scan by including a stage Jump instruction at the bottom of the rung. Then the ladder will execute on the last scan before its stage jumps to a new one.

Q. Isn't a Stage JMP just like a regular GOTO instruction used in software?

A. No, it is very different. A GOTO instruction sends the program execution immediately to the code location named by the GOTO. A Stage JMP simply resets the Stage Bit of the current stage, while setting the Stage Bit of the stage named in the JMP instruction. Stage bits are 0 or 1, determining the inactive/active status of the corresponding stages. A stage JMP has the following results:

- When the JMP is executed, the remainder of the current stage's rungs are executed, even if they reside past(under) the JMP instruction. On the following scan, that stage is not executed, because it is inactive.
- The Stage named in the Stage JMP instruction will be executed upon its next occurrence. If located past (under) the current stage, it will be executed on the same scan. If located before (above) the current stage, it will be executed on the following scan.

Q. How can I know when to use stage JMP, versus a Set Stage Bit or Reset Stage Bit?

A. These instructions are used according to the state diagram topology you have derived:

- Use a Stage JMP instruction for a state transition... moving from one state to another.
- Use a Set Stage Bit instruction when the current state is spawning a new parallel state or stage sequence, or when a supervisory state is starting a state sequence under its command.
- Use a Reset Bit instruction when the current state is the last state in a sequence and its task is complete, or when a supervisory state is ending a state sequence under its command.

Q. What is an initial stage, and when do I use it?

A. An initial stage (ISG) is automatically active at powerup. Afterwards, it works just like any other stage. You can have multiple initial stages, if required. Use an initial stage for ladder that must always be active, or as a starting point.

Q. Can I have place program ladder rungs outside of the stages, so they are always on?

A. It is possible, but it's not good software design practice. Place ladder that must always be active in an initial stage, and do not reset that stage or use a Stage JMP instruction inside it. It can start other stage sequences at the proper time by setting the appropriate Stage Bit(s).

Q. Can I have more than one active stage at a time?

A. Yes, and this is a normal occurrence for many programs. However, it is important to organize your application into separate processes, each made up of stages. And a good process design will be mostly sequential, with only one stage on at a time. However, all the processes in the program may be active simultaneously.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Loop Operation



In This Chapter...

DL250-1 and DL260 PID Loop Features 8–2
Introduction to PID Control 8–4
Introducing DL205 PID Control 8–6
PID Loop Operation 8–9
Ten Steps to Successful Process Control 8–16
PID Loop Setup 8–18
PID Loop Tuning 8–41
Using the Special PID Features 8–52
Ramp/Soak Generator 8–57
DirectSOFT Ramp/Soak Example 8–62
Cascade Control 8–64
Time-Proportioning Control 8–67
Feedforward Control 8–69
PID Example Program 8–71
Troubleshooting Tips 8–74
Glossary of PID Loop Terminology 8–76
Bibliography 8–78

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

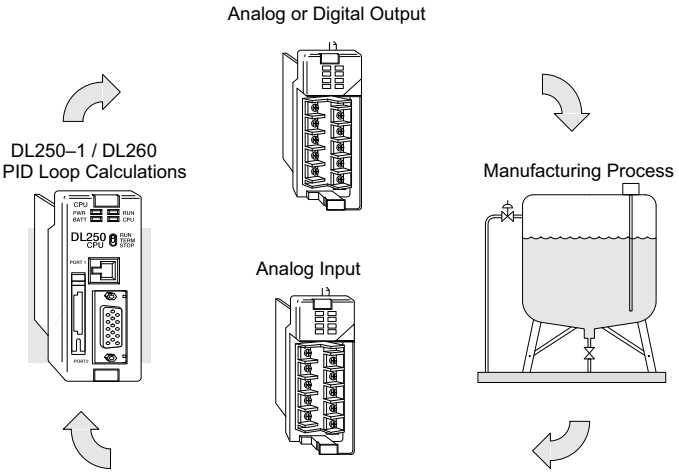
DL250-1 and DL260 PID Loop Features

Main Features

The DL250–1 and DL260 CPUs process loop control offers a sophisticated set of features to address many application needs. The main features are:

- DL250–1 – up to 4 loops, individual programmable sample rates
- DL260 – up to 16 loops, individual programmable sample rates
- Manual/ Automatic/Cascaded loop capability available
- Two types of bumpless transfer available
- Full-featured alarms
- Ramp/soak generator with up to 16 segments
- Auto Tuning

The DL250–1 and DL260 CPUs have process control loop capability in addition to ladder program execution. You can select and configure up to four loops for the DL250-1 or sixteen loops for the DL260. All sensor and actuator wiring connects to standard DL205 I/O modules, as shown below. All process variables, gain values, alarm levels, etc., associated with each loop reside in a Loop Variable Table in the CPU. The CPU reads process variable (PV) inputs during each scan. Then it makes PID loop calculations during a dedicated time slice on each PLC scan, updating the control output value. The control loops use the Proportional-Integral-Derivative (PID) algorithm to generate the control output command. This chapter describes how the loops operate, and what you must do to configure and tune the loops.



The best tool for configuring loops in the CPU is the ***Direct***SOFT programming software, Release 2.1 or later. ***Direct***SOFT uses dialog boxes to create a forms-like editor to let you individually set up the loops. After completing the setup, you can use ***Direct***SOFT's PID Trend View to tune each loop. The starting address and number of loops is stored in system memory, but tuning parameters are stored in V-memory. The loop parameters also may be saved to disk for recall later.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Loop Feature	Specifications
Number of loops	DL250-1 - selectable up to 4; DL260 - selectable up to 16
CPU V-memory needed	32 words (V locations) per loop selected, 64 words if using ramp/soak
PID algorithm	Position or Velocity form of the PID equation
Control Output polarity	Selectable direct-acting or reverse-acting
Error term curves	Selectable as linear, square root of error, and error squared
Loop update rate (time between PID calculation)	0.05 to 99.99 seconds, user programmable
Minimum loop update rate	0.05 seconds for 1 to 4 loops (DL250-1/260) 0.1 seconds for 5 to 8 loops (DL260) 0.2 seconds for 9 to 16 loops (DL260)
Loop modes	Automatic, Manual (operator control), or Cascade control
Ramp/Soak Generator	Up to 8 ramp/soak steps (16 segments) per loop with indication of ramp/soak step number
PV curves	Select standard linear, or square-root extract (for flow meter input)
Set Point Limits	Specify minimum and maximum setpoint values
Process Variable Limits	Specify minimum and maximum Process Variable values
Proportional Gain	Specify gains of 0.01 to 99.99
Integrator (Reset)	Specify reset time of 0.01 to 99.98 in units of seconds or minutes
Derivative (Rate)	Specify the derivative time from 0.01 to 99.99 seconds
Rate Limits	Specify derivative gain limiting from 1 to 20
Bumpless Transfer I	Automatically sets the bias equal to the control output and the setpoint equal to the process variable when control switches from manual to automatic.
Bumpless Transfer II	Automatically sets the bias equal to the control output when control switches from manual to automatic
Step Bias	Provides proportional bias adjustment for large setpoint changes
Anti-windup (Freeze Bias)	For position form of PID, this inhibits integrator action when the control output reaches 0% or 100% (speeds up loop recovery when output recovers from saturation)
Error Deadband	Specify a tolerance (plus and minus) for the error term (SP-PV), so that no change in control output value is made

Alarm Feature	Specifications
PV Alarm Hysteresis	Specify 1 to 200 (word/binary) does not affect all alarms, such as PV Rate-of-Change Alarm
PV Alarm Points	Select PV alarm settings for Low-low, Low, High, and High-high conditions
PV Deviation	Specify alarms for two ranges of PV deviation from the setpoint value
Rate of Change	Detect when PV exceeds a rate of change limit you specify

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction to PID Control

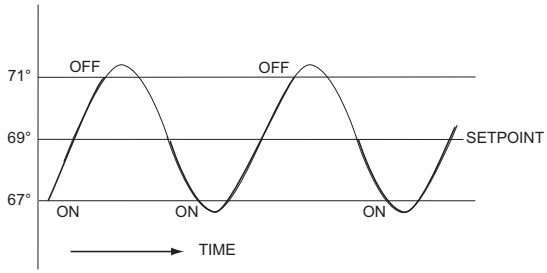
Why use PID Control?

There are many analog control strategies available for industrial processes, and the best one to choose depends on the particular application. Let's compare the two most common types of analog control used throughout industry:

1. The ON-OFF controller, sometimes referred to as an open loop controller.
2. The PID controller, sometimes called a closed loop controller.

Regardless of type, analog controllers require input signals from electronic sensors such as pressure, differential pressure, level, flow meter or thermocouples. As an example, one of the most common analog control applications is located in your house for controlling either heat or air conditioning, the thermostat.

You wish for your house to be at a comfortable temperature so you set a thermostat to a desired temperature (setpoint). You then select the "comfort" mode, either heat or A/C. A temperature sensing device, normally a thermistor, is located within the thermostat. If the thermostat is set for heat and the setpoint is set for 69°, the furnace will be turned on to provide heat at, normally, 2° below the setpoint. In this case, it would turn on at 67°. When the temperature reaches 71°, 2° above setpoint, the furnace will turn off. In the opposite example, if the thermostat is set for A/C (cooling), the thermostat will turn the A/C unit on/off opposite the heat setting. For instance, if the thermostat is set to cool at 76°, the A/C unit will turn on when the sensed temperature reaches 2° above the setpoint or 78°, and turn off when the temperature reaches 74°. This would be considered to be an ON-OFF controller. The waveform below shows the action of the heating cycle. Note that there is a slight overshoot at the turn-off point, also a slight undershoot at the turn-on point.



The ON-OFF controller is used in some industrial control applications, but is not practical in the majority of industrial control processes.

The most common process controller that is used in industry is the PID controller.

The PID controller controls a continuous feedback loop that keeps the process output (control variable) flowing normally by taking corrective action whenever there is a deviation from the desired value (setpoint) of the process variable (PV) such as, rate of flow, temperature, voltage, etc. An "error" occurs when an operator manually changes the setpoint or when an event (valve opened, closed, etc.) or a disturbance (cold water, wind, etc.) changes the load, thus causing a change in the process variable.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

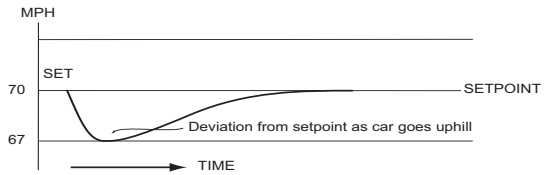
The PID controller receives signals from sensors and computes corrective action to the actuator from a computation based on the error (Proportional), the sum of all previous errors (Integral) and the rate of change of the error (Derivative).

We can look at the PID controller in more simple terms. Take the cruise control on an automobile as an example. Let's say that we are cruising on an interstate highway in a car equipped with cruise control. The driver decides to engage the cruise control by turning it ON, then he manually brings the car to the desired cruising speed, say 70 miles per hour. Once the cruise speed is reached, the SET button is pushed fixing the speed at 70 mph, the setpoint. Now, the car is cruising at a steady 70 mph until it comes to a hill to go up. As the car goes up the hill, it tends to slow down. The speed sensor senses this and causes the throttle to increase the fuel to the engine. The vehicle speeds up to maintain 70 mph without jerking the car and it reaches the top at the set speed. When the car levels out after reaching the top of the hill it will speed up. The speed sensor senses this and signals the throttle to provide less fuel to the engine, thus, the engine slows down allowing the car to maintain the 70 mph speed. How does this application apply to PID control? Lets look at the function of P, I and D terms:

- **Proportional** - is commonly referred to as Proportional Gain. The proportional term is the corrective action which is proportional to the error, that is, the change of the manipulated variable is equal to the proportional gain multiplied by the error (the activating signal). In mathematical terms:

  Proportional action = proportional gain X error
  Error = Setpoint (SP) - Process Variable (PV)

  Applying this to the cruise control, the speed was set at 70 mph which is the Setpoint. The speed sensor senses the actual speed of the car and sends this signal to the cruise



  controller as the Process Variable (PV). When the car is on a level highway, the speed is maintained at 70 mph, thus, no error since the error would be SP - PV = 0. When the car goes up the hill, the speed sensor detected a slow down of the car, SP-PV = error. The proportional gain would cause the output of the speed controller to bring the car back to the setpoint of 70 mph. This would be the Controlled Output.

- **Integral** - this term is often referred to as Reset action. It provides additional compensation to the control output, which causes a change in proportion to the value of the error over a period of time. In other words, the reset term is the integral sum of the error values over a period of time.
- **Derivative** - this term is referred to as rate. The Rate action adds compensation to the control output, which causes a change in proportion to the rate of change of error. Its job is to anticipate the probable growth of the error and generate a contribution to the output in advance.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introducing DL205 PID Control

The DL205 (DL250-1 and DL260) is capable of controlling a process where PID control is required. As previously mentioned, the control of a variable, such as temperature, at a given level (setpoint) as long as there are no disturbances (such as cold water entering a process that heats water) in the process.

The DL205 PLC has the ability to directly accept signals from electronic sensors, such as thermocouples, pressure, VFDs, etc. These signals may be used in mathematically derived control systems.

In addition, the DL205 has built-in PID control algorithms that can be implemented. The basic function of PID closed loop process control is to maintain certain process characteristics at desired setpoints. As a rule, the process deviates from the desired setpoint reference as a result of load material changes and interaction with other processes. During this control, the actual condition of the process characteristics (liquid level, temperature, motor control, etc.) is measured as a *process variable* (PV) and compared with the target setpoint (SP). When deviations occur, an error is generated by the difference between the process variable (actual value) and the setpoint (desired value). Once an error is detected, the function of the control loop is to modify the control output in order to force the error to zero.

The DL205 PID control provides feedback loops using the PID algorithm. The control output is computed from the measured process variable as follows:

Let:

K_c = proportional gain

T_i = Reset or integral time

T_d = Derivative time or rate

SP = Setpoint

PV(t) = Process Variable at time "t"

e(t) = SP-PV(t) = PV deviation from setpoint at time "t" or PV error.

Then:

M(t) = Control output at time "t"

$$M(t) = Kc \left[e(t) + 1/T_i \int_0^t e(x)\, dx + T_d\, d/dt\, e(t) \right] + M_o$$

The analog input module receives the process variable in analog form along with an operator entered setpoint; the CPU computes the error. The error is used in the algorithm computation to provide corrective action at the control output. The function of the control action is based on an output control, which is proportional to the instantaneous error value. The *integral control action* (reset action) provides additional compensation to the control output, which causes a change in proportion to the value of the change of error over a period of time. The *derivative control action* (rate change) adds compensation to the control output, which causes a change in proportion to the rate of change of error. These three modes are used to provide the desired control action in Proportional (P), Proportional-Integral (PI), or Proportional-Integral-Derivative (PID) control fashion.

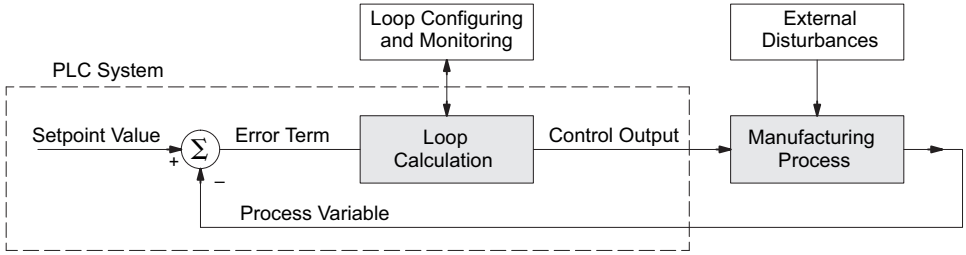
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Standard DL205 analog input modules are used to interface to field transmitters to obtain the PV. These transmitters normally provide a 4-20mA current or an analog voltage of various ranges for the control loop.

For temperature control, thermocouple or RTD can be connected directly to the appropriate module. The PID control algorithm, residing in the CPU memory, receives information from the user program, primarily control parameters and setpoints. Once the CPU makes the PID calculation, the result may be used to directly control an actuator connected to a 4-20mA current output module to control a valve.

With *Direct*SOFT, additional ladder logic programming, both time proportioning (e.g. heaters for temperature control) and position actuator (e.g. reversible motor on a valve) type of control schemes can be easily implemented. This chapter will explain how to set up the PID control loop, how to implement the software and how to tune the loop.

The following block diagram shows the key parts of a PID control loop. The path from the PLC to the manufacturing process and back to the PLC is the closed loop control.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Process Control Definitions

Manufacturing Process – the set of actions that adds value to raw materials. The process can involve physical changes and/or chemical changes to the material. The changes render the material more useful for a particular purpose, ultimately used in a final product.

Process Variable – The controlled variable part of the process that you wish to control. It may be temperature, pressure, level, flow, composition, density, the ratio of two streams, etc. Also known as the actual value.

Setpoint – This is the target for the process variable. When all conditions of the process are correct, the process variable will equal the setpoint.

Control Output – The result of the loop calculation, which becomes a command for the process (such as the heater level in an oven). This is sometimes referred to as control variable.

Error Term – The algebraic difference between the process variable and the setpoint. This is the control loop error, and is equal to zero when the process variable is equal to the setpoint (desired) value. A well-behaved control loop is able to maintain a small error term magnitude.

Manipulated Variable – This is what is used to affect the controlled variable. For example, the fuel used in a furnace might be manipulated in order to control the temperature.

Disturbance – Something in the system that changes such that corrective action is required. For instance, when controlling a flow and the upstream pressure drops, the control valve must open wider in order to keep flow constant. The drop in upstream pressure is the disturbance.

Final Control Element – This is the physical device used to control the manipulated variable. Valves are probably the most widely used final control element.

Lag Time – The time it takes for the process to respond to a change in manipulated variable. This is also known as the capacitance of the system. When you're in the shower and you turn up the hot water a little, the time it takes before the water gets hot is the lag time.

Dead Time – The time it takes for a change in the process to be recognized. Composition analyzers and quality control are usually sources of significant dead time.

Loop Configuring – Operator-initiated selections which set up and optimize the performance of a control loop. The loop calculation function uses the configuration parameters in real time to adjust gains, offsets, etc.

Loop Monitoring – The function which allows an operator to observe the status and performance of a control loop. This is used in conjunction with the loop configuring to optimize the performance of a loop (minimize the error term).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Loop Operation

The Proportional–Integral–Derivative (PID) algorithm is widely used in process control. The PID method of control adapts well to electronic solutions, whether implemented in analog or digital (CPU) components. The DL205 CPU implements the PID equations digitally by solving the basic equations in software. I/O modules serve only to convert electronic signals into digital form (or vice versa).

The DL205 uses two types of PID controls: "position" and "velocity". These terms usually refer to motion control situations, but here we use them in a different sense:

- PID *Position* Algorithm – The control output is calculated so it responds to the displacement (position) of the PV from the SP (error term).
- PID *Velocity* Algorithm – The control output is calculated to represent the rate of change (velocity) for the PV to become equal to the SP.

Position Form of the PID Equation

Referring to the control output equation on page 8-6, the DL205 CPU approximates the output M(t) using a discrete position form of the PID equation.

Let:

Ts = Sample rate
Kc = Proportional gain
Ki = Kc * (Ts/Ti) = Coefficient of integral term
Kr = Kc * (Td/Ts) = Coefficient of derivative term
Ti = Reset or integral time
Td = Derivative time or rate
SP = Setpoint
PV_n = Process variable at n^{th} sample
en = SP – PV_n = Error at n^{th} sample
M_o = Value to which the controller output has been initialized

Then:

M_n = Control output at n^{th} sample

$$M_n = Kc * e_n + Ki \sum_{i=1}^{n} e_i + Kr\,(e_n - e_{n-1}) + M_o$$

This form of the PID equation is referred to as the position form since the actual actuator position is computed. The velocity form of the PID equation computes the change in actuator position. The CPU modifies the standard equation slightly to use the derivative of the process variable instead of the error as follows:

$$M_n = Kc * e_n + Ki \sum_{i=1}^{n} e_i + Kr\,(PV_n - PV_{n-1}) + M_o$$

These two forms are equivalent unless the setpoint is changed. In the original equation, a large step change in the setpoint will cause a correspondingly large change in the error resulting in a bump to the process due to derivative action. This bump is not present in the second form of the equation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The DL205 also combines the integral sum and the initial output into a single term called the bias (Mx). This results in the following set of equations:

$$Mx_o = M_o$$

$$Mx = Ki * e_n + Mx_{n-1}$$

$$M_n = Kc * e_n - Kr(PV_n - PV_{n-1}) + Mx_n$$

The DL205 by default will keep the normalized output M in the range of 0.0 to 1.0. This is done by clamping M to the nearer of 0.0 or 1.0 whenever the calculated output falls outside this range. The DL205 also allows you to specify the minimum and maximum output limit values (within the range 0 to 4095 in binary if using 12-bit unipolar).



NOTE: The equations and algorithms , or parts of, in this chapter are only for references. Analysis of these equations can be found in most good text books about process control.

Reset Windup Protection

Reset windup can occur if reset action (integral term) is specified and the computation of the bias term Mx is:

$$Mx = Ki * e_n + Mx_{n-1}$$

For example, assume the output is controlling a valve and the PV remains at some value greater than the setpoint. The negative error (e_n) will cause the bias term (Mx) to constantly decrease until the output M goes to 0 closing the valve. However, since the error term is still negative, the bias will continue to decrease becoming ever more negative. When the PV finally does come back down below the SP, the valve will stay closed until the error is positive for long enough to cause the bias to become positive again. This will cause the process variable to undershoot.

One way to solve the problem is to simply clamp the normalized bias between 0.0 and 1.0. The DL205 CPU does this. However, if this is the only thing that is done, then the output will not move off 0.0 (thus opening the valve) until the PV has become less than the SP. This will also cause the process variable to undershoot.

The DL205 CPU is programmed to solve the overshoot problem by either freezing the bias term, or by adjusting the bias term.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Freeze Bias

If the "Freeze Bias" option is selected when setting up the PID loop (discussed later) then the CPU simply stops changing the bias (Mx) whenever the computed normalized output (M) goes outside the interval 0.0 to 1.0.

$$Mx = Ki * e_n + Mx_{n-1}$$

$$M = Kc * e_n - Kr(PV_n - PV_{n-1}) + Mx$$

$$M_n = 0 \quad \text{if } M < 0$$
$$M_n = M \quad \text{if } 0 \leq M \leq 1$$
$$M_n = 1 \quad \text{if } M > 1$$

$$Mx_n = Mx \quad \text{if } 0 \leq M \leq 1$$
$$Mx_n = Mx_{n-1} \quad \text{otherwise}$$

Thus in this example, the bias will probably not go all the way to zero so that, when the PV does begin to come down, the loop will begin to open the valve sooner than it would have if the bias had been allowed to go all the way to zero. This action has the effect of reducing the amount of overshoot.

Adjusting the Bias

The normal action of the CPU is to adjust the bias term when the output goes out of range as shown below.

$$Mx = Ki * e_n + Mx_{n-1}$$
$$M = Kc * e_n - Kr(PV_n - PV_{n-1}) + Mx$$

$$M_n = 0 \quad \text{if } M < 0$$
$$M_n = M \quad \text{if } 0 \leq M \leq 1$$
$$M_n = 1 \quad \text{if } M > 1$$

$$Mx_n = Mx \quad \text{if } 0 \leq M \leq 1$$
$$Mx_n = M_n - Kc * e_n - Kr(PV_n - PV_{n-1}) \quad \text{otherwise}$$

By adjusting the bias, the valve will begin to open as soon as the PV begins to come down. If the loop is properly tuned, overshoot can be eliminated entirely. If the output went out of range due to a setpoint change, then the loop probably will oscillate because we must wait for the bias term to stabilize again.

The choice of whether to use the default loop action or to freeze the bias is dependent on the application. **If large, step changes to the setpoint are anticipated, then it is probably better to select the freeze bias option** (see page 8-35).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step Bias Proportional to Step Change in SP

This feature reduces oscillation caused by a step change in setpoint when the adjusting bias feature is used.

$Mx = Mx * SP_n / SP_{n-1}$ if the loop is direct acting

$Mx = Mx * SP_{n-1} / SP_n$ if the loop is reverse acting

$Mx_n = 0$ if $Mx < 0$
$Mx_n = Mx$ if $0 \leq Mx \leq 1$
$Mx_n = 1$ if $Mx > 1$

Eliminating Proportional, Integral or Derivative Action

It is not always necessary to run a full three mode PID control loop. Most loops require only the PI terms or just the P term. Parts of the PID equation may be eliminated by choosing appropriate values for the gain (Kc), reset (Ti) and rate (Td) yielding a P, PI, PD, I and even an ID and a D loop.

Eliminating Integral Action The effect of integral action on the output may be eliminated by setting Ti = 9999 or 0000. When this is done, the user may then manually control the bias term (Mx) to eliminate any steady-state offset.

Eliminating Derivative Action The effect of derivative action on the output may be eliminated by setting Td = 0 (most loops do not require a D parameter; it may make the loop unstable).

Eliminating Proportional Action Although rarely done, the effect of proportional term on the output may be eliminated by setting Kc = 0. Since Kc is also normally a multiplier of the integral coefficient (Ki) and the derivative coefficient (Kr), the CPU makes the computation of these values conditional on the value of Kc as follows:

$Ki = Kc * (Ts / Ti)$ if $Kc \neq 0$
$Ki = Ts / Ti$ if $Kc = 0$ (I or ID only)
$Kr = Kc * (Td / Ts)$ if $Kc \neq 0$
$Kr = Td / Ts$ if $Kc = 0$ (ID or D only)

Velocity Form of the PID Equation

The standard position form of the PID equation computes the actual actuator position. An alternative form of the PID equation computes the change in actuator position. This form of the equation is referred to as the velocity PID equation and is obtained by subtracting the equation at time "n" from the equation at time "n-1".

The velocity equation is given by:

$\Delta M_n = M - M_{n-1}$

$\Delta M_n = Kc * (e_n - e_{n-1}) + Ki * (PV_n - 2 * PV_{n-1} + PV_{n-2})$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bumpless Transfer

The DL205 loop controller provides for bumpless mode changes. A bumpless transfer from manual mode to automatic mode is achieved by preventing the control output from changing immediately after the mode change.

When a loop is switched from Manual mode to Automatic mode, the setpoint and Bias are initialized as follows:

Position PID Algorithm	Velocity PID Algorithm
SP = PV	SP = PV
Mx = M	

The bumpless transfer feature of the DL205 is available in two types: Bumpless I and Bumpless II (see page 8-27). The transfer type is selected when the loop is set up.

Loop Alarms

The DL205 allows the user to specify alarm conditions that are to be monitored for each loop. Alarm conditions are reported to the CPU by setting up the alarms in ***Direct***SOFT using the PID setup alarm dialog when the loop is set up. The alarm features for each loop are:

- **PV Limit** – Specify up to four PV alarm points.

High-High	PV rises above the programmed High-High Alarm Limit.
High	PV rises above the programmed High Alarm Limit.
Low	PV falls below the Low Alarm Limit.
Low-Low	PV falls below the Low-Low Limit.

- **PV Deviation Alarm** – Specify an alarm for High and Low PV deviation from the setpoint (Yellow Deviation). An alarm for High High and Low Low PV deviation from the setpoint (Orange Deviation) may also be specified. When the PV is further from the setpoint than the programmed Yellow or Orange Deviation Limit the corresponding alarm bit is activated.
- **Rate of Change** – This alarm is set when the PV changes faster than a specified rate-of-change limit.
- **PV Alarm Hysteresis** – The PV Limit Alarms and PV Deviation Alarms are programmed using threshold values. When the absolute value or deviation exceeds the threshold, the alarm status becomes true. Real-world PV signals have some noise on them, which can cause some fluctuation in the PV value in the CPU. As the PV value crosses an alarm threshold, its fluctuations will cause the alarm to be intermittent and annoy process operators. The solution is to use the PV Alarm Hysteresis feature.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Loop Operating Modes

The DL205 loop controller operates in one of three modes, either *Manual*, *Automatic* or *Cascade*.

Manual

In manual mode, the control output is determined by the operator, not the loop controller. While in manual mode, the loop controller will still monitor all of the alarms including High-High, High, Low, Low-Low, Yellow deviation, Orange deviation and Rate-of-Change.

Automatic

In automatic mode, the loop controller computes the control output based on the programmed parameters stored in V-memory. All alarms are monitored while in automatic.

Cascade

Cascade mode is an option with the DL205 PLC and is used in special control applications. If the cascade feature is used, the loop will operate as it would if in automatic mode except that a cascaded loop has a setpoint which is the control output from another loop.

Special Loop Calculations

Reverse Acting Loop

Although the PID algorithm is used in a direct, or forward, acting loop controller, there are times when a reverse acting control output is needed. The DL205 loop controller allows a loop to operate as reverse acting. With a reverse acting loop, the output is driven in the opposite direction of the error. For example, if SP > PV, then a reverse acting controller will decrease the output to increase the PV.

$$Mx = -Ki * e_n + Mx_{n-1}$$

$$M = -Kc * e_n + Kr(PV_n - PV_{n-1}) + Mx_n$$

Square Root of the Process Variable

Square root is selected whenever the PV is from a device such as an orifice meter which requires this calculation.

Error Squared Control

Whenever error squared control is selected, the error is calculated as:

$$e_n = (SP - PV_n) * ABS(SP - PV_n)$$

A loop using the error squared is less responsive than a loop using just the error; however, it will respond faster with a large error. The smaller the error, the less responsive the loop. Error squared control would typically be used in a pH control application.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Error Deadband Control

With error deadband control, no control action is taken if the PV is within the specified deadband area around the setpoint. The error deadband is the same above and below the setpoint.

Once the PV is outside of the error deadband around the setpoint, the entire error is used in the loop calculation.

$e_n = 0$ SP - Deadband_Below_SP < PV < SP - Deadband_Above_SP
$e_n = P - PV_n$ otherwise

The error will be squared first if both Error Squared and Error Deadband is selected.

Derivative Gain Limiting

When the coefficient of the derivative term, Kr, is a large value, noise introduced into the PV can result in erratic loop output. This problem is corrected by specifying a derivative gain limiting coefficient, Kd. Derivative gain limiting is a first order filter applied to the derivative term computation, Y_n, as shown below.

$$Y_n = Y_{n-1} + \frac{Ts}{Ts + (\frac{Td}{Kd})} * (PV_n - Y_{n-1})$$

Position Algorithm

$$Mx = Ki * e_n + Mx_{n-1}$$
$$M = Kc * e_n - Kr * (Y_n - Y_{n-1}) + Mx$$

Velocity Algorithm

$$\Delta M = Kc * (e_n - e_{n-1}) + Ki * e_n - Kr * (Y_n - 2 * Y_{n-1} + Y_{n-2})$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Ten Steps to Successful Process Control

Controllers such as the DL205 PLC provide sophisticated process control features. Automated control systems can be difficult to debug, because a given symptom can have many possible causes. We recommend a careful, step-by-step approach to bringing new control loops online:

Step 1: Know the Recipe

The most important knowledge is – how to produce your product. This knowledge is the foundation for designing an effective control system. A good process *recipe* will do the following:

- Identify all relevant Process Variables (PV), such as temperature, pressure, or flow rates, etc. which need precise control.
- Plot the desired Setpoint values for each process variable for the duration of one process cycle.

Step 2: Plan Loop Control Strategy

This simply means choosing the method the machine will use to maintain control over the Process Variables to follow their Setpoints. This involves many issues and trade-offs, such as energy efficiency, equipment costs, ability to service the machine during production, and more. You must also determine how to generate the Setpoint (SP) value during the process, and whether a machine operator can change the SP.

Step 3: Size and Scale Loop Components

Assuming the control strategy is sound, it is still crucial to *properly size the actuator and properly scale the sensors.*

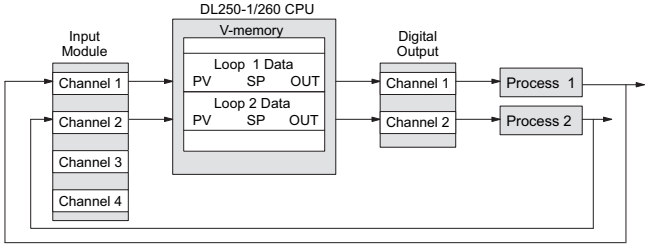
- Choose an actuator (heater, pump. etc.) which matches the size of the load. An oversized actuator will have an overwhelming effect on your process after a SP change. However, an undersized actuator will allow the PV to lag or drift away from the SP after a SP change or process disturbance.
- Choose a PV sensor which matches the range of interest (and control) for your process. Decide the resolution of control you need for the PV (such as within 2°C), and make sure the sensor input value provides the loop with at least 5 times that resolution (at LSB level). However, an over-sensitive sensor can cause control oscillations, etc.The DL205 provides 12–bit and 15–bit unipolar and bipolar data format options, and a 16–bit unipolar option. This selection affects SP, PV, Control Output and Integrator sum.

Step 4: Select I/O Modules

After deciding the number of loops, PV variables to measure, and SP values, you can choose the appropriate I/O module. Refer to the figure on the next page. In many cases, you will be able to share input or output modules, or use an analog I/O combination module, among several control loops. The example shown sends the PV and Control Output signals for two loops through the same set of modules.

AutomationDirect offers DL205 analog input modules with 4 and 8 channels per module that accept 0–20mA or 4-20mA signals and voltage. Analog combination I/O modules are also available. Thermocouple and RTD modules can also be used to maintain temperatures to a 10th of a degree. Refer to the sales catalog for further information on these modules, or find the modules on our website, **www.automationdirect.com.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Step 5: Wiring and Installation

After selection and procurement of all loop components and I/O module(s), you can perform the wiring and installation. Refer to the wiring guidelines in Chapter 2 of this manual, and to the D2-ANLG-M manual. The most common wiring errors when installing PID loop controls are:

- Reversing the polarity of sensor or actuator wiring connections.
- Incorrect signal ground connections between loop components.

Step 6: Loop Parameters

After wiring and installation, choose the loop setup parameters. The easiest method for programming the loop tables is using *Direct*SOFT (5.0 or later). This software provides PID Setup using dialog boxes to simplify the task. **Note: It is important to understand the meaning of all loop parameters mentioned in this chapter before choosing values to enter.**

Step 7: Check Open Loop Performance

With the sensor and actuator wiring done, and loop parameters entered, we must manually and carefully check out the new control system using the Manual mode).

- Verify that the PV value from the sensor is correct.
- If it is safe to do so, gradually increase the control output up above 0%, and see if the PV responds (and moves in the correct direction!).

Step 8: Loop Tuning

If the Open Loop Test (page 8–41) shows the PV reading is correct and the control output has the proper effect on the process; you can follow the closed loop tuning procedure (see page 8–47). In this step, the loop is tuned so the PV automatically follows the SP.

Step 9: Run Process Cycle

If the closed loop test shows the PV will follow small changes in the SP, consider running an actual process cycle. You will need to have completed the programming which will generate the desired SP in real time. In this step, you may want to run a small test batch of product through the machine, watching the SP change according to the recipe.

WARNING: Be sure the Emergency Stop and power-down provision is readily accessible, in case the process goes out of control. Damage to equipment and/or serious injury to personnel can result from loss of control of some processes.

Step 10: Save Parameters

When the loop tests and tuning sessions are complete, be sure to save all loop setup parameters to disk.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Loop Setup

Some Things to Do and Know Before Starting

Install the analog module, and be sure it is operational before beginning the loop setup (refer to the DL205 analog modules user manual, D2-ANLG-M. The DL205 PLC gets its PID loop processing instructions from V-memory tables. There isn't a PID instruction that can be used in RLL, such as a block, to setup the PID loop control. Instead, the CPU reads the setup parameters from system V-memory locations. These locations are shown in the table below for reference only; they can be used in a RLL program if needed.

Address	Setup Parameter	Data type	Ranges	Read/Write
V7640	Loop Parameter Table Pointer	Octal	V1400 – V7340 V10000-V17740 (DL250-1) V10000 - V37740 (DL260)	write
V7641	Number of Loops	BCD	1 – 4 (DL250-1) 1 - 16 (DL260)	write
V7642	Loop Error Flags	BITS	0 or 1	read



NOTE: The V-memory data is stored in RAM memory. If power is removed from the CPU for an extended period of time, the PID Setup Parameters will be lost. It is recommended to use the optional D2-BAT-1 for memory backup.

8

PID Error Flags



The CPU reports any programming errors of the setup parameters in V7640 and V7641. It does this by setting the appropriate bits in V7642 on program-to-run mode transitions.

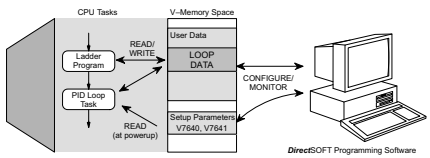
If you use the *Direct*SOFT loop setup dialog box, its automatic range checking prohibits possible setup errors. However, the setup parameters may be written using other methods such as RLL, so the error flag register may be helpful in those cases. The following table lists the errors reported in V7642.

Bit	Error Description (0 = no error, 1 = error)
0	The starting address (in V7640) is out of the lower V-memory range.
1	The starting address (in V7640) is out of the upper V-memory range.
2	The number of loops selected (in V7641) is greater than 4 (DL250-1) and 16 (DL260).
3	The loop table extends past (straddles) the boundary at V7377. Use an address closer to V1400.
4	The loop table extends past (straddles) the boundary at V17777 (DL250-1) or V35777 (DL260). Use an address closer to V10000.

If the CPU is in Run mode and V7642=0000, there are no programming errors.

Establishing the Loop Table Size and Location

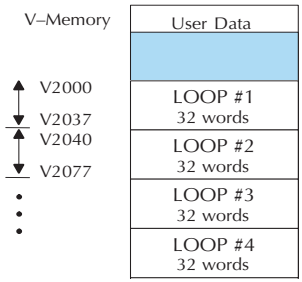
On a PROGRAM-to-RUN mode transition, the CPU reads the loop setup parameters as pictured below. At that moment, the CPU learns the location of the loop table and the number of loops it configures. Then, during the ladder program scan, the PID Loop task uses the loop data to perform calculations, generate alarms, and so on. There are some loop table parameters the CPU will read or write on every loop calculation.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

*NOTE: The DL205 CPU's PID algorithm requires **Direct**SOFT and the DL250-1 or DL260 CPUs with any firmware version.*

The Loop Table contains data for only the number of loops that are selected. The address for the table is stored in V7641. Each loop configuration occupies 32 words (0 to 37 octal) in the loop table.

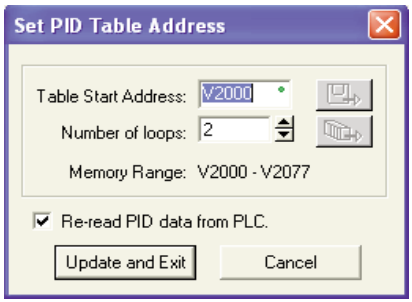


For example, consider an application with 4 loops, and V2000 has been chosen as the starting location. The Loop Parameter will occupy V2000 – V2037 for loop 1, V2040 – V2077 for loop 2 and so on. Loop 4 occupies V2140 - V2177.

Determine the block of V-memory to be used for each PID loop. Besides being the beginning of the PID parameter memory block, the first address will be the start of loop 1 parameters. Remember, there are 32 words (0 to 37 octal) needed for each loop. Once you have determined the beginning V-memory address to be used, you can setup and store the PID parameters either directly in your RLL program or by using the PID Setup in ***Direct***SOFT.

*NOTE: Whether one or more loops are being setup, this block of V-memory will only be used for the PID loop parameters, **do not use this block of memory for anything else in your program.***

Using ***Direct***SOFT is the simplest way to setup the parameters. To setup the PID parameters, the DL205 must be powered up and connected to the programming computer. The parameters can only be entered in PID setup when the PLC is in the Program mode. Once the parameters have been entered and saved for each loop, changes through the PID setup can be made, but only in Program Mode. You can type the beginning address in the PID Table Address dialog found when the PID Setup is opened in ***Direct***SOFT. This can be seen in the diagram below. After the address has been entered, the memory range will appear. Also, entering the number of PID loops (1 to 4 for the DL250-1, 1 to 16 for the DL260) will set the total V-memory range for the number of loops entered. After the V-memory address has been entered, the necessary PID parameters for a basic loop operation for each loop can be setup with the dialogs made available.



*NOTE: Have a **Direct**SOFT project open, then click on PLC > Setup > PID to access the Setup PID dialog.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Loop Table Word Definitions

These are the loop parameters associated with each of the four loops available in the DL205. The parameters are listed in the following table. The address offset is in octal, to help you locate specific parameters in the loop table. For example, if a table begins at V2000, then the location of the reset (integral) term is Addr+11, or V2011. Do not use the Word # (in the first column) to calculate addresses.

Word #	Address+Offset	Description	Format	Change on-the-fly[1]
1	Addr + 0	PID Loop Mode Setting 1	bits	Yes
2	Addr + 1	PID Loop Mode Setting 2	bits	Yes
3	Addr + 2	Setpoint Value (SP)	word/binary	Yes
4	Addr + 3	Process Variable (PV)	word/binary	Yes
5	Addr + 4	Bias (Integrator) Value	word/binary	Yes
6	Addr + 5	Control Output Value	word/binary	Yes
7	Addr + 6	Loop Mode and Alarm Status	bits	–
8	Addr + 7	Sample Rate Setting	word/BCD	Yes
9	Addr + 10	Gain (Proportional) Setting	word/BCD	Yes
10	Addr + 11	Reset (Integral) Time Setting	word/BCD	Yes
11	Addr + 12	Rate (Derivative) Time Setting	word/BCD	Yes
12	Addr + 13	PV Value, Low-low Alarm	word/binary	No[2]
13	Addr + 14	PV Value, Low Alarm	word/binary	No[2]
14	Addr + 15	PV Value, High Alarm	word/binary	No[2]
15	Addr + 16	PV Value, High-high Alarm	word/binary	No[2]
16	Addr + 17	PV Value, deviation alarm (YELLOW)	word/binary	No[2]
17	Addr + 20	PV Value, deviation alarm (RED)	word/binary	No[2]
18	Addr + 21	PV Value, rate-of-change alarm	word/binary	No[2]
19	Addr + 22	PV Value, alarm hysteresis setting	word/binary	No[2]
20	Addr + 23	PV Value, error deadband setting	word/binary	Yes
21	Addr + 24	PV low-pass filter constant	word/BCD	Yes
22	Addr + 25	Loop derivative gain limiting factor setting	word/BCD	No[3]
23	Addr + 26	SP value lower limit setting	word/binary	Yes
24	Addr + 27	SP value upper limit setting	word/binary	Yes
25	Addr + 30	Control output value lower limit setting	word/binary	No[3]
26	Addr + 31	Control output value upper limit setting	word/binary	No[3]
27	Addr + 32	Remote SP Value V-Memory Address Pointer	word/hex	Yes
28	Addr + 33	Ramp/Soak Setting Flag	bit	Yes
29	Addr + 34	Ramp/Soak Programming Table Starting Address	word/hex	No[3]
30	Addr + 35	Ramp/Soak Programming Table Error Flags	bits	No[3]
31	Addr + 36	PV auto transfer, channel number	word/hex	Yes
32	Addr + 37	Control output auto transfer, channel number	word/hex	Yes

Notes:

[1] "Yes" in the column indicates the PID loop will use an updated PID parameter value immediately.

[2] Read data only when alarm enable bit transitions from 0 to 1.

[3] Read data only on PLC Mode change.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Mode Setting 1 Bit Descriptions (Addr + 00)

The individual bit definitions of the PID Mode Setting 1 word (Addr+00) are listed in the following table.

Bit	PID Mode Setting 1 Description	Read/Write	Bit=0	Bit=1
0	Manual Mode Loop Operation request	write	–	01 request
1	Automatic Mode Loop Operation request	write	–	01 request
2	Cascade Mode Loop Operation request	write	–	01 request
3	Bumpless Transfer select	write	Mode I	Mode II
4	Direct or Reverse-Acting Loop select	write	Direct	Reverse
5	Position/Velocity Algorithm select	write	Position	Velocity
6	PV Linear/Square Root Extract select	write	Linear	Sq. root
7	Error Term Linear/Squared select	write	Linear	Squared
8	Error Deadband enable	write	Disable	Enable
9	Derivative Gain Limit select	write	Off	On
10	Bias (Integrator) Freeze select	write	Off	On
11	Ramp/Soak Operation select	write	Off	On
12	PV Alarm Monitor select	write	Off	On
13	PV Deviation alarm select	write	Off	On
14	PV rate-of-change alarm select	write	Off	On
15	Loop mode is independent from CPU mode when set	write	Loop with CPU mode	Loop Independent of CPU mode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Mode Setting 2 Bit Descriptions (Addr + 01)

The individual bit definitions of the PID Mode Setting 2 word (Addr+01) are listed in the following table.

Bit	PID Mode 2 Word Description	Read/Write	Bit=0	Bit=1
0	Input (PV) and Control Output Range Unipolar/Bipolar select (See Notes 2 and 3)	write	unipolar	bipolar
1	Input/Output Data Format select (See Notes 2 and 3)	write	12 bit	15 bit
2	Analog Input filter	write	off	on
3	SP Input limit enable	write	disable	enable
4	Integral Gain (Reset) units select	write	seconds	minutes
5	Select Auto tune PID algorithm	write	closed loop	open loop
6	Auto tune selection	write	PID	PI only (rate = 0)
7	Auto tune start (See Note 1)	read/write	auto tune cancel/done	force start
8	PID Scan Clock (internal use)	read	–	–
9	Input/Output Data Format 16-bit select (See Notes 2 and 3)	write	not 16 bit	select 16 bit
10	Select separate data format for input and output (See Notes 3 and 4)	write	same format	separate formats
11	Control Output Range Unipolar/Bipolar select (See Notes 3 and 4)	write	unipolar	bipolar
12	Output Data Format select (See Notes 3 and 4)	write	12 bit	15 bit
13	Output data format 16-bit select (See Notes 3 and 4)	write	not 16 bit	select16 bit
14–15	Reserved for future use	–	–	–



NOTE 1: Bit 7 can be used to cancel Autotune mode by setting it to 0.



NOTE 2: If the value in bit 9 is 0, then the values in bits 0 and 1 are read. If the value in bit 9 is 1, then the values in bits 0 and 1 are not read, and bit 9 defines the data format (the range is automatically unipolar).



NOTE 3: If the value in bit 10 is 0, then the values in bits 0, 1 and 9 define the input and output ranges and data formats (the values in bits 11, 12, and 13 are not read). If the value in bit 10 is 1, then the values in bits 0, 1, and 9 define only the input range and data format, and bits 11, 12, and 13 are read and define the output range and data format.



NOTE 4: If bit 10 has a value of 1 and bit 13 has a value of 0, then bits 11 and 12 are read and define the output range and data format. If bit 10 and bit 13 each have a value of 1, then bits 11 and 12 are not read, and bit 13 defines the data format, (the output range is automatically unipolar).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Mode/Alarm Monitoring Word (Addr + 06)

The individual bit definitions of the Mode/Alarm monitoring (Addr+06) word is listed in the following table.

Bit	Mode/Alarm Bit Description	Read/Write	Bit=0	Bit=1
0	Manual Mode Indication	read	–	Manual
1	Automatic Mode Indication	read	–	Auto
2	Cascade Mode Indication	read	–	Cascade
3	PV Input LOW–LOW Alarm	read	Off	On
4	PV Input LOW Alarm	read	Off	On
5	PV Input HIGH Alarm	read	Off	On
6	PV Input HIGH–HIGH Alarm	read	Off	On
7	PV Input YELLOW Deviation Alarm	read	Off	On
8	PV Input RED Deviation Alarm	read	Off	On
9	PV Input Rate-of-Change Alarm	read	Off	On
10	Alarm Value Programming Error	read	–	Error
11	Loop Calculation Overflow/Underflow	read	–	Error
12	Loop in Auto-Tune indication	read	Off	On
13	Auto-Tune error indication	read	–	Error
14–15	Reserved for Future Use	–	–	–

Ramp/Soak Table Flags (Addr + 33)

The individual bit definitions of the Ramp/Soak Table Flag (Addr+33) word is listed in the following table.

Bit	Ramp/Soak Flag Bit Description	Read/Write	Bit=0	Bit=1
0	Start Ramp/Soak Profile	write	–	01 Start
1	Hold Ramp/Soak Profile	write	–	01 Hold
2	Resume Ramp/Soak Profile	write	–	01 Resume
3	Jog Ramp/Soak Profile	write	–	01 Jog
4	Ramp/Soak Profile Complete	read	–	Complete
5	PV Input Ramp/Soak Deviation	read	Off	On
6	Ramp/Soak Profile in Hold	read	Off	On
7	Reserved	read	–	–
8–15	Current Step in Ramp/Soak Profile	read	decode as byte (hex)	

Bits 8–15 must be read as a byte to indicate the current segment number of the Ramp/Soak generator in the profile. This byte will have the values 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, and 10, which represent segments 1 to 16 respectively. If the byte=0, then the Ramp/Soak table is not active.

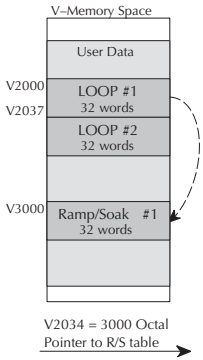
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Ramp/Soak Table Location (Addr + 34)

Each loop that you configure has the option of using a built-in Ramp/Soak generator dedicated to that loop. This feature generates setpoint (SP) values that follow a profile. To use the Ramp/Soak feature, you must program a separate table of 32 words with appropriate values. A *Direct*SOFT dialog box makes this easy to do.

In the loop table, the Ramp/Soak Table Pointer at Addr+34 must point to the start of the Ramp/Soak data for that loop. This may be anywhere in user memory, and does not have to adjoin to the Loop Parameter table, as shown to the left. Each Ramp/Soak table requires 32 words, regardless of the number of segments programmed.

The Ramp/Soak table parameters are defined in the table below. Further details are in the section on Ramp/Soak Operation in this chapter.



Addr Offset	Step	Description	Addr Offset	Step	Description
+ 00	1	Ramp End SP Value	+ 20	9	Ramp End SP Value
+ 01	1	Ramp Slope	+ 21	9	Ramp Slope
+ 02	2	Soak Duration	+ 22	10	Soak Duration
+ 03	2	Soak PV Deviation	+ 23	10	Soak PV Deviation
+ 04	3	Ramp End SP Value	+ 24	11	Ramp End SP Value
+ 05	3	Ramp Slope	+ 25	11	Ramp Slope
+ 06	4	Soak Duration	+ 26	12	Soak Duration
+ 07	4	Soak PV Deviation	+ 27	12	Soak PV Deviation
+ 10	5	Ramp End SP Value	+ 30	13	Ramp End SP Value
+ 11	5	Ramp Slope	+ 31	13	Ramp Slope
+ 12	6	Soak Duration	+ 32	14	Soak Duration
+ 13	6	Soak PV Deviation	+ 33	14	Soak PV Deviation
+ 14	7	Ramp End SP Value	+ 34	15	Ramp End SP Value
+ 15	7	Ramp Slope	+ 35	15	Ramp Slope
+ 16	8	Soak Duration	+ 36	16	Soak Duration
+ 17	8	Soak PV Deviation	+ 37	16	Soak PV Deviation

Ramp/Soak Table Programming Error Flags (Addr + 35)

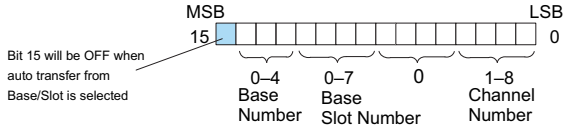
The individual bit definitions of the Ramp/Soak Table **Programming Error Flags** word (Addr+35) is listed in the following table. Further details are given in the PID Loop Mode section and in the PV Alarm section later in this chapter.

Bit	R/S Error Flag Bit Description	Read/Write	Bit=0	Bit=1
0	Starting Addr out of lower V-memory range	read	–	Error
1	Starting Addr out of upper V-memory range	read	–	Error
2–3	Reserved for Future Use	–	–	–
4	Starting Addr in System Parameter V-memory Range	read	–	Error
5–15	Reserved for Future Use	–	–	–

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PV Auto Transfer (Addr + 36) from I/O Module Base/Slot/Channel Option

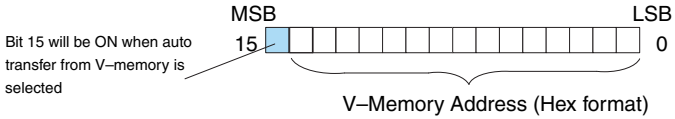
The nibble definitions for PV Auto Transfer word (Addr + 36) are listed in the table below for the Transfer from Base/Slot option. **When this option is used for any channel on an analog input module, the ladder logic pointer method can not be used for this module.** (Refer to the DL205 Analog I/O Modules (D2-ANLG-M) for pointer method information).



CPU	Base Number	Base Slot Number	Channel Number
DL250-1	Local CPU base = 0 Local expansion base = 1-2	0-7	1-8
DL260	Local CPU base = 0 Local expansion base = 1-4	0-7	1-8

PV Auto Transfer (Addr + 36) from V-memory Option

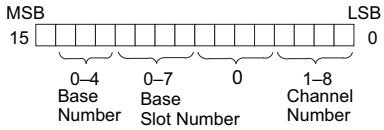
The nibble definitions for PV Auto Transfer word (Addr + 36) are listed in the table below for the Transfer from V-memory option. The ladder logic pointer method can be used with this option to get the analog module's channel values into V-memory. (Refer to the DL205 Analog I/O Modules (D2-ANLG-M) for pointer method information).



Memory Type	DL250-1 Range	DL260 Range
V-memory	V1400-V7377 V10000-1777	V400-V677 V1400-V7377 V10000-V35777

Control Output Auto Transfer (Addr + 37)

The nibble definitions for Control Output Auto Transfer word (Addr + 37) are listed in the table below. **When the Control Output Auto Transfer function is used for any channel on an analog input module, the ladder logic pointer method can not be used for this module.** (Refer to the DL205 Analog I/O Modules (D2-ANLG-M) for pointer method information).

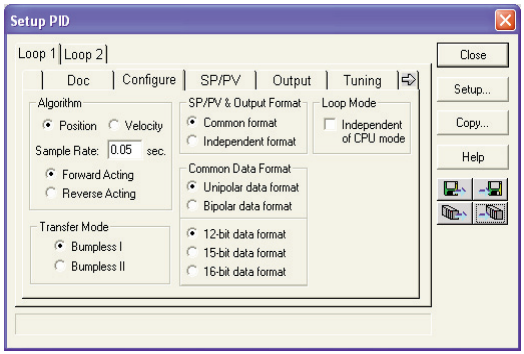


CPU	Base Number	Base Slot Number	Channel Number
DL250-1	Local CPU base = 0 Local expansion base = 1-2	0-7	1-8
DL260	Local CPU base = 0 Local expansion base = 1-4	0-7	1-8

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configure the PID Loop

Once the PID table is established in V-memory, configuring the PID loop continues with the *Direct*SOFT PID setup configuration dialog. You will need to check and fill in the data required to control the PID loop. Select Configure and the following dialog will appear for this process.



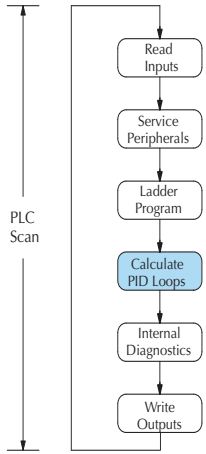
8

Select the Algorithm Type

Chose either *Position* or *Velocity*. The default algorithm is Position. This is the choice for most applications which include heating and cooling loops as well as most position and level control loops. A typical velocity control will consist of a process variable such as a flow totalizer in a flow control loop.

Enter the Sample Rate

The main tasks of the CPU fall into categories as shown to the right. The list represents the tasks done when the CPU is in Run Mode, on each PLC scan. Note that PID loop calculations occur after the ladder logic task.



The sample rate of a control loop is simply the frequency of the PID calculation. Each calculation generates a new control output value. With the DL205 CPUs, you can set the sample rate of a loop from 50 ms to 99.99 seconds. Most loops do not require a fresh PID calculation on every PLC scan. Some loops may need to be calculated only once in 1000 scans.

Enter 0.05 sec., or the sample rate of your choice, for each loop, and the CPU automatically schedules and executes PID calculations on the appropriate scans.



NOTE: If more than 4 loops are programmed, enter a minimum of 0.1 second.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Select Forward/Reverse

It is important to know which direction the control output will respond to the error (SP-PV), either *forward* or *reverse.* A forward (direct) acting control loop means that whenever the control output increases, the process variable will also increase. The control output of most PID loops are forward acting, such as a heating control loop. An increase in heat applied will increase the PV (temperature).

A reverse acting control loop is one where an increase in the control output results in a decrease in the PV. A common example of this would be a refrigeration system, where an increase in the cooling input causes a decrease in the PV (temperature).

The Transfer Mode

Choose either Bumpless I or Bumpless II to provide a smooth transition of the control output from Manual Mode to Auto Mode. Choosing Bumpless I will set the SP equal to the PV when the control output is switched from Manual to Auto. If this is not desired, choose Bumpless II.

The characteristics of Bumpless I and II transfer types are listed in the chart below. Note that their operation also depends on which PID algorithm you are using, the position or velocity form of the PID equation. Note that you must use Bumpless Transfer type I when using the velocity form of the PID algorithm.

TransferType	Transfer Select Bit 3	PID Algorithm	Manual-to-Auto Transfer Action	Auto-to-Cascade Transfer Action
Bumpless Transfer I	0	Position	Forces Bias = Control Output Forces SP = PV	Forces Major Loop Output = Minor Loop PV
		Velocity	Forces SP = PV	Forces Major Loop Output = Minor Loop PV
Bumpless Transfer II	1	Position	Forces Bias = Control Output	none
		Velocity	none	none

The transfer type can also be selected in a RLL program by setting bit 3 of PID Mode 1, V+00 setting as shown.



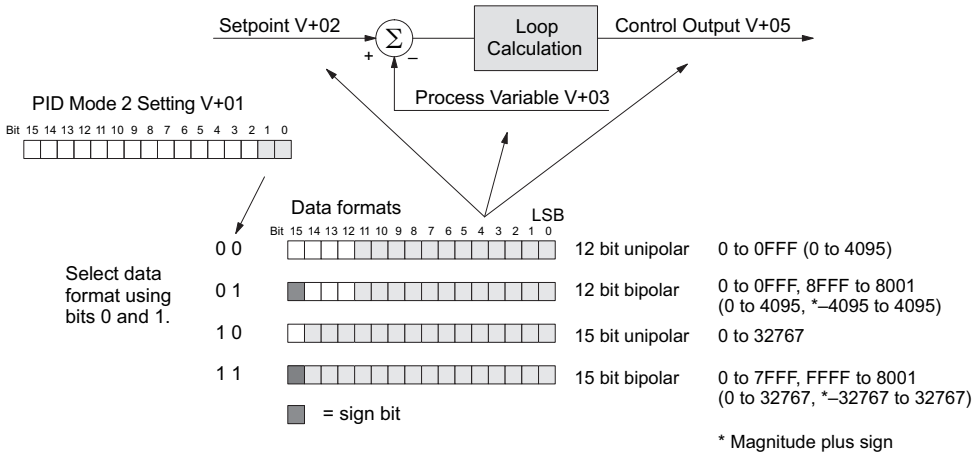
SP/PV & Output Format

This block allows you to select either *Common format* or *Independent format*. Common format is the default and is most commonly used. With this format both SP/PV and Output will have the same data structure. Both will have the same number of bits and either bipolar or unipolar. If Independent format is selected, the data structure selections will be grayed out. They will be independently selectable in the *SP/PV* and the *Output* dialogs.

Common Data Format

Select either *Unipolar data format* (which is positive data only) in 12-bit (0 to 4095), 15-bit (0 to 32767), or 16-bit (0 to 65535) format, or *Bipolar data format,* which ranges from negative to positive (-4095 to 4095 or -32767 to 32767) and requires a sign bit. Bipolar selection displays input/output as magnitude plus sign, not two's complement. The bipolar selection is not available when 16-bit data format is selected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



The data format determines the numerical interface between the PID loop and the PV sensor, and the control output device. This selects the data format for both the SP and the PV.

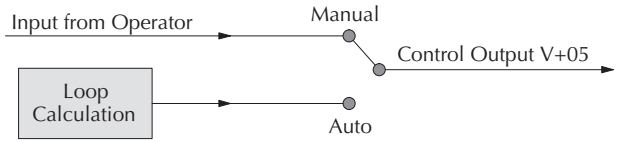
8

Loop Mode

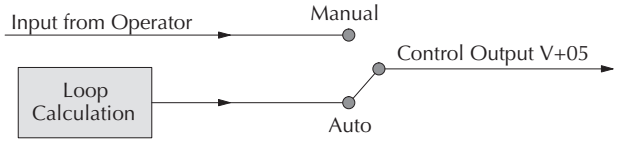
Loop Mode is a special feature that allows the PID loop controller to perform closed-loop control while the CPU is in the Program Mode. Careful thought must be taken before using this feature called *Independent of CPU mode* in the dialog. Before continuing with the PID setup, a knowledge of the three PID loop modes will be helpful.

The DL205 provides the three standard control modes: Manual, Automatic, and Cascade. The sources of the three basic variables SP, PV and control output are different for each mode.

In Manual Mode, the loop is not executing PID calculations (however, loop alarms are still active). With regard to the loop table, the CPU stops writing values to location V+05 (control output) for that loop. *It is expected that an operator or other intelligent source* is manually controlling the output by observing the PV and writing data to the control output as necessary to keep the process under control. The drawing below shows the equivalent schematic diagram of manual mode operation.

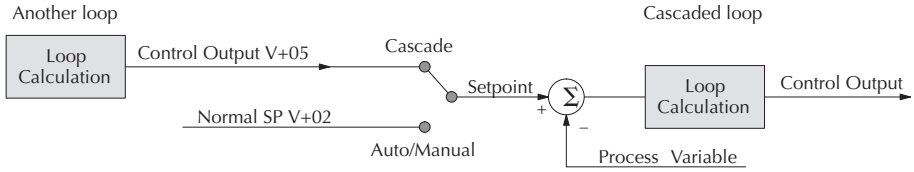


In Automatic Mode, the loop operates normally and generates new control output values. It calculates the PID equation and writes the result in location V+05 every sample period of that loop. The equivalent schematic diagram is shown below.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

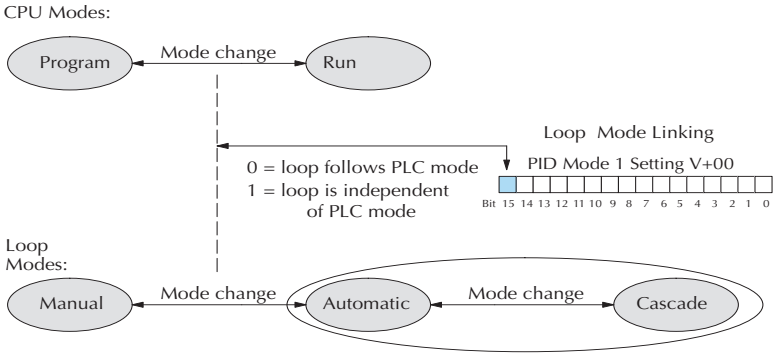
In Cascade Mode, the loop operates as it does in Automatic Mode, with one important difference. The data source for the SP changes from its normal location at V+02 to using the control output value, V+05, from another loop. So in Auto or Manual modes, the loop calculation uses the data at V+02. In Cascade Mode, the loop calculation reads the control output from another loop's parameter table, V+05.



As pictured below, a loop can be changed from one mode to another, but *cannot go from Manual Mode directly to Cascade, or vice versa*. This mode change is prohibited because a loop would be changing two data sources at the same time, and could cause a loss of control.



Once the CPU is operating in the Run Mode, the normal operation of the PID loop controller is to read the loop data and perform calculations on each scan of the RLL program. When the CPU is placed in the Program Mode, the RLL program halts operation and all PID loops are automatically put into the Manual Mode. The PID parameters can then be changed if desired. Similarly, by placing the CPU in the Run mode, the PID loops are returned to the operational mode which they were previously in, i.e., Manual, Automatic and Cascade. With this selection you automatically affect the modes by changing the CPU mode.



If bit 15 is set to 1, then the loops will run independent of the CPU mode. It is like having two independent processors in the CPU...one is running the RLL program and the other is running the process loops.

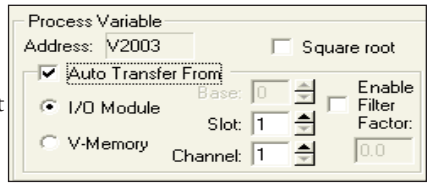
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Having the ability to run loops independently of the RLL program makes it feasible to make a ladder logic change while the process is still running. This is especially beneficial for large-mass continuous processes that are difficult or costly to interrupt. The Independent of CPU feature is used for this.

If you need to operate the PID loops while the RLL program is halted, in Program Mode, either select the Independent of CPU mode in the dialog or edit your program to set and reset bit 15 of PID Mode 1 word (V+00) in your RLL program. If the bit is set to a zero, the loop will follow the CPU mode, then when the CPU is placed in the Program Mode, all loops will be forced into the Manual Mode.

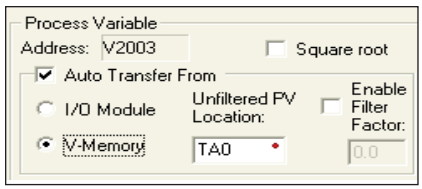
When Independent of CPU mode is used, you should also set the PV to be read directly from an analog input module. This can easily be done in the PID setup dialog, SP/PV.

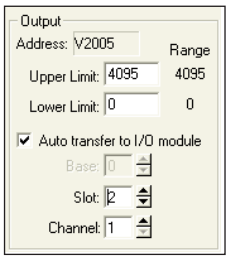
The SP/PV dialog has a block titled *Process Variable*. There is a block within this block called *Auto Transfer From* (from analog input) with the information grayed out. Checking the box to the left of the Auto Transfer From will highlight the information. Select *I/O Module*, then enter the slot number the input module resides in. Next, select the analog input channel of your choice.



The second choice is *V-Memory.* When this is selected, the V-memory address from where the PV is transferred from must be specified.

Whichever method of auto transfer is used, it is recommended to check the *Enable Filter Factor* (a low-pass filter) and specify the coefficient.





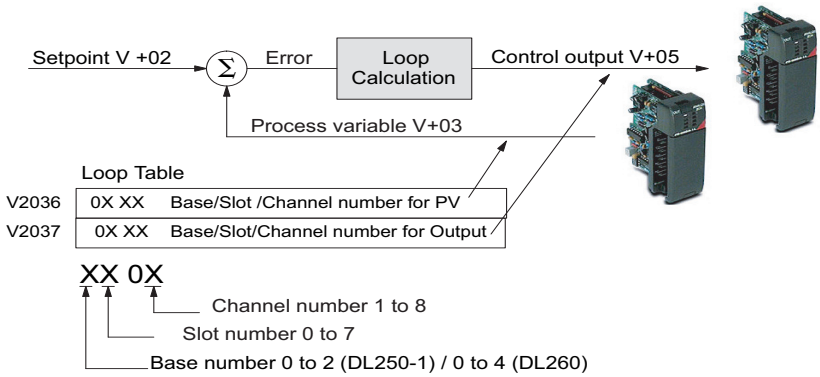
You should also select the analog output for the control output to be transferred to. This is done in the PID setup *Output* dialog shown here. The block of information in this dialog is "grayed-out" until the box next to *Auto transfer to I/O module* is checked. Once checked, enter the slot number where the output module is residing and then enter the analog output channel number.



***NOTE:** To make changes to any loop table parameters, the PID loop must be in Manual mode and the PLC must be stopped. If you have selected to operate the PID loop independent of the CPU mode, then you must take certain steps to make it possible to make loop parameter changes. You can temporarily make the loops follow the CPU mode by changing bit 15 to 0. Then you will be able to place the loop into Manual Mode using **Direct**SOFT. After you change the loop's parameter settings, just restore bit 15 to a value of 1 to re-establish PID operation independent of the CPU.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You may optionally configure each loop to access its analog I/O (PV and control output) by placing proper values in the associated loop table registers in your RLL program. The following figure shows the loop table parameters at V+36 and V+37 and their auto transfer role to access the analog values directly.



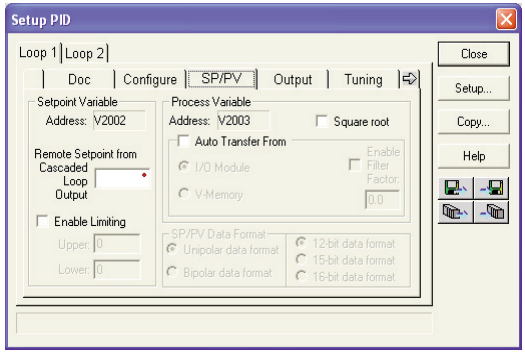
When these loop table parameters are programmed directly, a value of "0102" in register V2036 directs the loop controller to read the PV data from channel 1 of the analog input. A value of "0000" in either register tells the loop controller not to access the corresponding analog value directly. In that case, ladder logic must be used to transfer the value between the analog input and the loop table.



NOTE: When auto transfer to/from I/O is used, the analog data for all of the channels on the analog module cannot be accessed by any other method, i.e., pointer or multiplex.

SP/PV Addresses

An SP/PV dialog will be made available to setup how the setpoint (SP) and the process variable (PV) will be used in the loop. If this loop is the minor loop of a cascaded pair, enter that control output address in the *Remote SP from Cascaded Loop Output* area. It is sometimes desirable to limit the range of setpoint values allowed to be entered. To activate this feature, check the box next to *Enable Limiting*. This will activate the *Upper* and *Lower* fields for the values to be entered.



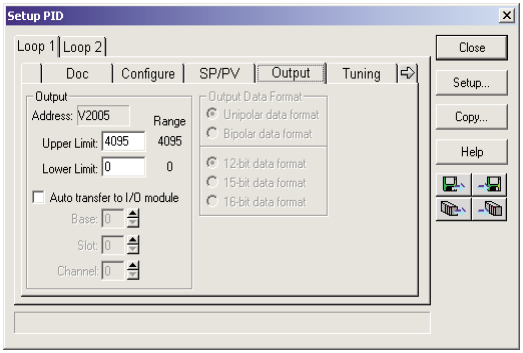
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Set the limits around the SP value to prevent an operator from entering a setpoint value outside of a safe range. The *Square root* box is only checked for certain PID loops, such as a flow control loop. If the *Auto transfer from I/O module* is selected, a first-order low-pass filter can be used by checking the *Enable Filter* box. The filter coefficient is user specified. The use of this filter is recommended during closed loop auto-tuning. If the Independent format had been checked previously, make the Data format selections here.

NOTE: The SP/PV dialog can be left as it first appears for basic PID operation.

Set Control Output Limits

Another dialog that will be available in the PID setup will be the Output dialog. The control output address, V+05, (determined by the PID loop table beginning address) will be in view. Enter the output range limits, *Upper Limit* and *Lower Limit,* that will meet the requirement of the process and which will agree with the data format that has been selected. For a basic PID operation using a 12-bit output module, set the Upper Limit to 4095 and leave the Lower Limit set to 0.



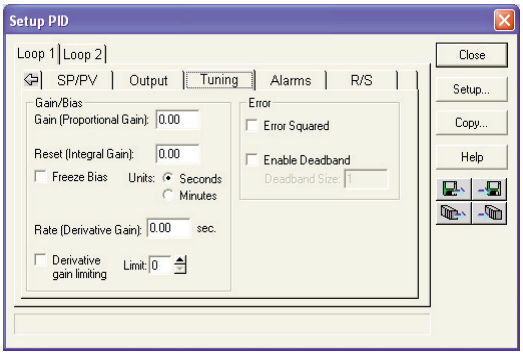
Check the box next for *Auto transfer to I/O module* if there is a need to send the control output to a certain analog output module, as in the case of using the Loop Mode independent of CPU Mode; otherwise, the PID output signal cannot control the analog output when the PLC is not in RUN Mode. If the *Auto transfer to I/O module* feature is checked, all channels of the module must be used for PID control outputs. If Independent format has been previously chosen, the *Output Data Format* will need to be set up here, that is, select Unipolar or Bipolar format and the bit structure. This area is not available and is grayed out if *Common format* has been chosen (see page 8-27).

WARNING: If the Upper Limit is set to zero, the output will never get above zero. In effect, there will be no control output.

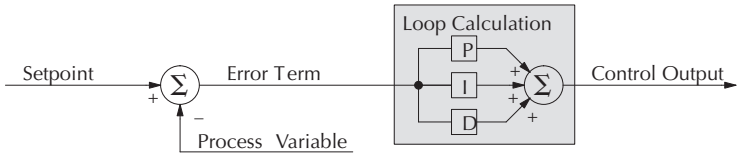
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Enter PID Parameters

Another PID setup dialog, Tuning, is for entering the PID parameters shown as: Gain (Proportional Gain), Reset (Integral Gain) and Rate (Derivative Gain).



Recall the position and velocity forms of the PID loop equations which were introduced earlier. The equations basically show the three components of the PID calculation: Proportional Gain (P), Integral Gain (I) and Derivative Gain (D). The following diagram shows a form of the PID calculation in which the control output is the sum of the proportional gain, integral gain and derivative gain. With each calculation of the loop, each term receives the same error signal value.



The P, I and D gains are 4-digit BCD numbers with values from 0000 to 9999. They contain an implied decimal point in the middle, so the values are actually 00.00 to 99.99. Some gain values have units – Proportional gain has no unit, Integral gain may be selected in seconds or in minutes, and Derivative gain is in seconds.

Gain (Proportional Gain) – This is the most basic gain of the three. Values range from 0000 to 9999, but they are used internally as xx.xx. An entry of "0000" effectively removes the proportional term from the PID equation. This accommodates applications which need integral-only loops.

Reset (Integral Gain) – Values range from 0001 to 9998, but they are used internally as xx.xx. An entry of "0000" or "9999"causes the integral gain to be "∞", effectively removing the integrator term from the PID equation. This accommodates applications which need proportional-only loops. The units of integral gain may be either seconds or minutes, as shown in the above dialog.

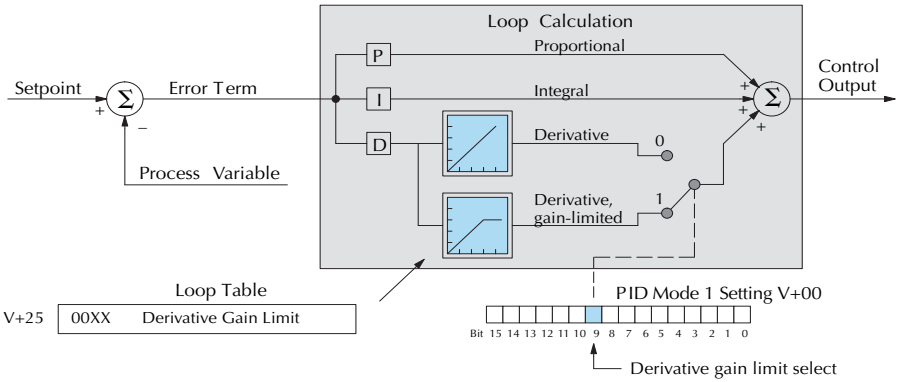
Rate (Derivative Gain) – Values which can be entered range from 0001 to 9999, but they are used internally as xx.xx. An entry of "0000" allows removal of the derivative term from the PID equation (a common practice). This accommodates applications which require only proportional and/or integral loops. Most control loops will operate as a PI loop.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

*NOTE: You may elect to leave the tuning dialog blank and enter the tuning parameters in the **Direct**SOFT PID View.*

Derivative Gain Limiting

The derivative gain (rate) has an optional gain-limiting feature. This is provided because the derivative gain reacts badly to PV signal noise or other causes of sudden PV fluctuations. The function of the gain-limiting is shown in the diagram below.



The gain limit can be particularly useful during loop tuning. Most loops can tolerate only a little derivative gain without going into uncontrolled oscillations.

If this option is checked, a *Limit* from 0 to 20 must also be entered.

*NOTE: When first configuring a loop, it's best to use the standard error term until after the loop is tuned. Once the loop is tuned, you will be able to tell if these functions will enhance control. The Error Squared and/or Enable Deadband can be selected later in the PID setup. Also, values are not required in the Tuning dialog, but can be set later in the **Direct**SOFT PID View.*

Error Term Selection

The error term is internal to the CPUs PID loop controller, and is generated again in each PID calculation. Although its data is not directly accessible, you can easily calculate it by subtracting: Error = (SP–PV). If the PV square-root extract is enabled, then: Error = $\sqrt{PV}$ In any case, the size of the error and algebraic sign determine the next change of the control output for each PID calculation.

Error Squared – When selected, the squared error function simply squares the error term (but preserves the original algebraic sign), which is used in the calculation. This affects the Control Output by diminishing its response to smaller error values, but maintaining its response to larger errors. Some situations in which the error squared term might be useful:

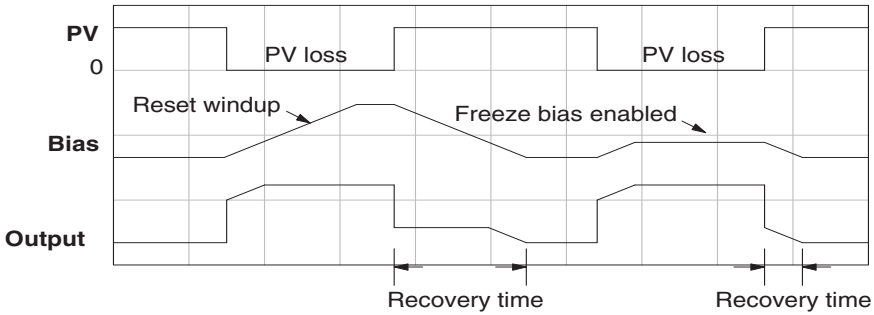
- Noisy PV signal – using a squared error term can reduce the effect of low-frequency electrical noise on the PV, which will make the control system jittery. A squared error maintains the response to larger errors.
- Non-linear process – some processes (such as chemical pH control) require non-linear controllers for best results. Another application is surge tank control, where the Control Output signal must be smooth.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Enable Deadband – When selected, the enable deadband function takes a range of small error values near zero, and simply substitutes zero as the value of the error. If the error is larger than the deadband range, then the error value is used normally.

Freeze Bias

The term *reset windup* refers to an undesirable characteristic of integrator behavior which occurs naturally under certain conditions. Refer to the figure below. Suppose the PV signal becomes disconnected, and the PV value goes to zero. While this is a serious loop fault, it is made worse by reset windup. Notice the bias (reset) term keeps integrating normally during the PV disconnect, until its upper limit is reached. When the PV signal returns, the bias value is saturated (windup) and takes a long time to return to normal. The loop output consequently has an extended recovery time. Until recovery, the output level is wrong and causes further problems.



In the second PV signal loss episode in the figure, the freeze bias feature is enabled. It causes the bias value to freeze when the control output goes out of bounds. Much of the reset windup is thus avoided, and the output recovery time is much less.

For most applications, the freeze bias feature will work with the loop as described above. It is suggested to enable this feature by selecting it in the dialog. Bit 10 of PID Mode 1 Setting (V+00) word can also be set in RLL.



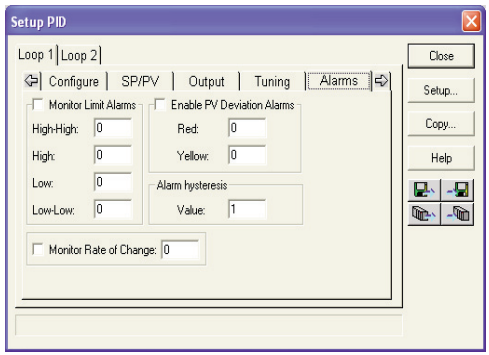
NOTE: The freeze bias feature stops the bias term from changing when the control output reaches the end of the data range. If you have set limits on the control output other than the range (i.e, 0–4095 for a unipolar/12-bit loop), the bias term still uses the end of range for the stopping point and bias freeze will not work.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setup the PID Alarms

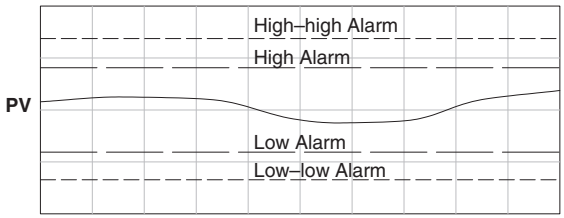
Although the setup of the PID alarms is optional, you surely would not want to operate a process without monitoring it. The performance of a process control loop may generally be measured by how closely the process variable matches the setpoint. Most process control loops in industry operate continuously, and will eventually lose control of the PV due to an error condition. Process alarms are vital in early discovery of a loop error condition and can alert plant personnel to manually control a loop or take other measures until the error condition has been repaired.

The alarm thresholds are fully programmable, and each type of alarm may be independently enabled and monitored. The following diagram shows the Alarm dialog in the PID setup which simplifies the alarm setup.



Monitor Limit Alarms

Checking this box will allow all of the PV limit alarms to be monitored once the limits are entered.The PV absolute value alarms are organized as two upper and two lower alarms. The alarm status is false as long as the PV value remains in the region between the upper and lower alarms, as shown below. The alarms nearest the safe zone are named *High Alarm* and *Low Alarm.* If the loop loses control, the PV will cross one of these thresholds first. Therefore, you can program the appropriate alarm threshold values in the loop table locations shown below to the right. The data format is the same as the PV and SP (12-bit or 15-bit). The threshold values for these alarms should be set to give an operator an early warning if the process loses control.



Loop Table

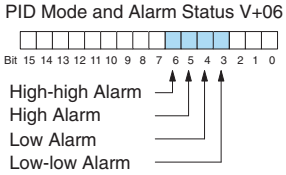
V+16	XXXX	High-high Alarm
V+15	XXXX	High Alarm
V+14	XXXX	Low Alarm
V+13	XXXX	Low-low Alarm

NOTE: The Alarm dialog can be left as it first appears, without alarm entries. The alarms can then be setup in the ***Direct****SOFT PID View.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

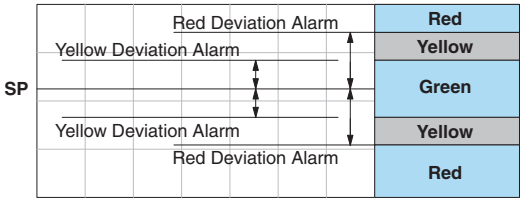
If the process remains out of control for some time, the PV will eventually cross one of the outer alarm thresholds, named High-high alarm and Low-low alarm. Their threshold values are programmed using the loop table registers listed above. A High-high or Low-low alarm indicates a serious condition exists, and needs the immediate attention of the operator.

The PV Absolute Value Alarms are reported in the four bits in the PID Mode and Alarm Status word in the loop table, as shown to the right. We highly recommend using ladder logic to monitor these bits. The bit-of-word instructions make this easy to do. Additionally, you can monitor PID alarms using *Direct*SOFT.



PV Deviation Alarms

The PV Deviation Alarms monitor the PV deviation with respect to the SP value. The deviation alarm has two programmable thresholds, and each threshold is applied equally above and below the current SP value. In the figure below, the smaller deviation alarm is called the "Yellow Deviation", indicating a cautionary condition for the loop. The larger deviation alarm is called the "Red Deviation", indicating a strong error condition for the loop. The threshold values use the loop parameter table locations V+17 and V+20 as shown.

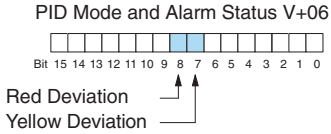


Loop Table

V+17	XXXX	Yellow Deviation Alarm
V+20	XXXX	Red Deviation Alarm

The thresholds define zones, which fluctuate with the SP value. The green zone which surrounds the SP value represents a safe (no alarm) condition. The yellow zones lie outside the green zone, and the red zones are beyond those.

The PV Deviation Alarms are reported in the two bits in the PID Mode and Alarm Status word in the loop table, as shown to the right. We highly recommend using ladder logic to monitor these bits. The bit-of-word instructions make this easy to do. Additionally, you can monitor PID alarms using *Direct*SOFT.



The PV Deviation Alarm can be independently enabled and disabled from the other PV alarms, using bit 13 of the PID Mode 1 Setting V+00 word.

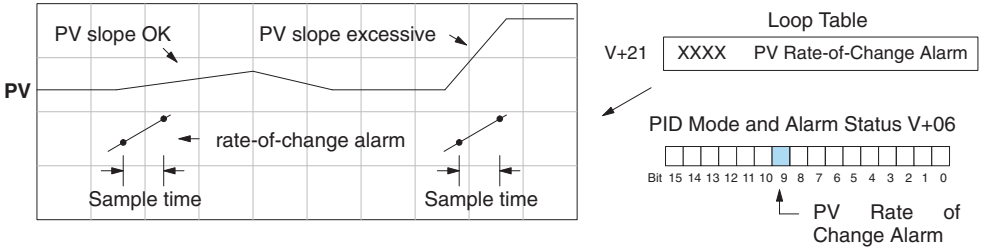
Remember the alarm hysteresis feature works in conjunction with both the deviation and absolute value alarms, and is discussed at the end of this section.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PV Rate-of-Change Alarm

An excellent way to get an early warning of a process fault is to monitor the *rate-of-change* of the PV. Most batch processes have large masses and slowly-changing PV values. A relatively fast-changing PV will result from a broken signal wire for either the PV or control output, a SP value error, or other causes. If the operator responds to a PV Rate-of-Change Alarm quickly and effectively, the PV absolute value will not reach the point where the material in process would be ruined.

The DL205 loop controller provides a programmable PV Rate-of-Change Alarm, as shown below. The rate-of-change is specified in PV units change per loop sample time. This value is programmed into the loop table location V+21.



As an example, suppose the PV is the temperature for your process, and you want an alarm whenever the temperature changes faster than 15 degrees/minute. The PV counts per degree and the loop sample rate must be known. Then, suppose the PV value (in V+03 location) represents 10 counts per degree, and the loop sample rate is 2 seconds. Use the formula below to convert our engineering units to counts/sample period:

$$\text{Alarm Rate-of-Change} = \frac{15 \text{ degrees}}{1 \text{ minute}} \times \frac{10 \text{ counts / degree}}{30 \text{ loop samples / min.}} = \frac{150}{30} = 5 \text{ counts}$$

From the calculation result, you would program the value 5 in the loop table for the rate-of-change. The PV Rate-of-Change Alarm can be independently enabled and disabled from the other PV alarms, using bit 14 of the PID Mode 1 Setting V+00 word.

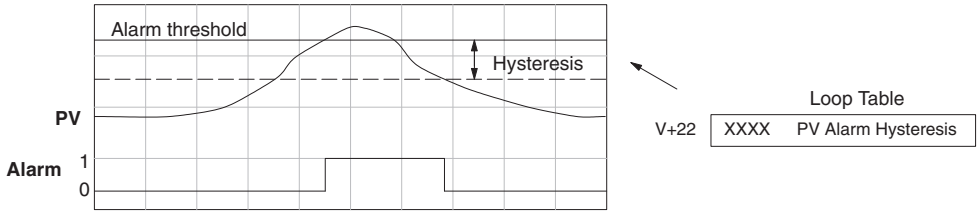
The alarm hysteresis feature (discussed next) does not affect the Rate-of-Change Alarm.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PV Alarm Hysteresis

The PV Absolute Value Alarm and PV Deviation Alarm are programmed using threshold values. When the absolute value or deviation exceeds the threshold, the alarm status becomes true. Real-world PV signals have some noise on them, which can cause some fluctuation in the PV value in the CPU. As the PV value crosses an alarm threshold, its fluctuations cause the alarm to be intermittent and annoy process operators. The solution is to use the PV Alarm Hysteresis feature.

The PV Alarm Hysteresis amount is programmable from 1 to 200 (binary/decimal). When using the PV Deviation Alarm, the programmed hysteresis amount must be less than the programmed deviation amount. The figure below shows how the hysteresis is applied when the PV value goes past a threshold and descends back through it.

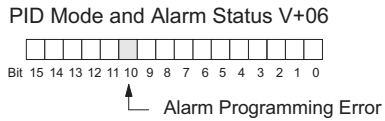


The hysteresis amount is applied after the threshold is crossed, and toward the safe zone. In this way, the alarm activates immediately above the programmed threshold value. It delays turning off until the PV value has returned through the threshold by the hysteresis amount.

Alarm Programming Error

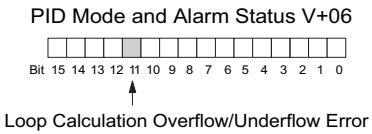
The PV Alarm threshold values must have certain mathematical relationships to be valid. The requirements are listed below. If not met, the Alarm Programming Error bit will be set, as indicated to the right.

- PV Absolute Alarm value requirements:
  Low-low < Low < High < High-high
- PV Deviation Alarm requirements:
  Yellow < Red



Loop Calculation Overflow/Underflow Error

This error occurs whenever the output reaches it's upper or lower limit and the PV does not reach the setpoint. A typical example might be when a valve is stuck, the output is at it's limit, but the PV has not reached setpoint.



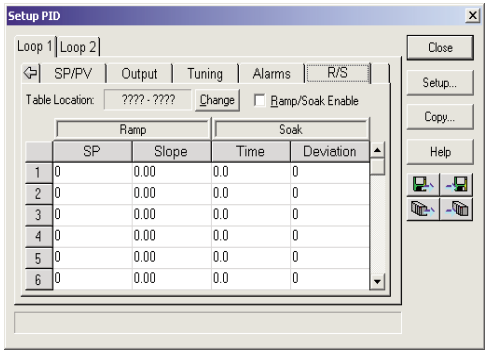


NOTE: Overflow/underflow can be alarmed in PID View. The optional C-more operator interface panel (see the automationdirect.com website) can also be setup to read these error bits using the PID Faceplate templates.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Ramp/Soak

R/S (Ramp/Soak) is the last dialog available in the PID setup. The basic PID does not require any entries to be made in order to operate the PID loop. Ramp/Soak will be discussed in another section.

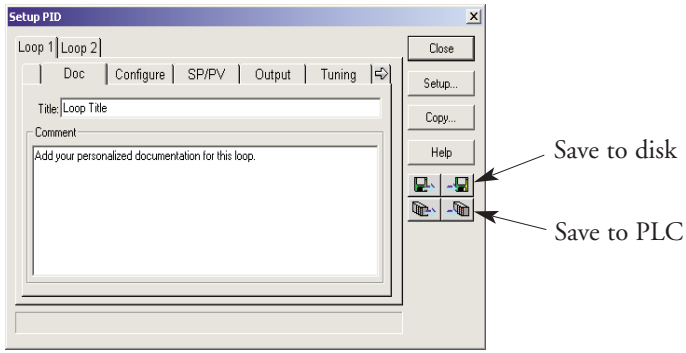


8

Complete the PID Setup

Once you have filled in the necessary information for the basic PID setup, the configuration should be saved. The icons on the Setup PID dialog will allow you to save the configuration to the PLC and to disk. The save to icons have the arrow pointing to the PLC and disk. The read from icons have the arrows pointing away from the PLC and disk.

An optional feature is available with the Doc tab in the Setup PID window. You enter a name and description for the loop. This is useful if there are more than one PID loop in your application.



NOTE: It is good practice to save your project after setting up the PID loop by selecting ***File*** *from the menu toolbar, then* ***Save project > to disk****. In addition to saving your entire project, all the PID parameters are also saved.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Loop Tuning

Once you have set up a PID loop, it must be tuned in order for it to work. The goal of loop tuning is to adjust the loop gains so the loop has optimal performance in dynamic conditions. The quality of a loop's performance may generally be judged by how well the PV follows the SP after a SP step change. It is important to keep in mind that understanding the process is fundamental to getting a well designed control loop. Sensors must be in appropriate locations and valves must be sized correctly with appropriate trim. **PID control does not have *typical* values**. There isn't one control process that is identical to another.

Manual Tuning vs. Auto Tuning

You may enter the PID gain values to tune your loops (manual tuning), or you can rely on the PID processing "engine" in the CPU to automatically calculate the gain values (auto tuning). Most experienced process engineers will have a favorite method; the DL205 will accommodate either preference. The use of auto tuning can eliminate much of the trial-and-error of the manual tuning approach, especially if you do not have a lot of loop tuning experience. However, performing the auto tuning procedure will get the gains close to optimal values, but additional manual tuning can get the gain values to their optimal values.



WARNING: Only authorized personnel fully familiar with all aspects of the process should make changes that affect the loop tuning constants. Using the loop auto tune procedures will affect the process, including inducing large changes in the control output value. Make sure you thoroughly consider the impact of any changes to minimize the risk of injury to personnel or damage to equipment. The auto tune in the DL205 is not intended to be used as a replacement for your process knowledge.

8

Open-Loop Test

Whether you use manual or auto tuning, it is very important to verify basic characteristics of a newly-installed process before attempting to tune it. With the loop in Manual Mode, verify the following items for each new loop.

- **Setpoint** – verify that the setpoint (SP) source can generate a setpoint. Put the PLC in Run Mode and leave the loop in Manual Mode, then monitor the loop table location V+02 to see the SP value(s). (If you are using the Ramp/Soak generator, test it now).
- **Process Variable** – verify that the PV value is an accurate measurement, and the PV data arriving in the loop table location V+03 is correct. If the PV signal is very noisy, consider filtering the input either through hardware (RC low-pass filter), or using the filter in this chapter.
- **Control Output** – if it is safe to do so, manually change the output a small amount (perhaps 10%) and observe its affect on the process variable. Verify the process is direct-acting or reverse acting, and check the setting for the control output (inverted or non-inverted). Make sure the control output upper and lower limits are not equal to each other.
- **Sample Rate** – while operating open-loop, this is a good time to find the ideal sample rate (see Configure the PID Loop earlier in this chapter). However, if you are going to use auto tuning, the auto tuning procedure will automatically calculate the sample rate in addition to the PID gains.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

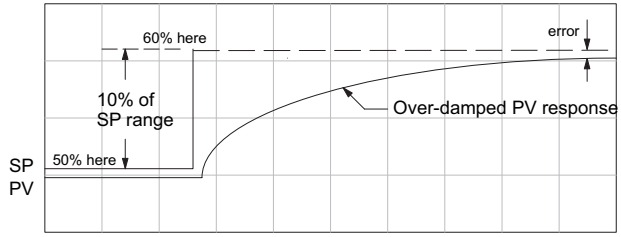
Manual Tuning Procedure

It is not necessary to try to obtain the best values for the P, I and D parameters in the PID loop by trial and error. Following is a typical procedure for tuning a temperature control loop which you may use to tune your loop.

Monitor the values of SP, PV and CV with a loop trending instrument or use the PID View feature in *Direct*SOFT (see page 8-49).

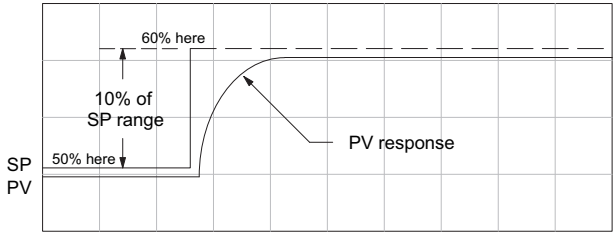
NOTE: We recommend using the PID View Tuning and Trending window to select manual for the vertical scale feature, for both SP/PV area and Bias/Control Output areas. The auto scaling feature would otherwise change the vertical scale on the process parameters and add confusion to the loop tuning process.

- Adjust the gains so the Proportional Gain = 0.5 or 1.0 (1.0 is a good value based on experience), Integral Gain = 9999 (this basically eliminates reset) and Derivative Gain =0000. This disables the integrator and derivative terms, and provides some proportional gain.
- Check the bias value in the PID View and set it to zero.
- Set the SP to a value equal to 50% of the full range.
- Now, select Auto Mode. If the loop will not stay in Auto Mode, check the troubleshooting tips at the end of this chapter. Allow the PV to stabilize around the 50% point of the range.
- Change the SP to the 60% point of the range.



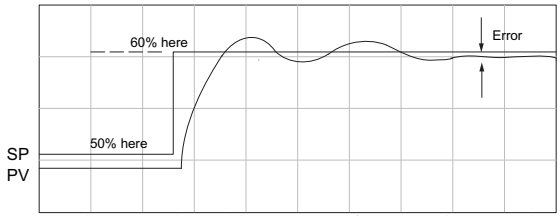
The response may take awhile, but you will see that there isn't any oscillation. This response is not desirable since it takes a long time to correct the error; also, there is a difference between the SP and the PV.

- Increase the Proportional gain, for example to 2.0. The control output will be greater and the response time will be quicker. The trend should resemble the figure below.

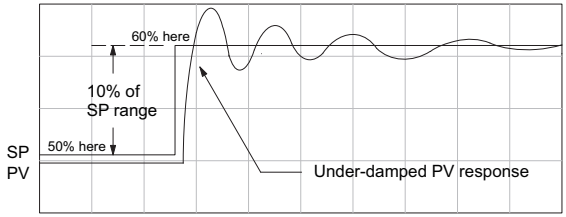


- Increase the Proportional gain in small increments, such as 4, 6, 7, etc. until the control output response begins to oscillate. This is the Proportional gain that should be recorded.

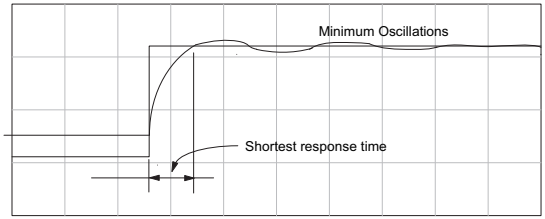
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



- Now, return the Proportional gain to the stable response, for example, 9.7. The error, SP-PV, should be small, but not at zero.
- Next, add a small amount of Integral gain (reset) in order for the error to reach zero. Begin by using 80 seconds (adjust in minutes if necessary). The error should get smaller.
- Set the Integral gain to a lower value, such as 50 for a different response. If there is no response, continue to decrease the reset value until the response becomes unstable. See the figure below.



- For discussion, let us say that a reset value of 35 made the control output unstable. Return the reset value to the stable value, such as 38. **Be careful with this adjustment since the oscillation can destroy the process.**
- The control output response should be optimal now, without a Derivative gain. The example recorded values are: Proportional gain = 9.7 and Integral gain = 38 seconds. Note that the error has been minimized.



The foregone method is the most common method used to tune a PID loop. Derivative gain is almost never used in a temperature control loop. This method can also be used for other control loops, but other parameters may need to be added for a stable control output.

Test your loop for a high PV of 80% and again for a low PV.of 20%, and correct the values if necessary. Small adjustments of the parameters can make the control output more precise or more unstable. It is sometimes acceptable to have a small overshoot to make the control output react quicker.

The derivative gain can be helpful for those control loops which are not controlling temperature. For these loops, try adding a value of 0.5 for the derivative gain and see if this improves the control output. If there is little or no response, increase the derivative by increments of 0.5 until there is an improvement to the output trend. Recall that the derivative gain reacts with a rate of change of the error.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Alternative Manual Tuning Procedures by Others

The following tuning procedures have been extracted from various publications about PID process control. These procedures are for comparison to the procedure in this manual.

Tuning PID Controllers

Two-Mode Simple Method - – for P-I controllers

1. Turn off reset and set the gain to a small value (0.5 - 1.0).
2. Increase gain until cycling starts, then decrease gain slightly.
3. Make setpoint changes to observe offset (error).
4. Increase reset to eliminate offset (error).
5. Repeat steps 2 through 4 until you obtain the largest gain and reset consistent with the criteria of the control desired, i.e., offset, overshoot, stability.

Zeigler-Nichols Method– "Quarter amplitude decay"

1. Turn off reset and rate; set the proportional gain to a fairly large value.
2. Make a small setpoint change and observe how the controlled variable cycles.
3. Adjust the gain until the cycle is self-sustaining, and of constant amplitude; this value is the ultimate gain (Gu).
4. Measure the period of cycling in minutes. This is the ultimate period (Pu).
5. Calculate the controller adjustments as follows:

P only:	G = Gu/2	
P & I :	G = Gu/2.2	
	Ti = 1.2/Pu	(repeats/minute)
P-I-D:	G = Gu/1.6	
	Ti = 2.0/Pu	(repeats/minute)
	Td = Pu/8.0	(minutes)

Pessen Method

1. Follow the procedure described above (Zeigler-Nichols) to determine the ultimate gain and ultimate period.
2. Apply the formulas below.

For no overshoot during startup:

G = Gu/5.0	
Ti = 3/Pu	(repeats/minute)
Td = Pu/2	(minutes)

For some overshoot, but better response to disturbances:

G = Gu/3	
Ti = 3/Pu	(repeats/minute)
Td = Pu/3	(minutes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Auto Tuning Procedure

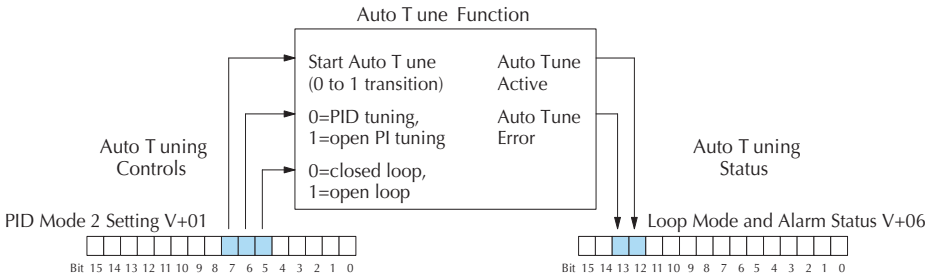
The auto tuning feature for the DL205 loop controller will only run once each time it is enabled in the PID table. Therefore, auto tuning does not run continuously during operation (this would be *adaptive* control). Whenever there is a substantial change in loop dynamics, such as mass of process, size of actuator, etc., the tuning process will need to be repeated in order to derive new gains required for optimal control.

WARNING: Only authorized personnel fully familiar with all aspects of the process should make changes that affect the loop tuning constants. Using the loop auto tuning procedures will affect the process, including inducing large changes in the control output value. Make sure you thoroughly consider the impact of any changes to minimize the risk of injury to personnel or damage to equipment. The auto tune in the DL205 is not intended to be used as a replacement for your process knowledge.

Once the physical loop components are connected to the PLC, auto tuning can be initiated within ***Direct***SOFT (see the ***Direct***SOFT Programming Software Manual), and it can be used to establish initial PID parameter values. Auto tuning is the best "guess" the CPU can do after some trial tests.

The loop controller offers both closed-loop and open-loop methods. The following sections describe how to use the auto tuning feature, and what occurs in open and closed-loop auto tuning.

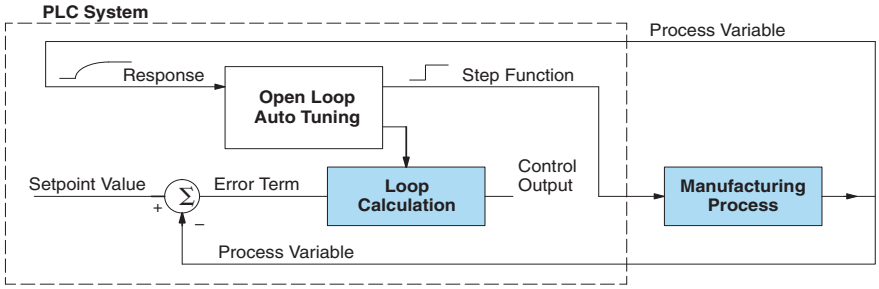
The controls for the auto tuning function use three bits in the PID Mode 2 word V+01, as shown below. ***Direct***SOFT will manipulate these bits automatically when you use the auto tune feature within ***Direct***SOFT. Or, you may have your ladder logic access these bits directly for allowing control from another source such as a dedicated operator interface. The individual control bits allow you to start the auto tune procedure, select PID or PI tuning and select closed-loop or open-loop tuning. If you select PI tuning, the auto tune procedure leaves the derivative gain at 0. The Loop Mode and Alarm Status word V+06 reports the auto tune status as shown. Bit 12 will be on (1) during the auto tune cycle, automatically returning to off (0) when done.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Open-Loop Auto Tuning

During an open-loop auto tuning cycle, the loop controller operates as shown in the diagram below. Before starting this procedure, place the loop in Manual Mode and ensure the PV and control output values are in the middle of their ranges (away from the end points).



NOTE: In theory, the SP value does not matter in this case, because the loop is not closed. However, the requirement of the firmware is that the SP value must be more than 5% of the PV range from the actual PV before starting the auto tune cycle (for the DL205, 12 bit PV should be 205 counts or more below the SP for forward-acting loops, or 205 counts or more above the SP for reverse-acting loops).

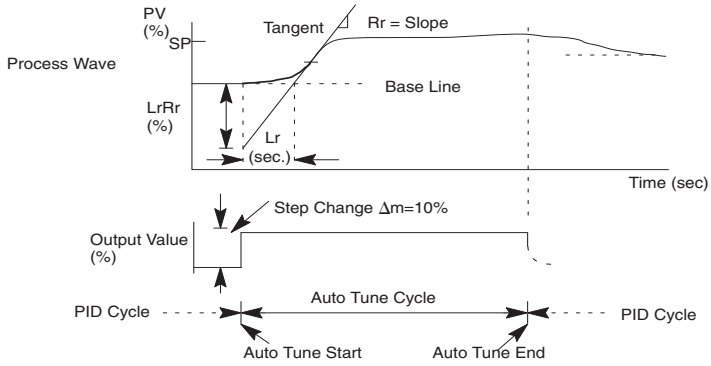
8

When auto tuning, the loop controller induces a step change on the output and simply observes the response of the PV. From the PV response, the auto tune function calculates the gains and the sample time. It automatically places the results in the corresponding registers in the loop table.

The following timing diagram shows the events which occur in the open-loop auto tuning cycle. The auto tune function takes control of the control output and induces a 10%-of-span step change. If the PV change which the loop controller observes is less than 2%, then the step change on the output is increased to 20%-of-span.

* When Auto Tune starts, step change output m=10%
* During Auto Tune, the controller output reached the full scale positive limit. Auto Tune stopped and the Auto Tune Error bit in the Alarm word bit turned on.
* When PV change is under 2%, output is changed at 20%.

Open Loop Auto Tune Cycle Wave: Step Response Method



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

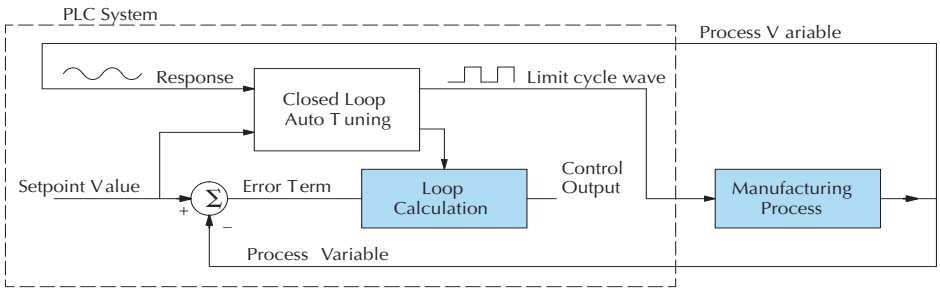
When the loop tuning observations are complete, the loop controller computes Rr (maximum slope in %/sec.) and Lr (dead time in sec). The auto tune function computes the gains according to the Zeigler-Nichols equations, shown below:

PID Tuning	PI Tuning
P=1.2*Δm/LrRr	P=0.9*Δm/LrRr
I=2.0* Lr	I=3.33* Lr
D=0.5* Lr	D=0
Sample Rate = 0.056* Lr	Sample Rate = 0.12*Lr
Δm = Output step change (10% = 0.1, 20% = 0.2)	

We highly recommend using *Direct*SOFT for the auto tuning interface. The duration of each auto tuning cycle will depend on the mass of the process. A slowly-changing PV will result in a longer auto tune cycle time. When the auto tuning is complete, the proportional, integral, and derivative gain values are automatically updated in loop table locations V+10, V+11, and V+12 respectively. The sample time in V+07 is also updated automatically. You can test the validity of the values the auto tuning procedure yields by measuring the closed-loop response of the PV to a step change in the output. The instructions on how to do this are in the section on the manual tuning procedure (located prior to this auto tuning section).

Closed-Loop Auto Tuning

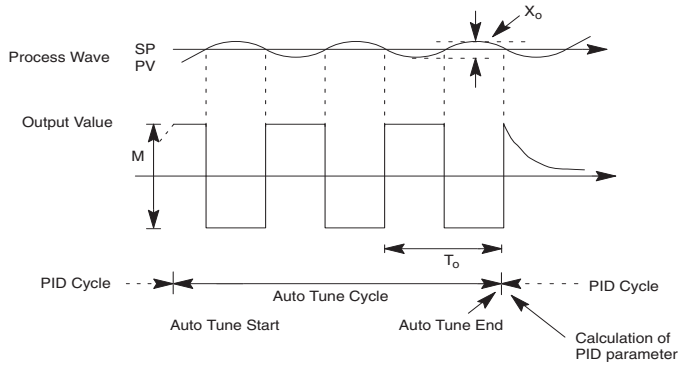
During a closed-loop auto tuning cycle the loop controller operates as shown in the diagram below.



When auto tuning, the loop controller imposes a square wave on the output. Each transition of the output occurs when the PV value crosses over/under the SP value. Therefore, the frequency of the limit cycle is roughly proportional to the mass of the process. From the PV response, the auto tune function calculates the gains and the sample time. It automatically places the results in the corresponding registers in the loop table.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following timing diagram shows the events which occur in the closed-loop auto tuning cycle. The auto tune function examines the direction of the offset of the PV from the SP. The auto tune function then takes control of the control output and induces a full-span step change in the opposite direction. Each time the sign of the error (SP – PV) changes, the output changes full-span in the opposite direction. This proceeds through three full cycles.



Mmax = Output Value upper limit setting. Mmin = Output Value lower limit setting.

This example is direct–acting.

When set to reverse–acting, the output will be inverted. When the loop tuning observations are complete, the loop controller computes To (bump period) and Xo (amplitude of the PV). Then it uses these values to compute Kpc (sensitive limit) and Tpc (period limit). From these values, the loop controller auto tune function computes the PID gains and the sample rate according to the Zeigler-Nichols equations shown below:

Kpc = 4M / (π *Xo) Tpc = 0	
M = Amplitude of output	
PID Tuning	**PI Tuning**
P = 0.45*Kpc	P = 0.30*Kpc
I = 0.60*Tpc	I = 1.00*Tpc
D = 0.10*Tpc	D = 0
Sample Rate = 0.014*Tpc	Sample Rate = 0.03*Tpc

Auto tuning error

In open-loop tuning, if the auto tune error bit (bit 13 of loop Mode/Alarm status word V+06) is on, please verify the PV and SP values are within 5% of full scale difference, as required by the auto tune function.



NOTE: If your PV fluctuates rapidly, you probably need to use the built-in analog filter (see page 8–54) or create a filter in ladder logic (see example on page 8–55).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

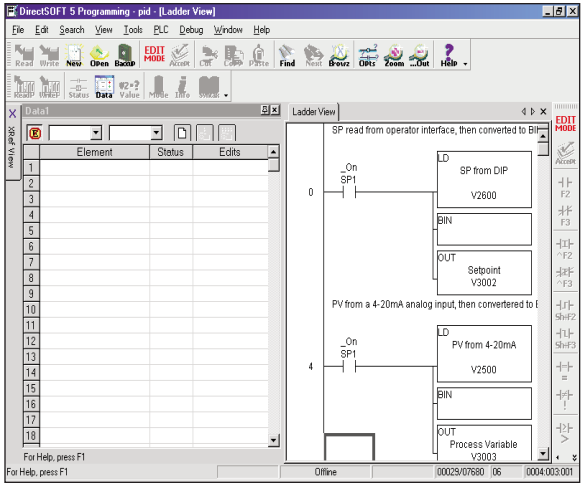
Use *Direct*SOFT Data View with PID View

The **Data View** window is a very useful tool which can be used to help tune your PID loop. You can compare the variables in the **PID View** with the actual values in the V-memory location with Data View.

Open a New Data View Window



A new Data View window can be opened in any one of three ways; the menu bar **Debug > Data View > New**, the keyboard shortcut **Ctrl + Shift + F3** or the **Data** button on the Status toolbar. By default, the Data View window is assigned Data1 as the default name. This name can be changed for the current view using the Options dialog. The following diagram is an example of a newly opened Data View. The window will open next to the Ladder View by default.



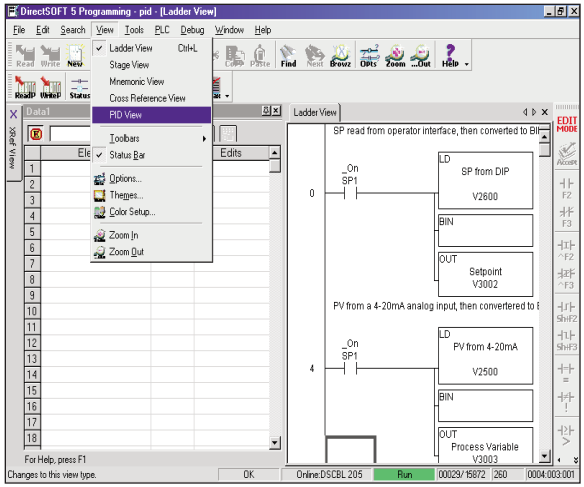
The Data View window can be used just as it is shown above for troubleshooting your PID logic, and it can be most useful when tuning the PID loop.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

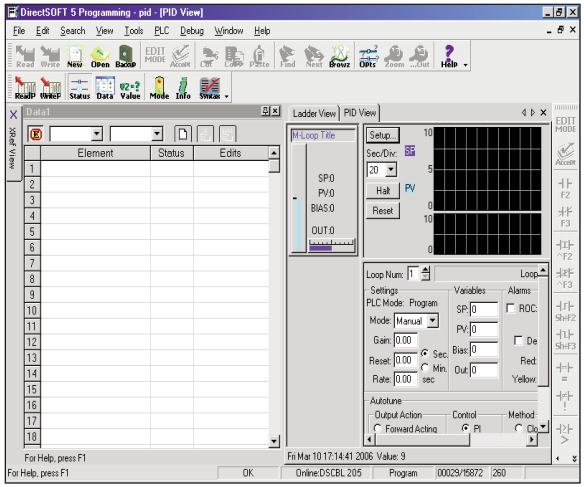
Open PID View



The PID View can only be opened after a loop has been setup in your ladder program. PID View is opened by selecting it from the View submenu on the Menu bar, **View > PID View**. The PID View can also be opened by clicking on the PID View button from the PLC Setup toolbar if it is in view.

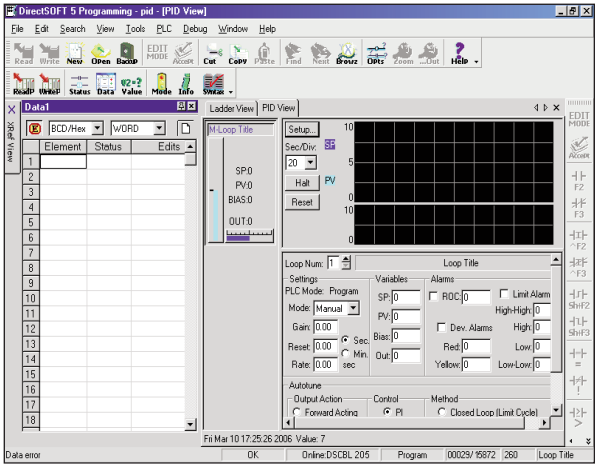


The PID View will open and appear over the Ladder View which can be brought into view by clicking on it's tab. When using the Data View and the PID View together, each view can be sized for better use as shown on the facing page.

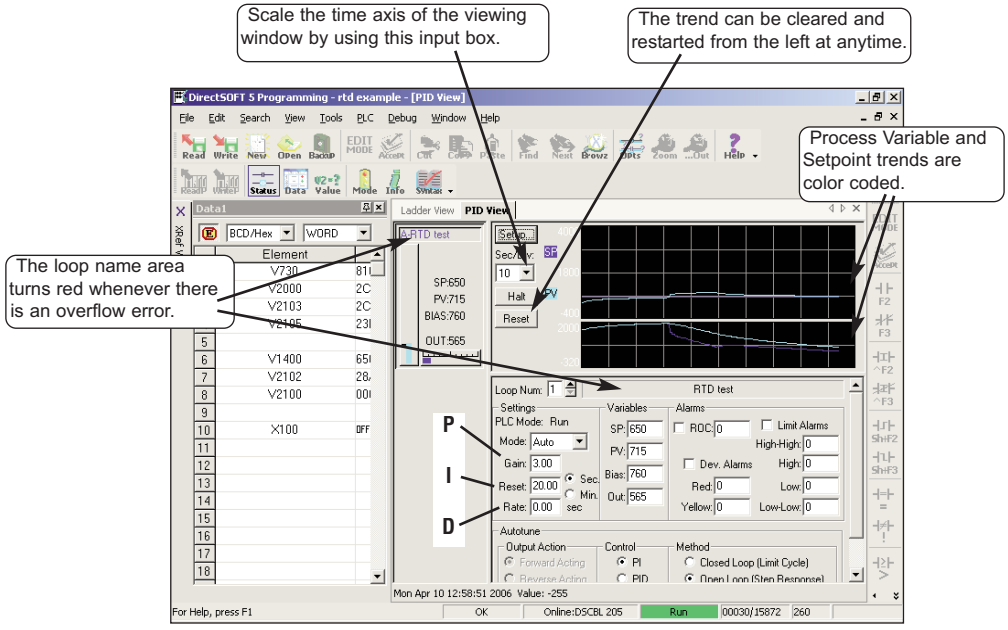


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The two views are now ready to be used to tune your loop. You will be able to see where the PID values have been set and see the process that it is controlling.



With both windows positioned in this manner, you are able to see where the PID values have been set and see the process that it is controlling. In the diagram below, you can see the current SP, PV and Output values, along with the other PID addresses. Refer to the Loop Table Word Definitions (page 8-20) for details for each word in the table. This is also a good data type reference for each word in the table.



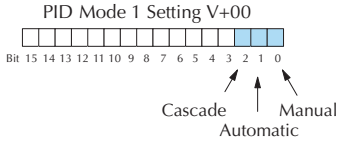
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the Special PID Features

It's a good idea to understand the special features of the DL205 and how to use them. You may want to incorporate some of these features for your PID.

How to Change Loop Modes

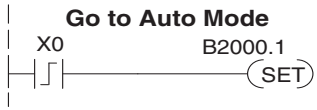
The first three bits of the PID Mode 1 word (V+00) request the operating mode of the corresponding loop. Note: these bits are mode change *requests*, not commands (certain conditions can prohibit a particular mode change – see next page).



The normal state of these mode request bits is "000". To request a mode change, you must SET the corresponding bit to a "1", using a one-shot. The PID loop controller automatically resets the bits back to "000" after it reads the mode change request. Methods of requesting mode changes are:

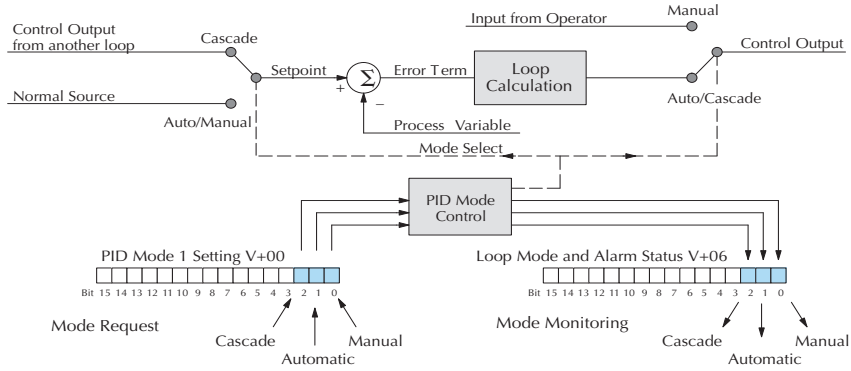
- ***Direct***SOFT's **PID View** – this is the easiest method. Use the pull-down menu, or click on one of the radio buttons if using older ***Direct***SOFT versions, and the appropriate bit will get set.
- **Ladder program**– ladder logic can request any loop mode when the PLC is in Run Mode. This will be necessary after application startup if mode changes are part of the application.

Use the program shown to the right to SET the mode bit (do not use an OUT coil). On a 0–1 transition of X0, the rung sets the Auto bit equal to 1. The loop controller resets it.



- **Operator panel** – interface the operator's panel to ladder logic using standard methods, then use the logic to the right to set the mode bit.

Since mode changes can only be *requested*, the PID loop controller will decide when to permit mode changes and provide the loop mode status. It reports the current mode on bits 0, 1, and 2 of the Loop Mode/Alarm Status word, location V+06 in the loop table. The parallel request/monitoring functions are shown in the figure below. The figure also shows the two possible mode-dependent SP sources, and the two possible Control Output sources.

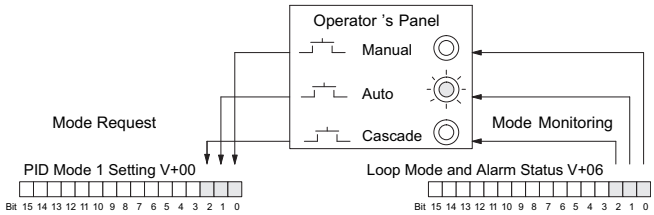


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Operator Panel Control of PID Modes

Since the modes Manual, Auto and Cascade are the most fundamental and important PID loop controls, you may want to "hard-wire" mode control switches to an operator's panel. Most applications will need only Manual and Auto selections (Cascade is used in special applications). Remember that mode controls are really *mode request* bits, and the actual loop mode is indicated elsewhere.

The following figure shows an operator's panel using momentary push-buttons to request PID mode changes. The panel's mode indicators do not connect to the switches, but interface to the corresponding data locations.



PLC Modes Effect on Loop Modes

If you have selected the option for the loops to follow the PLC mode, the PLC modes (Program, Run) interact with the loops as a group. The following summarizes this interaction:

- When the PLC is in Program Mode, all loops are placed in Manual Mode and no loop calculations occur. However, note that output modules (including analog outputs) turn off in PLC Program Mode. So, actual manual control is not possible when the PLC is in Program Mode.
- The only time the CPU will allow a loop mode change is during PLC Run Mode operation. As such, the CPU records the modes of all 4 loops as the desired mode of operation. If power failure and restoration occurs during PLC Run Mode, the CPU returns all loops to their prior mode (which could be Manual, Auto, or Cascade).
- On a Program-to-Run mode transition, the CPU forces each loop to return to its prior mode recorded during the last PLC Run Mode.
- You can add and configure new loops only when the PLC is in Program Mode. New loops automatically begin in Manual Mode.

Loop Mode Override

In normal conditions the mode of a loop is determined by the request to V+00, bits 0, 1, and 2. However, some conditions exist which will prevent a requested mode change from occurring:

- A loop that is not set independent of PLC mode cannot change modes when the PLC is in Program mode.
- A major loop of a cascaded pair of loops cannot go from Manual to Auto until its minor loop is in Cascade mode.

In other situations, the PID loop controller will automatically change the mode of the loop to ensure safe operation:

- A loop which develops an error condition automatically goes to Manual.
- If the minor loop of a cascaded pair of loops leaves Cascade Mode for any reason, its major loop automatically goes to Manual Mode.

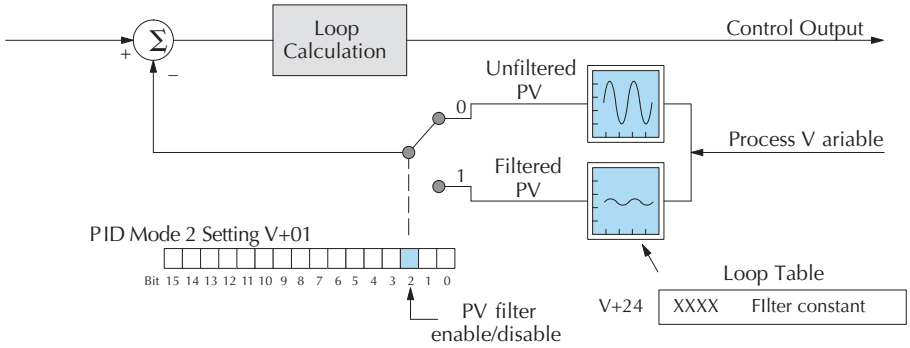
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PV Analog Filter

A noisy PV signal can make tuning difficult and can cause the control output to be more extreme than necessary, as the output tries to respond to the peaks and valleys of the PV. There are two equivalent methods of filtering the PV input to make the loop more stable. The first method is accomplished using the DL205's built-in filter. The second method achieves a similar result using ladder logic.

The DL205 Built-in Analog Filter

The DL205 provides a selectable first-order low-pass PV input filter. **We only recommend the use of a filter during auto tuning or PID control if there is noise on the input signal.** You may disable the filter after auto tuning is complete, or continue to use it if the PV input signal is noisy.



Bit 2 of PID Mode Setting 2 provides the enable/disable control for the low-pass PV filter (0=disable, 1=enable). The roll-off frequency of the single-pole low-pass filter is controlled by using register V+24 in the loop parameter table, the filter constant. The data format of the filter constant value is BCD, with an implied decimal point 00X.X, as follows:

- The filter constant has a valid range of 000.1 to 001.0. **The smaller the filter value, the greater the filtering performed (for example, the value 001.0 provides no filtering.)**
- ***Direct***SOFT converts values above the valid range to 001.0 and values below this range to 000.1
- Values close to 001.0 result in higher roll-off frequencies, while values closer to 000.1 result in lower roll-off frequencies.

We highly recommend using ***Direct***SOFT for the auto tuning interface. The duration of each auto tuning cycle will depend on the mass of your process. A slowly-changing PV will result in a longer auto tune cycle time.

When the auto tuning is complete, the proportional and integral gain values are automatically updated in loop table locations V+10 and V+11 respectively. The derivative is calculated if you autotune for PID and updated in loop table location V+12. The sample time in V+07 is also updated automatically. You can test the validity of the values the auto tuning procedure yields by measuring the closed-loop response of the PV to a step change in the output. The instructions on how to do this are in the section on the manual tuning procedure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The algorithm which the built-in filter follows is:

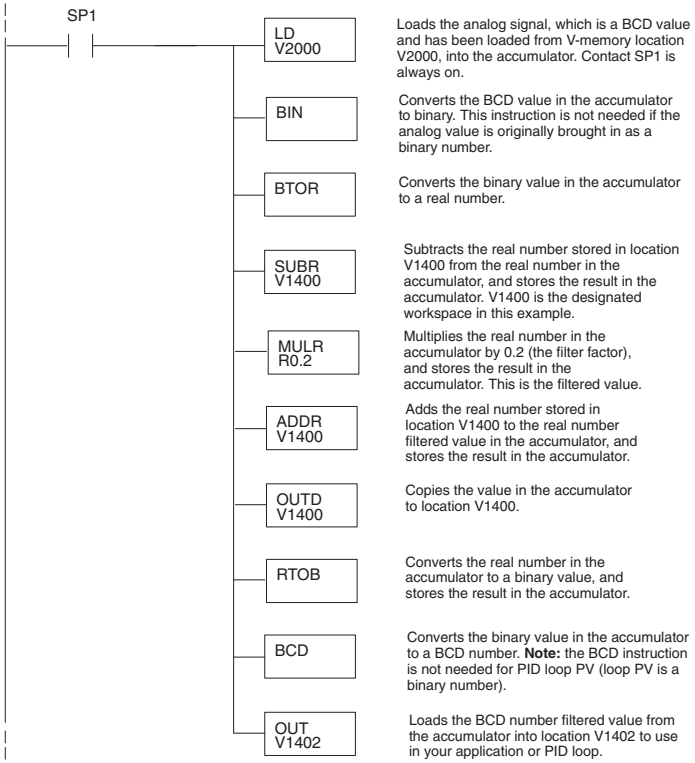
$$yi = k\,(xi - yi{-}1) + yi{-}1$$

yi is the current output of the filter
xi is the current input to the filter
yi–1 is the previous output of the filter
k is the PV Analog Input Filter Factor

Creating an Analog Filter in Ladder Logic

A similar algorithm can be built in your ladder program. Your analog inputs can be filtered effectively using either method. The following programming example describes the ladder logic you will need. Be sure to change the example memory locations to those that fit your application.

Filtering can induce a 1 part in 1000 error in your output because of "rounding." If your process cannot tolerate a 1 part in 1000 error, do not use filtering. Because of the rounding error, you should not use zero or full scale as alarm points. Additionally, the smaller the filter constant the greater the smoothing effect, but the slower the response time. Be sure a slower response is acceptable in controlling your process.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Use the *Direct*SOFT Filter Intelligent Box Instruction

For those who are using ***Direct*SOFT**, you have the opportunity to use Intelligent Box (IBox) instruction IB-402, Filter Over Time in Binary (decimal). This IBox will perform a first-order filter on the Raw Data on a defined time interval. The equation is,

New = Old + [(Raw - Old) / FDC] where:

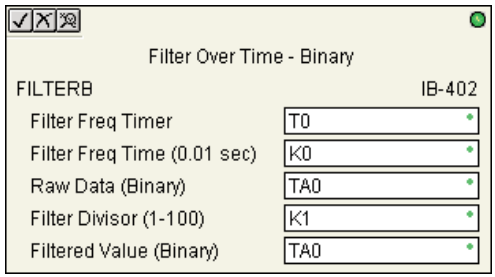
New =New Filtered Value

Old = Old Filtered Value

FDC = Filter Divisor Constant

Raw = Raw Data

The Filter Divisor Constant is an integer in the range K1 to K100, such that if it equaled K1, then no filtering is performed.

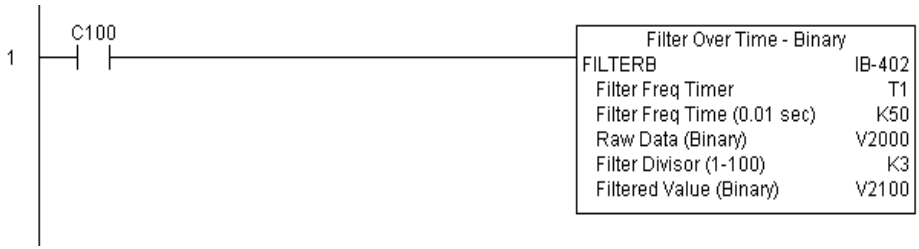


The rate at which the calculation is performed is specified by time in hundreths of a second (0.01 seconds) as the Filter Freq Time parameter. Note that this Timer instruction is embedded in the IBox and must NOT be used any other place in your program. Power flow controls whether the calculation is enabled. If it is disabled, the Filter Value is not updated. On the first scan from Program to Run mode, the Filter Value is initialized to 0 to give the calculation a consistent starting point.

Since the following binary filter example does not write directly to the PID PV location, the BCD filter could be used with BCD values and then converted to BIN.

FilterB Example

Following is an example of how the FilterB IBox is used in a ladder program. The instruction is used to filter a binary value that is in V2000. Timer (T1) is set to 0.5 seconds, the rate at which the filter calculation will be performed. The filter constant is set to 3.0. A larger value will increase the smoothing effect of the filter. A value of 1 results with no filtering. The filtered value will be placed in V2100.



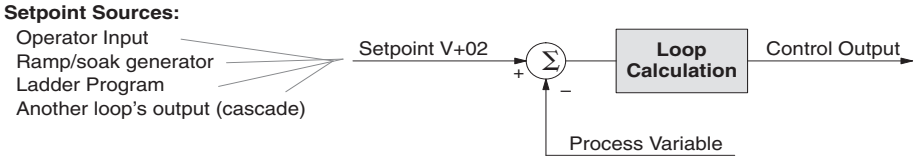
See Chapter 5, page 242, for more detailed information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Ramp/Soak Generator

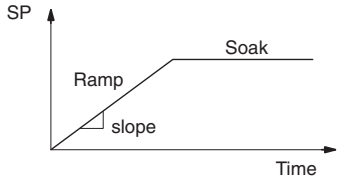
Introduction

Our discussion of basic loop operation noted the setpoint (SP) for a loop will be generated in various ways, depending on the loop operating mode and programming preferences. In the figure below, the ramp/soak generator is one of the ways the SP may be generated. It is the responsibility of your ladder program to ensure only one source attempts to write the SP value at V+02 at any particular time.



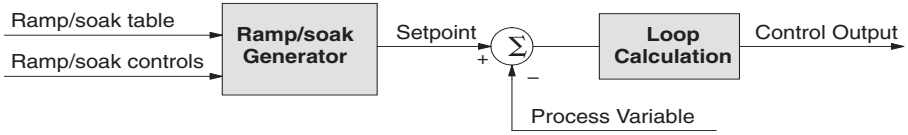
If the SP for your process rarely changes or can tolerate step changes, you probably will not need to use the ramp/soak generator. However, some processes require precisely-controlled SP value changes. *The ramp/soak generator can greatly reduce the amount of programming required for these applications.*

The terms "ramp" and "soak" have special meanings in the process control industry, and refer to desired setpoint values in temperature control applications. In the figure to the right, the setpoint increases during the ramp segment. It remains steady at one value during the soak segment.



Complex SP profiles can be generated by specifying a series of ramp/soak segments. The ramp segments are specified in SP units per second time. The soak time is also programmable in minutes.

It is instructive to view the ramp/soak generator as a dedicated function to generate SP values, as shown below. It has two categories of inputs which determine the SP values generated. The ramp/soak table must be programmed in advance, containing the values that will define the ramp/soak profile. The loop reads from the table during each PID calculation as necessary. The ramp/soak controls are bits in a special loop table word that control the real-time start/stop functionality of the ramp/soak generator. The ladder program can monitor the status of the ramp soak profile (current ramp/segment number).

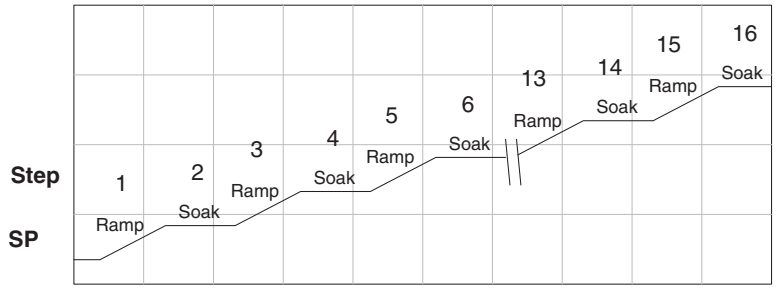


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Now that we have described the general ramp/soak generator operation, we list its specific features:

- Each loop has its own ramp/soak generator (use is optional).
- You may specify up to eight ramp/soak steps (16 segments).
- The ramp soak generator can run anytime the PLC is in Run mode. Its operation is independent of the loop mode (Manual or Auto).
- Ramp/soak real-time controls include Start, Hold, Resume, and Jog.
- Ramp/soak monitoring includes Profile Complete, Soak Deviation (SP minus PV), and current ramp/soak step number.

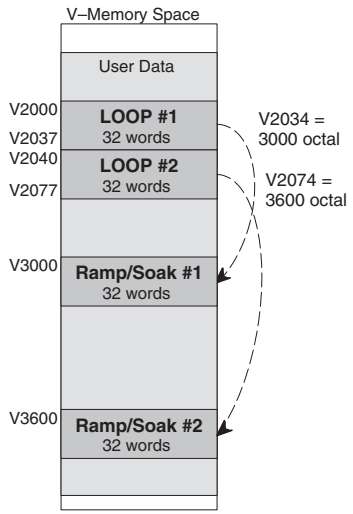
The following figure shows a SP profile consisting of ramp/soak segment pairs. The segments are individually numbered as steps from 1 to 16. The slope of each of the ramps may be either increasing or decreasing. The ramp/soak generator automatically knows whether to increase or decrease the SP based on the relative values of a ramp's end points. These values come from the ramp/soak table.



Ramp/Soak Table

The parameters which define the ramp/soak profile for a loop are in a ramp/soak table. Each loop may have its own ramp/soak table, but it is optional. Recall the Loop Parameter table consists of a 32-word block of memory for each loop, and together they occupy one contiguous memory area. However, the ramp/soak table for a loop is individually located, because it is optional for each loop. An address pointer in location V+34 in the loop table specifies the starting location of the ramp/soak table.

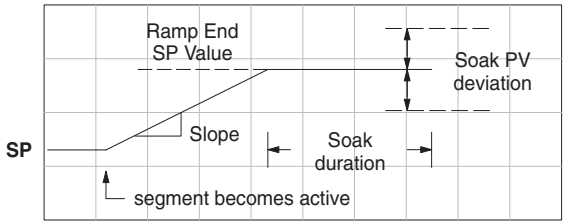
In the example to the right, the loop parameter tables for Loop #1 and #2 occupy contiguous 32-word blocks as shown. Each has a pointer to its ramp/soak table, independently located elsewhere in user V-memory. Of course, you may locate all the tables in one group, as long as they do not overlap.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The parameters in the ramp/soak table must be user-defined. the most convenient way is to use *Direct*SOFT, which features a special editor for this table. Four parameters are required to define a ramp and soak segment pair, as pictured below.

- **Ramp End Value** – specifies the destination SP value for the end of the ramp. Use the same data format for this number as you use for the SP. It may be above or below the beginning SP value, so the slope could be up or down (we don't have to know the starting SP value for ramp #1).
- **Ramp Slope** – specifies the SP increase in counts (units) per second. It is a BCD number from 00.00 to 99.99 (uses implied decimal point).
- **Soak Duration** – specifies the time for the soak segment in minutes, ranging from 000.1 to 999.9 minutes in BCD (implied decimal point).
- **Soak PV Deviation** – (optional) specifies an allowable PV deviation above and below the SP value during the soak period. A PV deviation alarm status bit is generated by the ramp/soak generator.



Ramp/Soak Table

V+00	XXXX	Ramp End SP Value
V+01	XXXX	Ramp Slope
V+02	XXXX	Soak Duration
V+03	XXXX	Soak PV Deviation

The ramp segment becomes active when the previous soak segment ends. If the ramp is the first segment, it becomes active when the ramp/soak generator is started, and automatically assumes the present SP as the starting SP.

Offset	Step	Description	Offset	Step	Description
+ 00	1	Ramp End SP Value	+ 20	9	Ramp End SP Value
+ 01	1	Ramp Slope	+ 21	9	Ramp Slope
+ 02	2	Soak Duration	+ 22	10	Soak Duration
+ 03	2	Soak PV Deviation	+ 23	10	Soak PV Deviation
+ 04	3	Ramp End SP Value	+ 24	11	Ramp End SP Value
+ 05	3	Ramp Slope	+ 25	11	Ramp Slope
+ 06	4	Soak Duration	+ 26	12	Soak Duration
+ 07	4	Soak PV Deviation	+ 27	12	Soak PV Deviation
+ 10	5	Ramp End SP Value	+ 30	13	Ramp End SP Value
+ 11	5	Ramp Slope	+ 31	13	Ramp Slope
+ 12	6	Soak Duration	+ 32	14	Soak Duration
+ 13	6	Soak PV Deviation	+ 33	14	Soak PV Deviation
+ 14	7	Ramp End SP Value	+ 34	15	Ramp End SP Value
+ 15	7	Ramp Slope	+ 35	15	Ramp Slope
+ 16	8	Soak Duration	+ 36	16	Soak Duration
+ 17	8	Soak PV Deviation	+ 37	16	Soak PV Deviation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Many applications do not require all 16 R/S steps. Use all zeros in the table for unused steps. The R/S generator ends the profile when it finds ramp slope = 0.

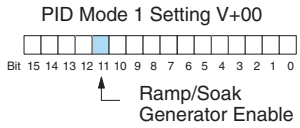
Ramp/Soak Table Flags

The individual bit definitions of the Ramp/Soak Table Flag (Addr+33) word is listed in the following table.

Bit	Ramp/Soak Flag Bit Description	Read/Write	Bit=0	Bit=1
0	Start Ramp/Soak Profile	write	–	01 Start
1	Hold Ramp/Soak Profile	write	–	01 Hold
2	Resume Ramp/Soak Profile	write	–	01 Resume
3	Jog Ramp/Soak Profile	write	–	01 Jog
4	Ramp/Soak Profile Complete	read	–	Complete
5	PV Input Ramp/Soak Deviation	read	Off	On
6	Ramp/Soak Profile in Hold	read	Off	On
7	Reserved	read	Off	On
8–15	Current Step in Ramp/Soak Profile	read	decode as byte (hex)	

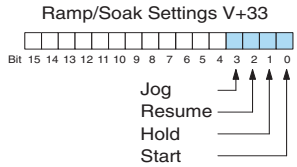
Ramp/Soak Generator Enable

The main enable control to permit ramp/soak generation of the SP value is accomplished with bit 11 in the PID Mode 1 Setting V+00 word, as shown to the right. The other ramp/soak controls in V+33 shown in the table above will not operate unless this bit=1 during the entire ramp/soak process.



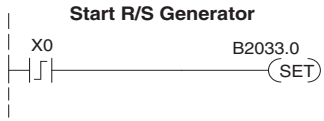
Ramp/Soak Controls

The four main controls for the ramp/soak generator are in bits 0 to 3 of the ramp/soak settings word in the loop parameter table. ***Direct***SOFT controls these bits directly from the ramp/soak settings dialog. However, you must use ladder logic to control these bits during program execution. We recommend using the bit-of-word instructions.



Ladder logic must set a control bit to a "1" to command the corresponding function. When the loop controller reads the ramp/soak value, it automatically turns off the bit for you. Therefore, a reset of the bit is not required, when the CPU is in Run Mode.

The example program rung to the right shows how an external switch X0 can turn on, and the PD contact uses the leading edge to set the proper control bit to start the ramp soak profile. This uses the Set Bit-of-word instruction.



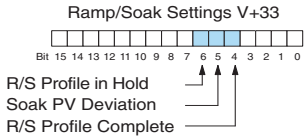
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The normal state for the ramp/soak control bits is all zeros. Ladder logic must set only one control bit at a time.

- **Start** – a 0 to 1 transition will start the ramp soak profile. The CPU must be in Run Mode, and the loop can be in Manual or Auto Mode. If the profile is not interrupted by a Hold or Jog command, it finishes normally.
- **Hold** – a 0 to 1 transition will stop the ramp/soak profile in its current state, and the SP value will be frozen.
- **Resume** – a 0 to 1 transition cause the ramp/soak generator to resume operation if it is in the hold state. The SP values will resume from their previous value.
- **Jog** – a 0 to 1 transition will cause the ramp/soak generator to truncate the current segment (step), and go to the next segment.

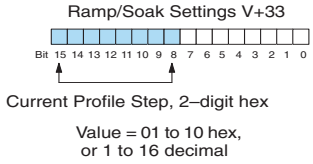
Ramp/Soak Profile Monitoring

You can monitor the Ramp/Soak profile status using other bits in the Ramp/Soak Settings V+33 word, shown to the right.



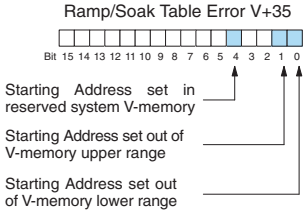
- R/S Profile Complete – equals 1 when the last programmed step is done.
- Soak PV Deviation – equals 1 when the error (SP–PV) exceeds the specified deviation in the R/S table.
- R/S Profile in Hold – equals 1 when the profile was active but is now in hold. Ramp/Soak Settings V+33

The number of the current step is available in the upper 8 bits of the Ramp/Soak Settings V+33 word. The bits represent a 2-digit hex number, ranging from 1 to 10. Ladder logic can monitor these to synchronize other parts of the program with the ramp/soak profile. Load this word to the accumulator and shift right 8 bits and you have the step number.



Ramp/Soak Programming Errors

The starting address for the ramp/soak table must be a valid location. If the address points outside the range of user V-memory, one of the bits to the right will turn on when the ramp/soak generator is started. We recommend using *Direct*SOFT to configure the ramp/soak table. It automatically range checks the addresses for you.



Testing Your Ramp/Soak Profile

It's a good idea to test your ramp/soak profile before using it to control the process. This is easy to do, because the ramp/soak generator will run even when the loop is in Manual Mode. Using *Direct*SOFT's PID View will be a real time-saver, because it will draw the profile on-screen for you. Be sure to set the trending timebase slow enough to display completed ramp-soak segment pairs in the waveform window.

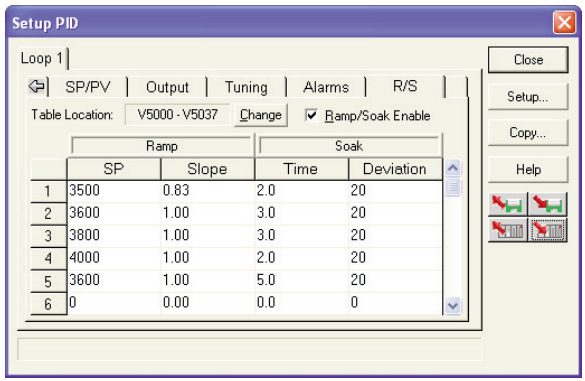
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

*Direct*SOFT Ramp/Soak Example

The following example will step you through the Ramp/Soak setup.

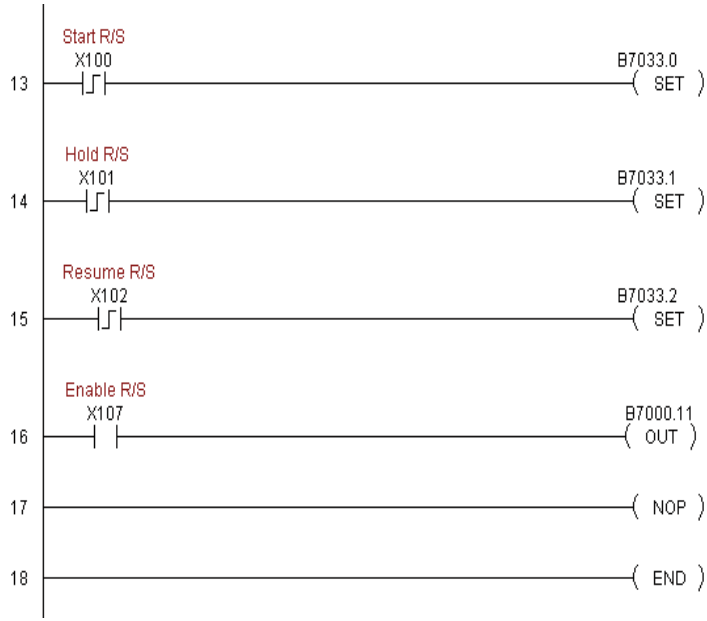
Setup the Profile in PID Setup

The first step is to use Setup PID in ***Direct*SOFT** to set the profile of your process. Open the Setup PID window and select the R/S tab, and then enter the Ramp and Soak data.



Program the Ramp/Soak Control in Relay Ladder

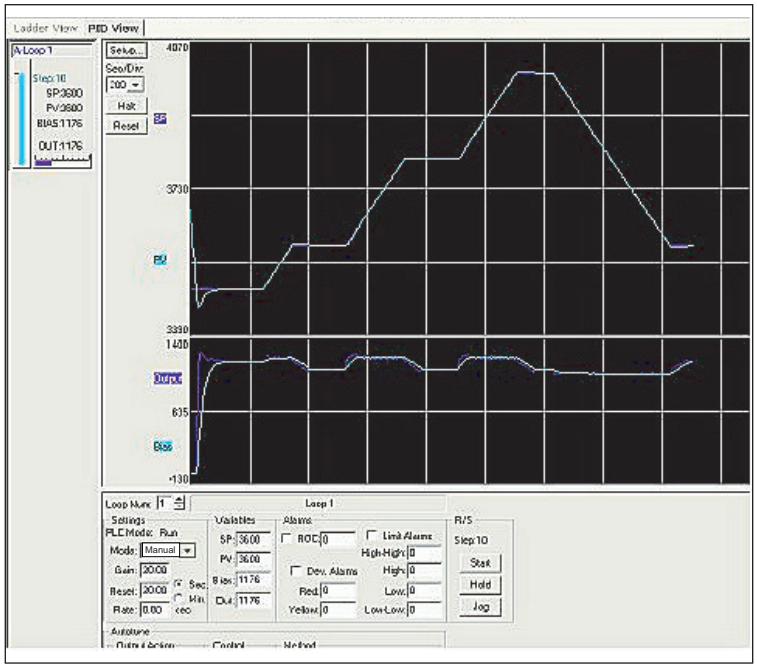
Refer to the Ramp/Soak Flag Bit Description table on page 8-60 when adding the control rungs to your program similar to the ladder rungs below.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Test the Profile

Test your profile using PID View.



8

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

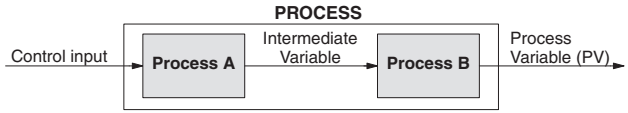
Cascade Control

Introduction

Cascaded loops are an advanced control technique that is superior to individual loop control in certain situations. As the name implies, cascade means that one loop is connected to another loop. In addition to Manual (open loop) and Auto (closed loop) Modes, the DL205 also provides Cascaded Mode.

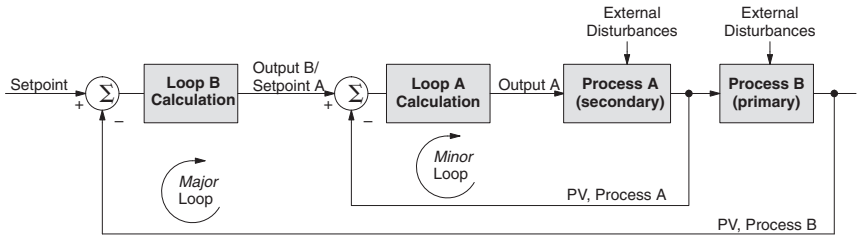
NOTE: Cascaded loops are an advanced process control technique. Therefore we recommend their use only for experienced process control engineers.

When a manufacturing process is complex and contains a lag time from control input to process variable output, even the most perfectly tuned single loop around the process may yield slow and inaccurate control. It may be the actuator operates on one physical property, which eventually affects the process variable, measured by a different physical property. Identifying the intermediate variable allows us to divide the process into two parts as shown in the following figure.



The principle of cascaded loops is simply that we add another process loop to more precisely control the intermediate variable! This separates the source of the control lag into two parts, as well.

The diagram below shows a cascade control system, showing that it is simply one loop nested inside another. The inside loop is called the minor loop, and the outside loop is called the major loop. For overall stability, the minor loop must be the fastest responding loop of the two (try a factor of 10 for a better response time). We do have to add the additional sensor to measure the intermediate variable (PV for process A). Notice the setpoint for the minor loop is automatically generated for us, by using the output of the major loop. Once the cascaded control is programmed and debugged, we only need to deal with the original setpoint and process variable at the system level. The cascaded loops behave as one loop, but with improved performance over the previous single-loop solution.



One of the benefits to cascade control can be seen by examining its response to external disturbances. Remember the minor loop is faster acting than the major loop. Therefore, if a disturbance affects process A in the minor loop, the Loop A PID calculation can correct the resulting error before the major loop sees the effect.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

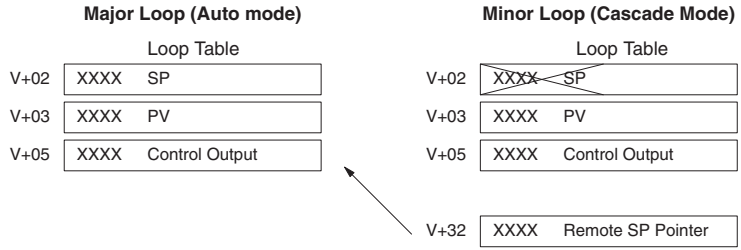
Cascaded Loops in the DL205 CPU

In the use of the term "cascaded loops", we must make an important distinction. Only the minor loop will actually be in the Cascade Mode. In normal operation, the major loop must be in Auto Mode. If you have more than two loops cascaded together, the outer-most (major) loop must be in Auto Mode during normal operation, and all inner loops in Cascade Mode.

NOTE: Technically, both major and minor loops are "cascaded" in strict process control terminology. Unfortunately, we are unable to retain this convention when controlling loop modes. Remember that all minor loops will be in Cascade Mode, and only the outer-most (major) loop will be in Auto Mode.

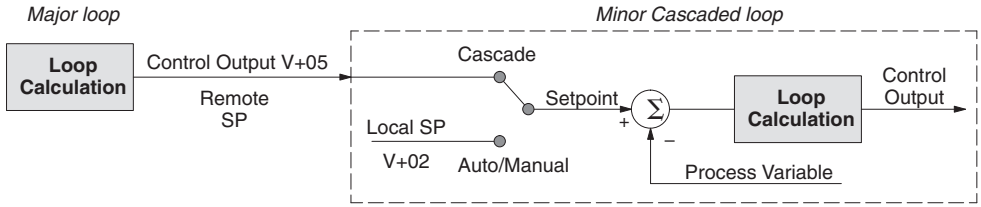
You can cascade together as many loops as necessary on the DL205, and you may have multiple groups of cascaded loops. For proper operation on cascaded loops you must use the same data range (12/15 bit) and unipolar/bipolar settings on the major and minor loop.

To prepare a loop for Cascade Mode operation as a minor loop, you must program its remote Setpoint Pointer in its loop parameter table location V+32, as shown below. The pointer must be the address of the V+05 location (control output) of the major loop. In Cascade Mode, the minor loop will ignore its local SP register (V+02), and read the major loop's control output as its SP instead.



When using *Direct*SOFT's PID View to monitor the SP value of the minor loop, *Direct*SOFT automatically reads the major loop's control output and displays it for the minor loop's SP. The minor loop's normal SP location, V+02, remains unchanged.

Now, we use the loop parameter arrangement above and draw its equivalent loop schematic, shown below.



Remember that a major loop goes to Manual Mode automatically if its minor loop is taken out of Cascade Mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tuning Cascaded Loops

In tuning cascaded loops, you will need to de-couple the cascade relationship and tune the loops individually, using one of the loop tuning procedures previously covered.

1. If you are not using auto tuning, then find the loop sample rate for the minor loop, using the method discussed earlier in this chapter. Then set the sample rate of the major loop slower than the minor loop by a factor of 10. Use this as a starting point.

2. Tune the minor loop first. Leave the major loop in Manual Mode and generate SP changes for the minor loop manually as described in the loop tuning procedure.

3. Verify the minor loop gives a critically-damped response to a 10% SP change while in Auto Mode. Then we are finished tuning the minor loop.

4. In this step, you will need to get the minor loop in Cascade Mode, and then the Major loop in Auto Mode. We will be tuning the major loop with the minor loop treated as a series component in its overall process. Therefore, do not go back and tune the minor loop again while tuning the major loop.

5. Tune the major loop by following the standard loop tuning procedure in this section. The response of the major loop PV is actually the overall response of the cascaded loops together.

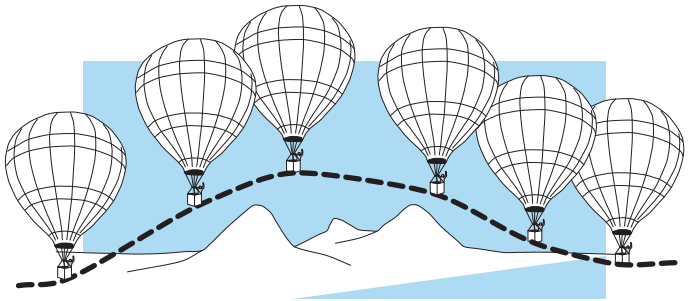
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Time-Proportioning Control

The PID loop controller in the DL205 CPU generates a smooth control output signal across a numerical range. The control output value is suitable to drive an analog output module, which connects to the process. In the process control field, this is called *continuous control*, because the output is on (at some level) continuously.

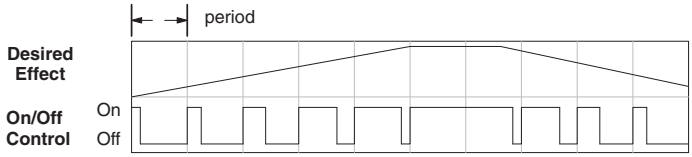
While continuous control can be smooth and robust, the cost of the loop components (such as actuator, heater amplifiers) can be expensive. A simpler form of control is called *time-proportioning control.* This method uses an actuator which is either on or off (no in-between). Loop components for on/off-based control systems are lower cost than their continuous control counterparts.

In this section, we will show you how to convert the control output of a loop to time-proportioning control for the applications that need it. Let's take a moment to review how



alternately turning a load on and off can control a process. The diagram below shows a hot-air balloon following a path across some mountains. The desired path is the *setpoint*. The balloon pilot turns the burner on and off alternately, which is his *control output*. The large mass of air in the balloon effectively averages the effect of the burner, converting the bursts of heat into a continuous effect: slowly changing balloon temperature and ultimately the altitude, which is the *process variable*.

Time-proportioning control approximates continuous control by virtue of its duty-cycle – the ratio of ON time to OFF time. The following figure shows an example of how duty-cycle approximates a continuous level when it is averaged by a large process mass.

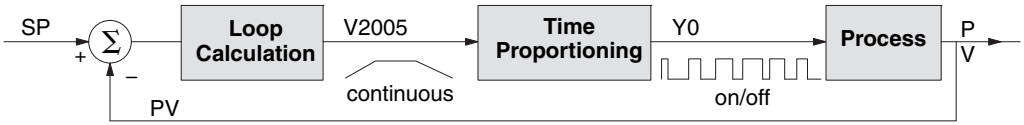


If we were to plot the on/off times of the burner in the hot-air balloon, we would probably see a very similar relationship to its effect on balloon temperature and altitude.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

On/Off Control Program Example

The following ladder segment provides a time proportioned on/off control output. It converts the continuous output in V2005 to on/off control using the output coil, Y0.

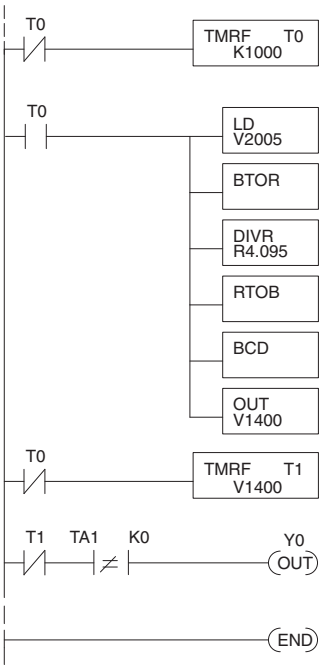


The example program uses two timers to generate On/Off control. It makes the following **assumptions**, which you can alter to fit your application:

- The loop table starts at V2000, so the control output is at V2005.
- The data format of the control output is 12-bit, unipolar (0 – FFF).
- The time base (one full cycle) for the On/Off waveform is 10 seconds. We use a fast timer (0.01 sec/tick), counting to 1000 ticks (10 seconds).
- The On/Off control output is Y0.

The time proportioning program must match the resolution of the output (1 part in 1000) to the resolution of the time base of T0 (also 1 part in 1000).

NOTE: Some processes change too fast for time proportioning control. Consider the speed of your process when you choose this control method. Use continuous control for processes that change too fast for time proportioning control. Also, consider using a solid state switch for a longer switch life instead of a relay.



A fast timer (0.01 sec. timebase) establishes the primary time interval. The constant, K1000, sets the preset at 10 seconds (1,000 ticks). The N.C. enabling contact, T0, makes the timer self-resetting. T0 is on for one scan each 10 seconds, when it resets itself and T1.

At the end of the 10 second period, T0 turns on, and loads the control output value (binary) from the loop table V+05 location (V2005).

The BTOR instruction changes the number in the accumulator to a real number.

Dividing the control output by 4.095, converts the 0 – 4095 range to 0 – 1000, which "matchs" the number of ticks in the 10 second timer range.

This instruction converts the real number back to binary. This step prepares the number for conversion to BCD. There is no real-to-BCD instruction.

Convert the number in the accumulator to BCD format. This satisfies the timer preset format requirement.

Output the result to V1400. In our example, this is the location of the timer preset for the second timer.

The second fast timer also counts in increments of .01 seconds, so its range is variable from 0 to a maximum of 1000 ticks, or 10 seconds. This timer's output, T1, turns off the output coil, Y0, when the preset is reached.

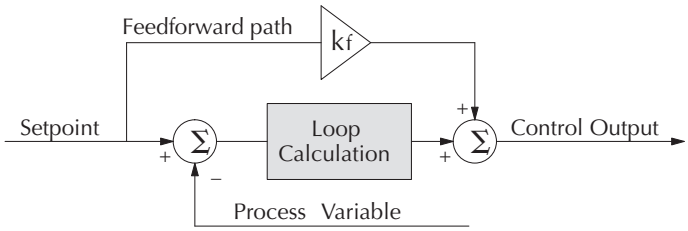
The N.C. T1 contact, inverts the T1 timer output. The control output is on at the beginning of the 10-second time interval. Y0 turns off when T1 times out. The STRNE contact prevents Y0 from energizing during the one scan when T0 resets T1. Y0 is the actual control output.

END coil marks the end of the main program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Feedforward Control

Feedforward control is an enhancement to standard closed-loop control. It is most useful for diminishing the effects of a *quantifiable and predictable* loop disturbance or sudden change in setpoint. Use of this feature is an option available to you on the DL205. However, it's best to implement and tune a loop without feedforward, and adding it only if better loop performance is still needed. The term "feed-forward" refers to the control technique involved, shown in the diagram below. The incoming setpoint value is fed forward around the PID equation, and summed with the output.

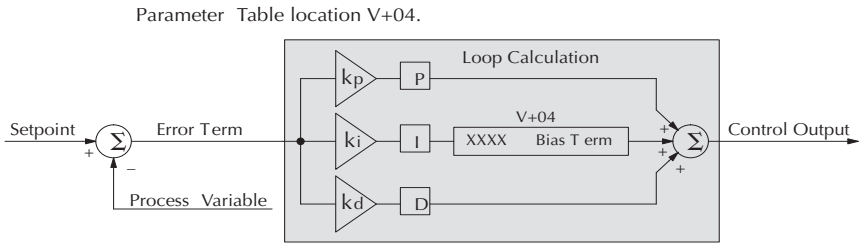


In the previous section on the bias term, we said that "the bias term value establishes a "working region" or operating point for the control output. *When the error fluctuates around its zero point, the output fluctuates around the bias value.*" Now, when there is a change in setpoint, an error is generated and the output must change to a new operating point. This also happens if a disturbance introduces a new offset in the loop. The loop does not really "know its way" to the new operating point... the integrator (bias) must increment/decrement until the error disappears, and then the bias has found the new operating point.

Suppose that we are able to know a sudden setpoint change is about to occur (common in some applications). We can avoid much of the resulting error in the first place, if we can quickly change the output to the new operating point. If we know (from previous testing) what the operating point (bias value) will be after the setpoint change, we can artificially change the output directly (which is feedforward). The benefits from using feedforward are:

- The SP–PV error is reduced during predictable setpoint changes or loop offset disturbances.
- Proper use of feedforward will allow us to reduce the integrator gain. Reducing integrator gain gives us an even more stable control system.

Feedforward is very easy to use in the DL205 loop controller, as shown below. The bias term has been made available to the user in a special read/write location, at PID Parameter Table location V+04.



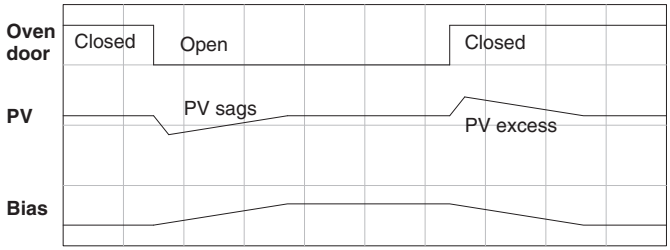
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To change the bias (operating point), ladder logic only has to write the desired value to V+04. The PID loop calculation first reads the bias value from V+04 and modifies the value based on the current integrator calculation. Then it writes the result back to location V+04. This arrangement creates a sort of "transparent" bias term. All you have to do to implement feed forward control is write the correct value to the bias term at the right time (see the following example).

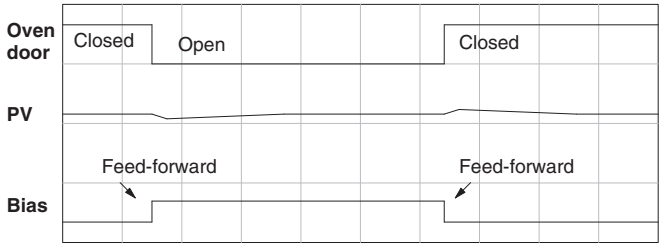
NOTE: When writing the bias term, one must be careful to design ladder logic to write the value only once, at the moment when the new bias operating point is to occur. If ladder logic writes the bias value on every scan, the loop's integrator is effectively disabled.

Feedforward Example

How do we know when to write to the bias term, and what value to write? Suppose we have an oven temperature control loop, and we have already tuned the loop for optimal performance. Refer to the figure below. We notice that when the operator opens the oven door, the temperature sags a bit while the loop bias adjusts to the heat loss. Then when the door closes, the temperature rises above the SP until the loop adjusts again. Feedforward control can help diminish this effect.



First, we record the amount of bias change the loop controller generates when the door opens or closes. Then, we write a ladder program to monitor the position of an oven door limit switch. When the door opens, our ladder program reads the current bias value from V+04, adds the desired change amount, and writes it back to V+04. When the door closes, we duplicate the procedure, but subtracting desired change amount instead. The following figure shows the results.



The step changes in the bias are the result of our two feed-forward writes to the bias term. We can see the PV variations are greatly reduced. The same technique may be applied for changes in setpoint.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Example Program

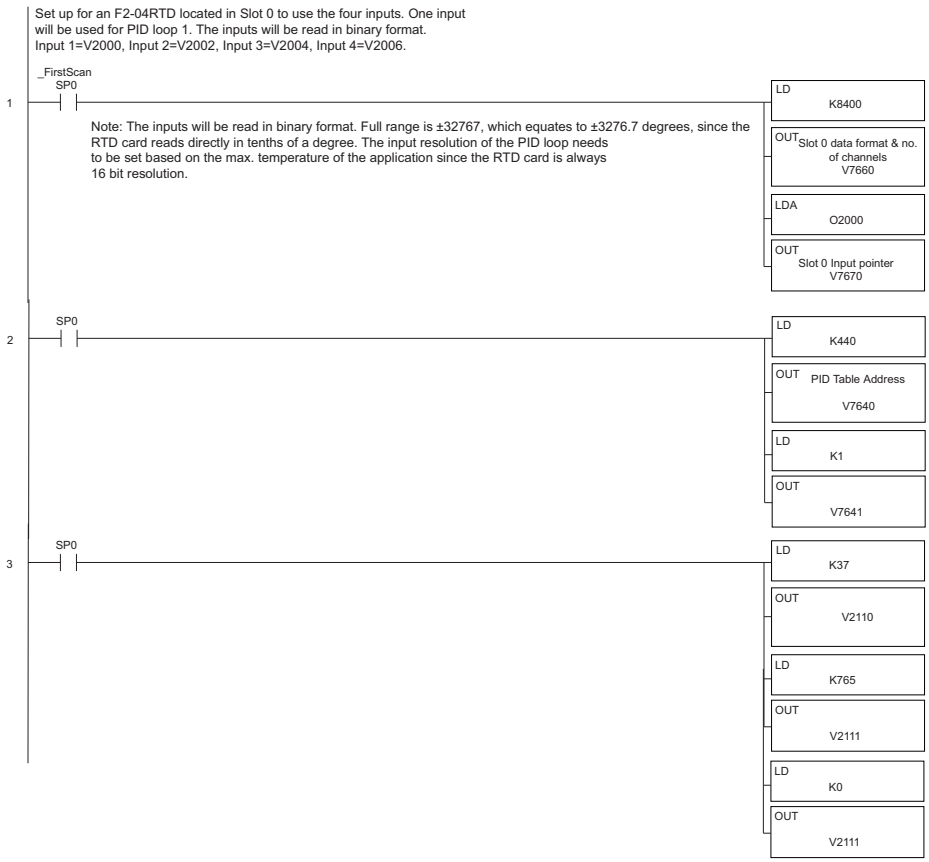
Program Setup for the PID Loop

After the PID loop, or loops, have been setup with *Direct*SOFT, you will need to edit your RLL program to include the rungs needed to setup the analog I/O module to be used by the PID loop(s).

The following example program shows how an RTD module, F2-04RTD, and an analog output module, F2-02DA-2, are used and setup for a PID loop. This example assumes that the PID table for loop 1 has a beginning address of V2100.

All of the analog I/O modules used with the DL205 are setup in a similar manner. Refer to the DL205 Analog Manual for the setup information for the particular module that you will be using.

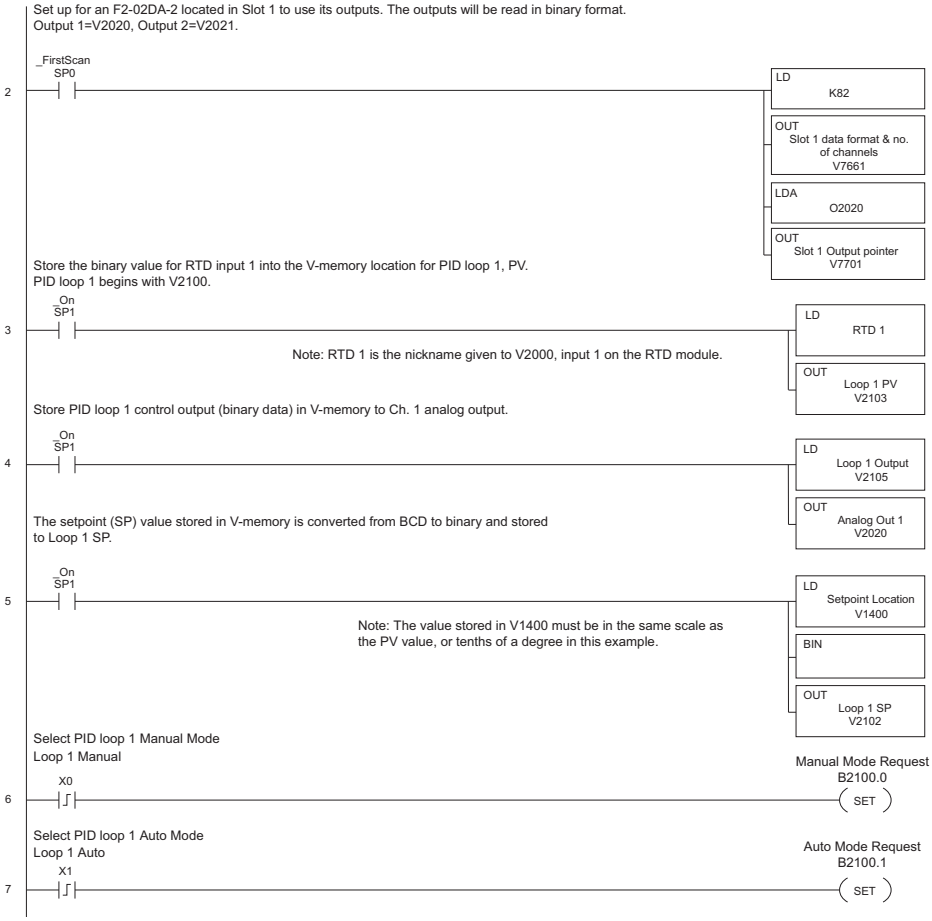
DirectSOFT



Program continued on next page

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example program continued



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note that the modules used in the PID loop example program were set up for binary format. They could have been set up for BCD format. In the later case, the BCD data would have to be converted to binary format before being stored to the setpoint and process variable, and the control output would have to be converted from binary to BCD before being stored to the analog output.

By following the steps outlined in this chapter, you should be able to setup workable PID control loops. The ***Direct***SOFT Programming Software Manual provides more information for the use of PID View.

For a step-by-step tutorial, go to the Technical Support section located on our website, **www.automationdirect.com**. Once you are at the website, click on **Technical Support Home**. After this page opens, find and select **Guided Tutorials** located under the **Using Your Products** column. An **Animated Tutorial** page will open. Under **Available Tutorials**, find **PID Trainer** and select **View the Powerpoint slide show** and begin viewing the tutorial. The Powerpoint Viewer can be downloaded if your computer does not have Powerpoint installed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Troubleshooting Tips

Q. The loop will not go into Automatic Mode.

A. Check the following for possible causes:

- A PV alarm exists, or a PV alarm programming error exists.
- The loop is the major loop of a cascaded pair, and the minor loop is not in Cascade Mode.

Q. The Control Output stays at zero constantly when the loop is in Automatic Mode.

A. Check the following for possible causes:

- The Control Output upper limit in loop table location V+31 is zero.
- The loop is driven into saturation, because the error never goes to zero value and changes (algebraic) sign.

Q. The Control Output value is not zero, but it is incorrect.

A. Check the following for possible causes:

- The gain values are entered improperly. Remember, gains are entered in the loop table in BCD, while the SP and PV are in binary. If you are using *Direct*SOFT, it displays the SP, PV, Bias and Control output in decimal (BCD), converting it to binary before updating the loop table.

Q. The Ramp/Soak Generator does not operate when I activate the Start bit.

A. Check the following for possible causes:

- The Ramp/Soak enable bit is off. Check the status of bit 11 of loop parameter table location V+00. It must be set =1.
- The hold bit or other bits in the Ramp/Soak control are on.
- The beginning SP value and the first ramp ending SP value are the same, so first ramp segment has no slope and consequently has no duration. The ramp/soak generator moves quickly to the soak segment, giving the illusion the first ramp is not working.
- The loop is in Cascade Mode, and is trying to get the SP remotely.
- The SP upper limit value in the loop table location V+27 is too low.
- Check your ladder program to verify it is not writing to the SP location (V+02 in the loop table). A quick way to do this is to temporarily place an end coil at the beginning of your program, then go to PLC Run Mode, and manually start the ramp/soak generator.

Q. The PV value in the table is constant, even though the analog module receives the PV signal.

A. Your ladder program must read the analog value from the module successfully and write it into the loop table V+03 location. Verify the analog module is generating the value, and the ladder is working.

Q. The Derivative gain doesn't seem to have any affect on the output.

A. The derivative limit is probably enabled (see section on derivative gain limiting).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Q. The loop Setpoint appears to be changing by itself.

A. Check the following for possible causes:

- The Ramp/Soak generator is enabled, and is generating setpoints.
- If this symptom occurs on loop Manual-to-Auto Mode changes, the loop is in Bumpless Transfer 1.
- Check your ladder program to verify it is not writing to the SP location (V+02 in the loop table). A quick way to do this is to temporarily place an end coil at the beginning of your program, then go to PLC Run Mode.

Q. The SP and PV values I enter with *Direct*SOFT work okay, but these values do not work properly when the ladder program writes the data.

A. The PID View in *Direct*SOFT lets you enter SP, PV, and Bias values in decimal, and displays them in decimal for your convenience. For example, when the data format is 12-bit unipolar, the values range from 0 to 4095. However, the loop table actually requires these in hex, so *Direct*SOFT converts them for you. The values in the table range from 0 to FFF, for 12-bit unipolar format.

Q. The loop seems unstable and impossible to tune, no matter what gains I use.

A. Check the following for possible causes:

- The loop sample time is set too long. Refer to the section near the front of this chapter on selecting the loop update time.
- The gains are too high. Start out by reducing the derivative gain to zero. Then reduce the integral gain, and the proportional gain if necessary.
- There is too much transfer lag in your process. This means the PV reacts sluggishly to control output changes. There may be too much "distance" between actuator and PV sensor, or the actuator may be weak in its ability to transfer energy into the process.
- There may be a process disturbance that is over-powering the loop. Make sure the PV is relatively steady when the SP is not changing.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Glossary of PID Loop Terminology

Automatic Mode An operational mode of a loop, in which it makes PID calculations and updates the loop's control output.

Bias Freeze A method of preserving the bias value (operating point) for a control output, by inhibiting the integrator when the output goes out-of-range. The benefit is a faster loop recovery.

Bias Term In the position form of the PID equation, it is the sum of the integrator and the initial control output value.

Bumpless Transfer A method of changing the operation mode of a loop while avoiding the usual sudden change in control output level. This consequence is avoided by artificially making the SP and PV equal, or the bias term and control output equal at the moment of mode change.

Cascaded Loops A cascaded loop receives its setpoint from the output of another loop. Cascaded loops have a major/minor relationship, and work together to ultimately control one PV.

Cascade Mode An operational mode of a loop, in which it receives its SP from another loop's output.

Continuous Control Control of a process done by delivering a smooth (analog) signal as the control output.

Control Output The numerical result of a PID equation which is sent by the loop with the intention of nulling out the current error.

Derivative Gain A constant that determines the magnitude of the PID derivative term in response to the current error.

Direct-Acting Loop A loop in which the PV increases in response to a control output increase. In other words, the process has a positive gain.

Error The difference in value between the SP and PV, Error = SP – PV

Error Deadband An optional feature which makes the loop insensitive to errors when they are small. You can specify the size of the deadband.

Error Squared An optional feature which multiplies the error by itself, but retains the original algebraic sign. It reduces the effect of small errors, while magnifying the effect of large errors.

Feedforward A method of optimizing the control response of a loop when a change in setpoint or disturbance offset is known and has a quantifiable effect on the bias term.

Integral Gain A constant that determines the magnitude of the PID integral term in response to the current error.

Major Loop In cascade control, it is the loop that generates a setpoint for the cascaded loop.

Manual Mode An operational mode of a loop, it which the PID calculations are stopped. The operator must manually control the loop by writing to the control output value directly.

Minor Loop In cascade control, the minor loop is the subordinate loop that receives its SP from the major loop.

On/Off Control A simple method of controlling a process, through on/off application of energy into the system. The mass of the process averages the on/off effect for a relatively smooth PV. A simple ladder program can convert the DL205's continuous loop output to on/off control.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PID Loop A mathematical method of closed-loop control involving the sum of three terms based on proportional, integral, and derivative error values. The three terms have independent gain constants, allowing one to optimize (tune) the loop for a particular physical system.

Position Algorithm The control output is calculated so it responds to the displacement (position) of the PV from the SP (error term)

Process A manufacturing procedure which adds value to raw materials. Process control particularly refers to inducing chemical changes to the material in process.

Process Variable (PV) A quantitative measurement of a physical property of the material in process, which affects final product quality and is important to monitor and control.

Proportional Gain A constant that determines the magnitude of the PID proportional term in response to the current error.

PV Absolute Alarm A programmable alarm that compares the PV value to alarm threshold values.

PV Deviation Alarm A programmable alarm that compares the difference between the SP and PV values to a deviation threshold value.

Ramp/Soak Profile A set of SP values called a profile, which is generated in real time upon each loop calculation. The profile consists of a series of ramp and soak segment pairs, greatly simplifying the task of programming the PLC to generate such SP sequences.

Rate Also called differentiator, the rate term responds to the changes in the error term.

Remote Setpoint The location where a loop reads its setpoint when it is configured as the minor loop in a cascaded loop topology.

Reset Also called integrator, the reset term adds each sampled error to the previous, maintaining a running total called the bias.

Reset Windup A condition created when the loop is unable to find equilibrium, and the persistent error causes the integrator (reset) sum to grow excessively (windup). Reset windup causes an extra recovery delay when the original loop fault is remedied.

Reverse-Acting Loop A loop in which the PV increases in response to a control output decrease. In other words, the process has a negative gain.

Sampling time The time between PID calculations. The CPU method of process control is called a sampling controller, because it samples the SP and PV only periodically.

Setpoint (SP) The desired value for the process variable. The setpoint (SP) is the input command to the loop controller during closed loop operation.

Soak Deviation The soak deviation is a measure of the difference between the SP and PV during a soak segment of the Ramp/Soak profile, when the Ramp/Soak generator is active.

Step Response The behavior of the process variable in response to a step change in the SP (in closed loop operation), or a step change in the control output (in open loop operation)

Transfer To change from one loop operational mode to another (between Manual, Auto, or Cascade). The word "transfer" probably refers to the transfer of control of the control output or the SP, depending on the particular mode change.

Velocity Algorithm The control output is calculated to represent the rate of change (velocity) for the PV to become equal to the SP.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bibliography

Fundamentals of Process Control Theory, Second Edition Author: Paul W. Murrill Publisher: Instrument Society of America ISBN 1–55617–297–4	Application Concepts of Process Control Author: Paul W. Murrill Publisher: Instrument Society of America ISBN 1–55617–080–7
PID Controllers: Theory, Design, and Tuning, 2nd Edition Author: K. Astrom and T Hagglund Publisher: Instrument Society of America ISBN 1–55617–516–7	Fundamentals of Temperature, Pressure, and Flow Measurements, Third edition Author: Robert P. Benedict Publisher: John Wiley and Sons ISBN 0–471–89383–8
Process / Industrial Instruments & Controls Handbook, Fourth Edition Author (Editor-in-Chief): Douglas M. Considine Publisher: McGraw-Hill, Inc ISBN 0-07-012445-0	pH Measurement and Control, Second Edition Author: Gregory K. McMillan Publisher: Instrument Society of America ISBN 1–55617–483–7
Programmable Controllers Concepts and Applications, First Edition Authors: C.T. Jones and L.A. Bryan Publisher: International Programmable Controls ISBN 0-915425-00-9	Fundamentals of Programmable Logic Controllers, Sensors, and Communications Author: Jon Stenerson Publisher: Prentice Hall ISBN 0-13-726860-2
Process Control, Third Edition Instrument Engineer's Handbook Author (Editor-in-Chief): Bela G. Liptak Publisher: Chilton. . . . . ISBN 0–8019–8242–1	Process Measurement and Analysis, Third Edition Instrument Engineer's Handbook Author (Editor-in-Chief): Bela G. Liptak Publisher: Chilton. . . . . ISBN 0–8019–8197–2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hardware Maintenance

Standard Maintenance

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

- Air quality (cabinet temperature, airflow, etc.)
- 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

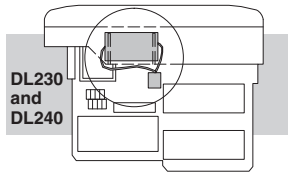
The CPU battery is used to retain program V-memory and the system parameters. The life expectancy of this battery is five years.



***NOTE**: Before installing or replacing your CPU battery, back-up your V-memory and system parameters. You can do this by using **Direct**SOFT 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:

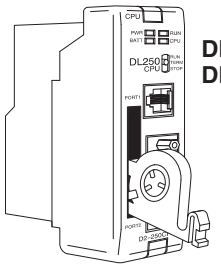
1. Gently push the battery connector onto the circuit board connector.
2. Push the battery into the retaining clip. Don't use excessive force. You may break the retaining clip.
3. Make a note of the date the battery was installed.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 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.



DL250–1
DL260

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. There are special relays the application program can 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:

- Backup battery voltage low
- All I/O module errors.
- Certain programming errors

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 or *Direct*SOFT.
- The handheld programmer displays error numbers and short descriptions of the error.
- *Direct*SOFT 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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

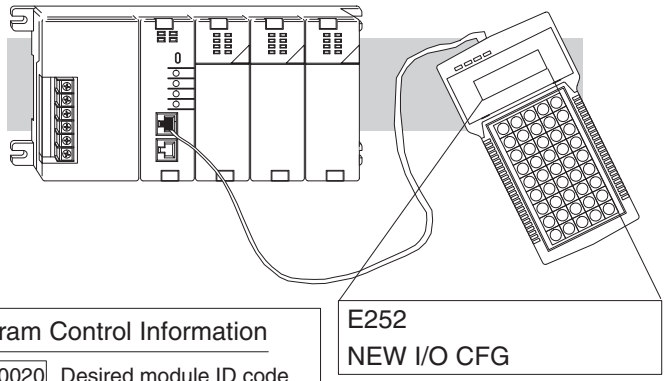
I/O Module Codes

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
04	CPU
03	I/O Base
20	8 pt. Output
21	8 pt. Input
24	4 input/output combination
28	12 pt. Output 16 pt. Output
3F	32 pt. Input
30	32 pt. Output
52	H2-ERM
51	H2-CTRIO

Code (Hex)	Component Type
36	Analog Input
2B	16 pt. Input
37	Analog Output
3D	Analog I/O Combo
4A	Counter Interface
7F	Abnormal
FF	No module detected
EE	D2-DCM H2-ECOM F2-CP128
BE	D2-RMSM

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



Program Control Information

V7752	0020	Desired module ID code
V7753	0026	Current module ID code
V7754	0002	Location of conflict
V7755	0252	Fatal error code
SP47	⊣⊢	I/O Configuration Error

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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. 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 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.

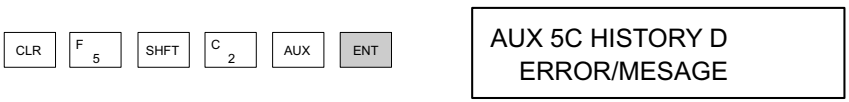
The following diagram shows an example of an error table for messages.

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 *Direct*SOFT's PLC Diagnostic sub-menus or from the Handheld Programmer. Details on how to access these logs are provided in the *Direct*SOFT manual.

The following examples show you how to use the Handheld 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.

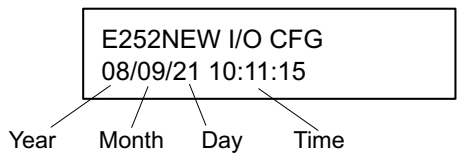
Use AUX 5C to view the tables



Use the arrow key to select Errors or Messages



Example of an error display



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

System Error Codes

☒ 230
☑ 240
☑ 250-1
☑ 260

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 a more complete description of the error codes.

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 a more complete description 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

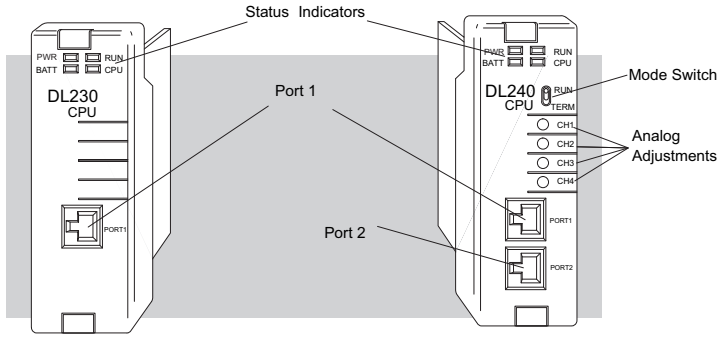
Error Code	Description
E461	Stack Overflow
E462	Stack Underflow
E463	Logic Error
E464	Missing Circuit
E471	Duplicate coil reference
E472	Duplicate TMR reference
E473	Duplicate CNT reference
E480	CV position error
E481	CV not connected
E482	CV exceeded
E483	CVJMP placement error
E484	No CV
E485	No CVJMP
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

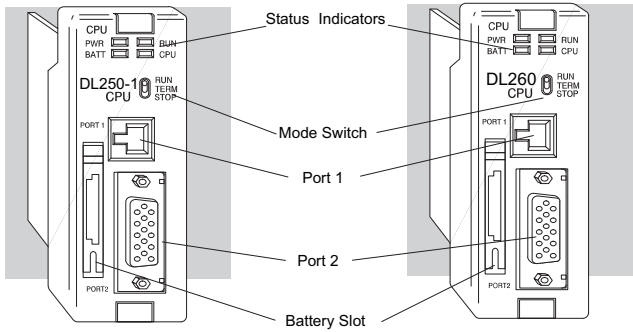
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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)	1. System voltage incorrect
	2. Power supply/CPU is faulty
	3. Other component such an I/O module has power supply shorted
	4. Power budget exceeded for the base being used
RUN (will not come on)	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)	1. Electrical noise interference
	2. CPU defective
BATT (on)	1. CPU battery low
	2. CPU battery missing, or disconnected





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PWR Indicator

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

1. Power to the base is incorrect or is not applied.
2. Base power supply is faulty.
3. Other component(s) have the power supply shut down.
4. 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 insure 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 which removes damaging voltage spikes should be used in the future.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

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 start-up 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 ***Direct***SOFT, 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

I/O Module Troubleshooting

Things to Check

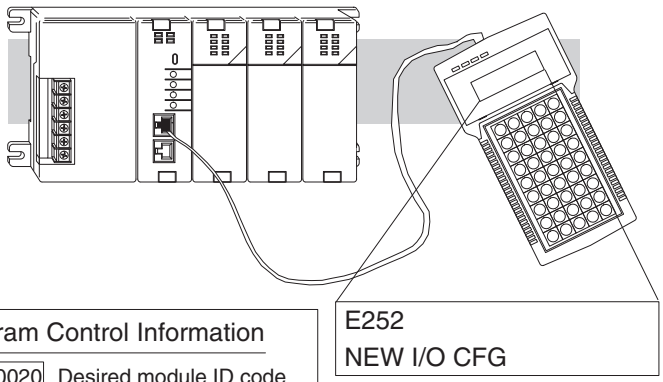
If you suspect an I/O error, there are several things that could be causing the problem.

- A blown fuse
- A loose terminal block
- The 24 VDC 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 *Direct*SOFT. 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 following figure 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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Some Quick Steps

When troubleshooting the DL205 series I/O modules there are a few facts you should be aware of which 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 in 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 10KΩ to 20KΩ 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

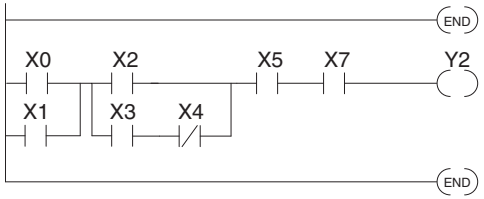
Testing Output Points

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 ***Direct***SOFT 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

STAT ENT

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

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

NEXT A 0 ENT

```
  Y 10   Y   0
□□□□□□□□□□□□□□□□
```

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

← ← SHFT ON INS

Y2 is now on

```
  Y 10   Y   0
□□□□□□□□□□□□□■□□
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 of the types of noise problems 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Machine Startup and Program Troubleshooting

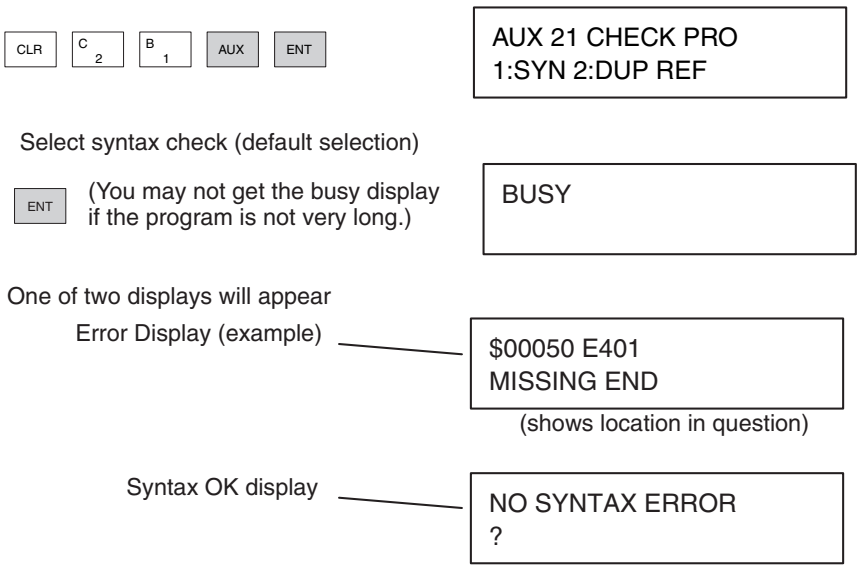
The DL205 CPUs provide several features to help you debug your program before and during machine startup. This section discusses the following topics which can be very helpful.

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

Syntax Check

Even though the Handheld Programmer and *Direct*SOFT 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 *Direct*SOFT. 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



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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 *Direct*SOFT. The following example shows how to perform the duplicate reference check with a Handheld Programmer.

Use AUX 21 to perform syntax check

CLR C 2 B 1 AUX ENT

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

Select duplicate reference check

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

```
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 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.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

With the Handheld, 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. ***Direct***SOFT provides more flexibility in selecting the various modes with different menu options. The following example shows how you can use the Handheld to select the Test Modes.

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

MODE NEXT ENT

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

Press ENT to confirm TEST-RUN Mode

ENT (Note, the TEST LED on the DL205 Handheld indicates the CPU is in TEST 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

CLR MODE NEXT NEXT ENT

```
*MODE CHANGE*
GO TO T–PGM MODE
```

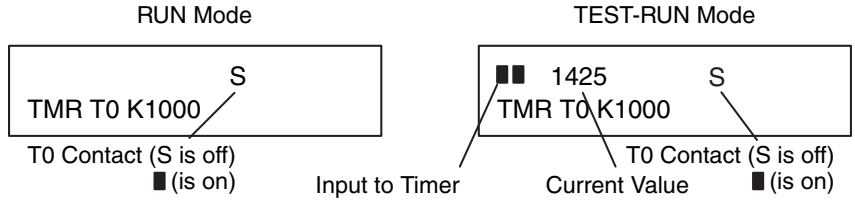
Press ENT to confirm TEST-PGM Mode

ENT (Note, the TEST LED on the DL205 Handheld indicates the CPU is in TEST Mode.)

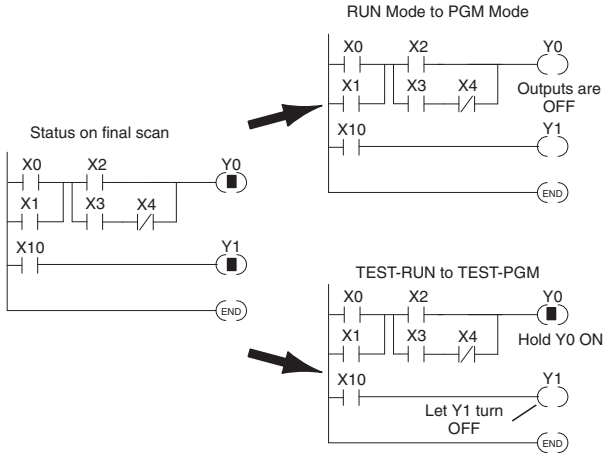
```
*MODE CHANGE*
CPU T–PGM
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 ***Direct***SOFT. 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.

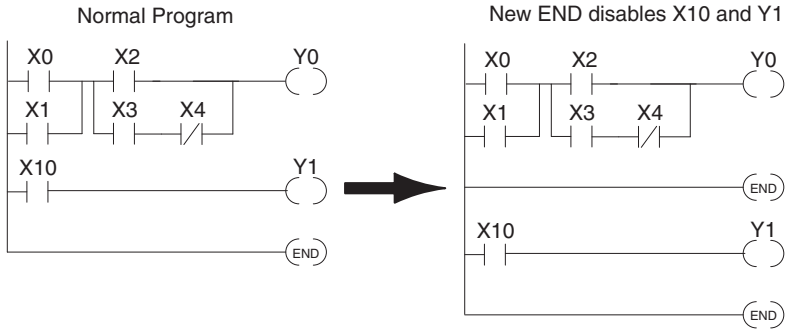
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Special Instructions

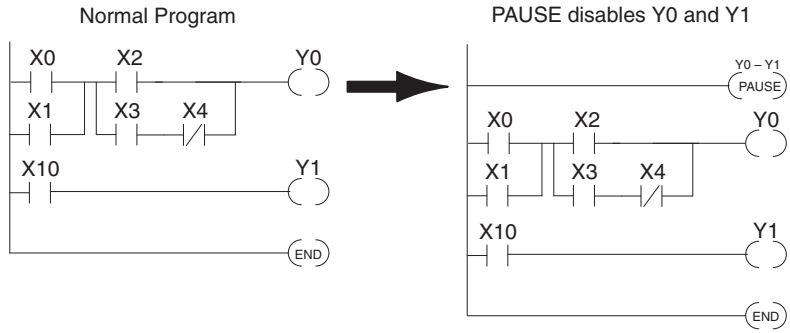
There are several instructions that can be used to help you debug your program during machine startup operations.

- 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.

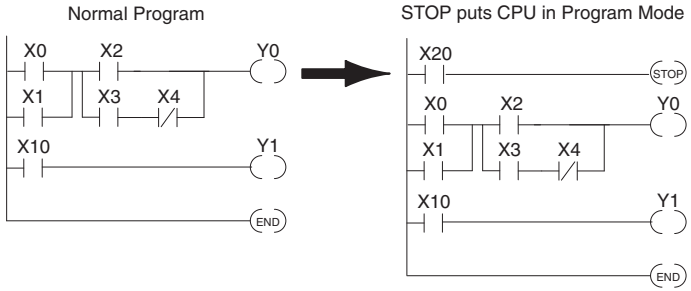


PAUSE Instruction: This instruction provides a quick way to allow the inputs (or other logic) to operate while disabling selected outputs. The output image register is still updated, but the output status is not written to the modules. For example, you could make this conditional by adding an input contact 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.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

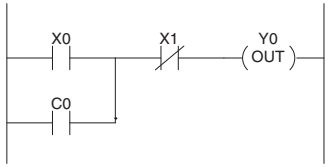
Not all instructions can be edited during a Run Time Edit session. The following list shows the instructions that can be edited.

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 NEXT NEXT ENT

```
*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)

SHFT X SET A 0 SHFT FD REF FIND

```
$00000 STR X0
```

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

→ → SHFT C 2 B 1 A 0 ENT

```
RUNTIME EDIT?
STR C10
```

Press ENT to confirm the change

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

```
OR C0
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

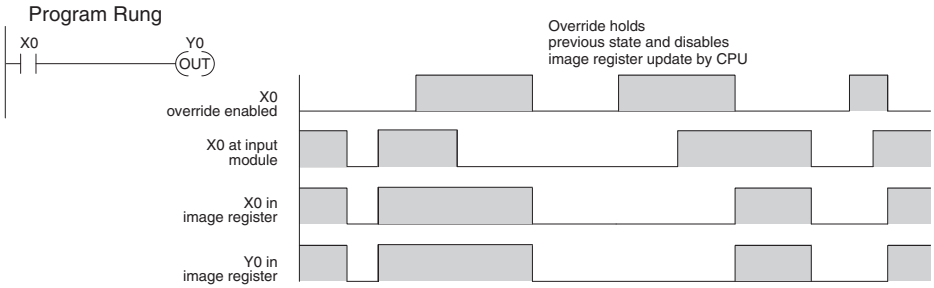
There are two types of forcing available with the DL205 CPUs. (Chapter 3 provides a detailed description of how the CPU processes each type of forcing request.)

- **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 *Direct*SOFT. 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".

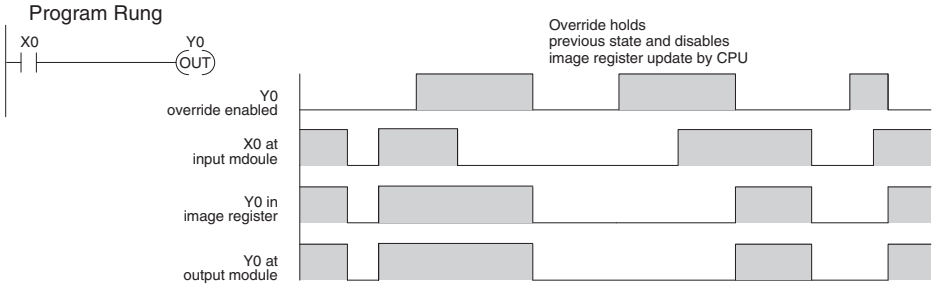
There is an advantage available when you use the bit override feature. The regular forcing is not disabled because the bit override is enabled. For example, if you enabled the Bit Override for Y0 and it was off at the time, the CPU would not change the state of Y0. However, you can still use a programming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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

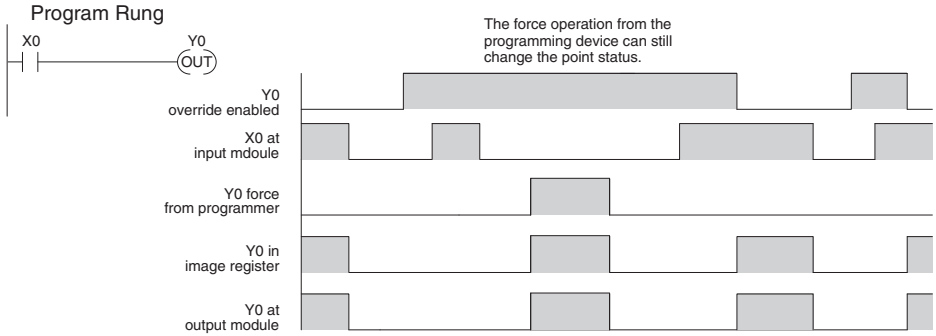
The following diagrams show how the bit override works for both input and output points. The example uses a simple rung, but the concepts are similar for any type of bit memory.



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.

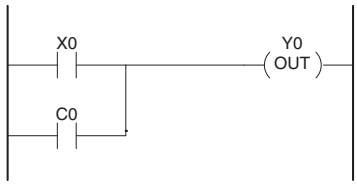


The following diagram shows how you can use a programming device in combination with the bit override to change the status of the point. Remember, bit override only disables CPU changes. You can still use a programming device to force the status of the point. Plus, since bit override maintains the current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



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.



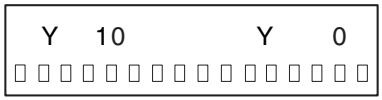
From a clear display, use the following keystrokes

STAT ENT



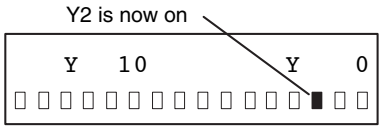
Use the PREV or NEXT keys to select the Y data type. (Once the Y appears, press 0 to start at Y0.)

NEXT A 0 ENT



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

← ← SHFT ON INS



Regular Forcing with Direct Access

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

SHFT Y MLS B 1 A 0 SHFT ON INS

Solid fill indicates point is on.



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

SHFT Y MLS B 1 A 0 SHFT OFF DEL

No fill indicates point is off.



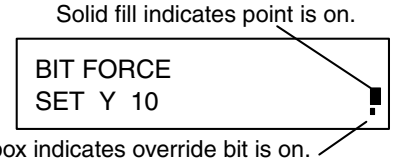
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit Override Forcing

☒ 230
☑ 240
☑ 250-1
☑ 260

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



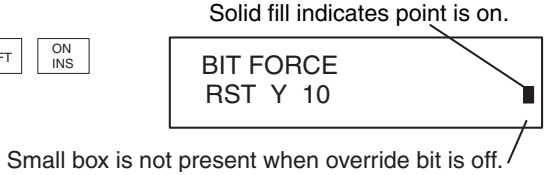




***NOTE**: At this point you can use the PREV and NEXT keys to move to adjacent memory locations and use the SHFT ON keys to set the override bit on.*

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





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.

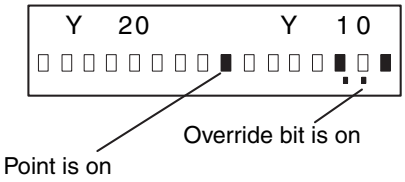
9

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.





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Auxiliary Functions



In This Appendix...

Introduction A–2
AUX 2* — RLL Operations A–4
AUX 3* — V-memory Operations A–5
AUX 4* — I/O Configuration A–5
AUX 5* — CPU Configuration A–7
AUX 6* — Handheld Programmer Configuration A–12
AUX 7* — EEPROM Operations A–12
AUX 8* — Password Operations A–14

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

A

What are Auxiliary Functions?

Many CPU setup tasks involve the use of Auxiliary (AUX) Functions. The AUX Functions perform many different operations, ranging from clearing ladder memory, displaying the scan time, copying programs to EEPROM in the handheld programmer, etc. They are divided into categories that affect different system parameters. You can access the AUX Functions from *Direct*SOFT or from the DL205 Handheld Programmer. The manuals for those products provide step-by-step procedures for accessing the AUX Functions. Some of these AUX Functions are designed specifically for the Handheld Programmer setup, so they will not be needed (or available) with the *Direct*SOFT package. Even though this Appendix provides many examples of how the AUX functions operate, you should supplement this information with the documentation for your choice of programming device. Note, the Handheld Programmer may have additional AUX functions that are not supported with the DL205 CPUs.

AUX Function and Description		230	240	250-1	260
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					
41	Show I/O Configuration	✓	✓	✓	✓
42	I/O Diagnostics	✓	✓	✓	✓
44	Power-up I/O Configuration Check	✓	✓	✓	✓
45	Select Configuration	✓	✓	✓	✓
46	Configure I/O	✗	✗	✓	✓
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	✗	✓	✓	✓
57	Set Retentive Ranges	✓	✓	✓	✓
58	Test Operations	✓	✓	✓	✓
59	Bit Override	✗	✓	✓	✓
5B	Counter Interface Config.	✓	✓	✓	✓
5C	Display Error History	✗	✓	✓	✓

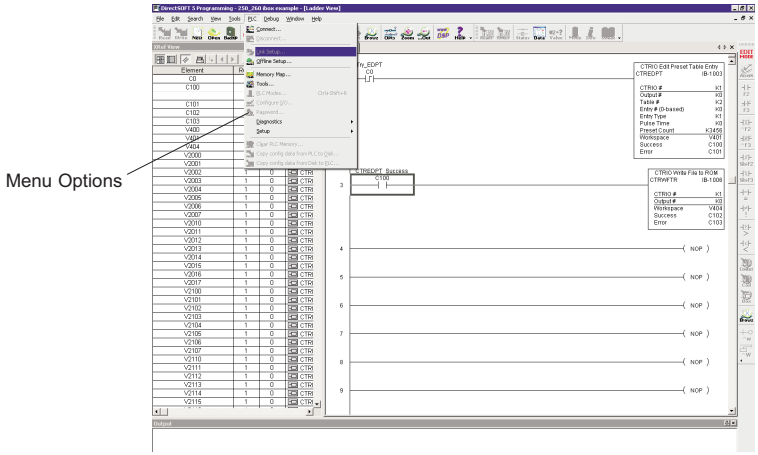
AUX Function and Description		230	240	250-1	260	HPP
AUX 6* — Handheld Programmer Configuration						
61	Show Revision Numbers	✓	✓	✓	✓	-
62	Beeper On / Off	✗	✗	✗	✗	✓
65	Run Self Diagnostics	✗	✗	✗	✗	✓
AUX 7* — EEPROM Operations						
71	Copy CPU memory to HPP EEPROM	✗	✗	✗	✗	✓
72	Write HPP EEPROM to CPU	✗	✗	✗	✗	✓
73	Compare CPU to HPP EEPROM	✗	✗	✗	✗	✓
74	Blank Check (HPP EEPROM)	✗	✗	✗	✗	✓
75	Erase HPP EEPROM	✗	✗	✗	✗	✓
76	Show EEPROM Type (CPU and HPP)	✗	✗	✗	✗	✓
AUX 8* — Password Operations						
81	Modify Password	✓	✓	✓	✓	-
82	Unlock CPU	✓	✓	✓	✓	-
83	Lock CPU	✓	✓	✓	✓	-

✓ supported
✗ not supported
- not applicable

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Accessing AUX Functions via *Direct*SOFT

*Direct*SOFT provides various menu options during both online and offline programming. Some of the AUX functions are only available during online programming, some only during offline programming, and some during both online and offline programming. The following diagram shows an example of the PLC operations menu available within *Direct*SOFT.



Accessing AUX Functions via the Handheld Programmer

You can also access the AUX functions by using a Handheld Programmer. Plus, remember some of the AUX functions are only available from the Handheld. Sometimes the AUX name or description cannot fit on one display. If you want to see the complete description, press the arrow keys to scroll left and right. Also, depending on the current display, you may have to press CLR more than once.

CLR AUX

```
AUX FUNCTION SELECTION
AUX 2* RLL OPERATIONS
```

Use NEXT or PREV to cycle through the menus

NEXT

```
AUX FUNCTION SELECTION
AUX 3* V OPERATIONS
```

Press ENT to select sub-menus

ENT

```
AUX 3* V OPERATIONS
AUX 31 CLR V MEMORY
```

You can also enter the exact AUX number to go straight to the sub-menu.

Enter the AUX number directly

CLR D 3 B 1 AUX

```
AUX 3* V OPERATIONS
AUX 31 CLR V MEMORY
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 2* — RLL Operations

AUX 21-24

There are four AUX functions available that you can use to perform various operations on the control program.

- AUX 21 - Check Program
- AUX 22 - Change Reference
- AUX 23 - Clear Ladder Range
- AUX 24 - Clear Ladders

AUX 21 Check Program

Both the Handheld and *Direct*SOFT automatically check for errors during program entry. However, there may be occasions when you want to check a program that has already been in the CPU. Two types of checks are available:

- Syntax
- Duplicate References

The Syntax check will find a wide variety of programming errors, such as missing END statements, incomplete FOR/NEXT loops, etc. If you perform this check and get an error, see Appendix B for a complete listing of programming error codes. Correct the problem and then continue running the Syntax check until the message "NO SYNTAX ERROR appears.

Use the Duplicate Reference check to verify you have not used the same output coil reference more than once. Note, this AUX function will also find the same outputs even if they have been used with the OROUT instruction, which is perfectly acceptable.

This AUX function is available on the PLC Diagnostics sub-menu from within *Direct*SOFT.

AUX 22 Change Reference

There will be times when you need to change an I/O address reference or control relay reference. AUX 22 allows you to quickly and easily change all occurrences, (within an address range), of a specific instruction. For example, you can replace every instance of X5 with X10.

AUX 23 Clear Ladder Range

There have been many times when you take existing programs and add or remove certain portions to solve new application problems. By using AUX 23 you can select and delete a portion of the program. *Direct*SOFT does not have a menu option for this AUX function, but you can select the appropriate portion of the program and cut it with the editing tools.

AUX 24 Clear Ladders

AUX 24 clears the entire program from CPU memory. Before you enter a new program, you should always clear ladder memory. This AUX function is available on the PLC/Clear PLC sub-menu within *Direct*SOFT.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 3* — V-memory Operations

AUX 31

- AUX 31 - Clear V-memory

AUX 31 Clear V-Memory

AUX 31 clears all the information from the V-memory locations available for general use. This AUX function is available on the PLC/Clear PLC sub-menu within *Direct*SOFT.

AUX 4* — I/O Configuration

AUX 41-46

There are several AUX functions available that you can use to setup, view, or change the I/O configuration.

- AUX 41 — Show I/O Configuration
- AUX 42 — I/O Diagnostics
- AUX 44 — Power-up Configuration Check
- AUX 45 — Select Configuration
- AUX 46 — Configure I/O

AUX 41 Show I/O Configuration

This AUX function allows you to display the current I/O configuration. With the Handheld Programmer, you can scroll through each base and I/O slot to view the complete configuration. The configuration shows the type of module installed in each slot. *Direct*SOFT provides the same information, but it is much easier to view because you can view a complete base on one screen.

AUX 42 I/O Diagnostics

This is one of the most useful AUX functions available in the DL205 system. This AUX function will show you the exact base and slot location of any I/O module error that has occurred. This feature is also available within *Direct*SOFT under the PLC/Diagnostics sub-menu.

AUX 44 Power-up Configuration Check

Select this feature to quickly detect any changes that may have occurred while the power was disconnected. For example, if someone placed an output module in a slot that previously held an input module, the configuration check would detect the change.

If the system detects a change in the I/O configuration at power-up, an error code E252 NEW I/O CONFIGURATION will be generated. You can use AUX 42 to determine the exact base and slot location where the change occurred.



WARNING: You should always correct any I/O configuration errors before you place the CPU into RUN mode. Uncorrected errors can cause unpredictable machine operation that can result in a risk of personal injury or damage to equipment.

This feature is also available within *Direct*SOFT under the PLC/Setup sub-menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 45 Select Configuration

Even though the CPU can automatically detect configuration changes, you may actually want the new I/O configuration to be used. For example, you may have intentionally changed a module to use with a new program. You can use AUX 45 to select the new configuration, or, keep the existing configuration that is stored in memory. This feature is also available within *Direct*SOFT from the PLC/Setup sub-menu.



WARNING: Make sure the I/O configuration being selected will work properly with the CPU program. You should always correct any I/O configuration errors before you place the CPU into RUN mode. Uncorrected errors can cause unpredictable machine operation that can result in a risk of personal injury or damage to equipment.

AUX 46 Configure I/O

You will probably never need to use this feature, but the DL250–1 and DL260 CPU allows you to use AUX 46 to manually assign I/O addresses for any or all I/O slots on the local or expansion bases. It is generally much easier to do the I/O configuration operations from within *Direct*SOFT. The software package provides a really nice screen that is available from the PLC/Configure I/O sub-menu.

This feature is useful if you have a standard configuration you must sometimes change slightly to accommodate special requests. For example, you may require two adjacent input modules to have addresses starting at X10 and X200 respectively.

In automatic configuration, the addresses were assigned on 8-point boundaries. Manual configuration assumes that all modules are at least 16 points, so you can only assign addresses that are a multiple of 20 (octal). For example, X30 and Y50 would not be valid starting addresses for a module. X20 and Y40 are valid examples of starting addresses in a manual configuration. This does not mean you can only use 16 or 32 point modules with manual configuration. You can use 8 point modules, but 16 addresses will be assigned and 8 of them are unused.



WARNING: If you manually configure an I/O slot, the I/O addressing for the other modules will change. This is because the DL205 products do not allow you to assign duplicate I/O addresses. You should always correct any I/O configuration errors before you place the CPU into RUN mode. Uncorrected errors can cause unpredictable machine operation that can result in a risk of personal injury or damage to equipment.

Once you have manually configured the addresses for an I/O slot, the system will automatically retain these values even after a power cycle. You can remove any manual configuration changes by simply performing an automatic configuration.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 5* — CPU Configuration

AUX 51-5C

There are several AUX functions available that you can use to setup, view, or change the CPU configuration.

- AUX 51 — Modify Program Name
- AUX 52 — Display / Change Calendar
- AUX 53 — Display Scan Time
- AUX 54 — Initialize Scratchpad
- AUX 55 — Set Watchdog Timer
- AUX 56 — CPU Network Address
- AUX 57 — Set Retentive Ranges
- AUX 58 — Test Operations
- AUX 59 — Bit Override
- AUX 5B — Counter Interface Configuration
- AUX 5C — Display Error / Message History

AUX 51 Modify Program Name

The DL205 products can use a program name for the CPU program or a program stored on EEPROM in the Handheld Programmer. Note, you cannot have multiple programs stored on the EEPROM. The program name can be up to eight characters in length and can use any of the available characters (A–Z, 0–9). AUX 51 allows you to enter a program name. You can also perform this operation from within ***Direct***SOFT by using the PLC/Setup sub-menu. Once you've entered a program name, you can only clear the name by using AUX 54 to reset the system memory. Make sure you understand the possible ramifications of AUX 54 before you use it!

AUX 52 Display/Change Calendar

The DL240, DL250–1, and the DL260 CPUs have a clock and calendar feature. If you are using this, you can use the Handheld and AUX 52 to set the time and date. The following format is used.

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

You can use the AUX function to change any component of the date or time. However, 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).

You can also perform this operation from within ***Direct***SOFT by using the PLC/Setup sub-menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 53 Display Scan Time

AUX 53 displays the current, minimum, and maximum scan times. The minimum and maximum times are the ones that have occurred since the last Program Mode to Run Mode transition. You can also perform this operation from within *Direct*SOFT by using the PLC/Diagnostics sub-menu.

AUX 54 Initialize Scratchpad

The DL205 CPUs maintain system parameters in a memory area often referred to as the "scratchpad". In some cases, you may make changes to the system setup that will be stored in system memory. For example, if you specify a range of Control Relays (CRs) as retentive, these changes are stored.



NOTE: You may never have to use this feature unless you have made changes that affect system memory. Usually, you'll only need to initialize the system memory if you are changing programs and the old program required a special system setup. You can usually change from program to program without ever initializing system memory.

AUX 54 resets the system memory to the default values. You can also perform this operation from within *Direct*SOFT by using the PLC/Setup sub-menu.

AUX 55 Set Watchdog Timer

The DL205 CPUs 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 PGM mode. The Handheld displays the following message E003 S/W TIMEOUT when the scan overrun occurs.

Use AUX 55 to increase or decrease the watchdog timer value. You can also perform this operation from within *Direct*SOFT by using the PLC/Setup sub-menu.

AUX 56 CPU Network Address

Since the DL240, DL250–1 and DL260 CPUs have an additional communication port, you can use the Handheld to set the network address for the port and the port communication parameters. The default settings are:

- Station address 1
- HEX mode
- Odd parity

You can use this port with either the Handheld Programmer, *Direct*SOFT, or, as a *Direct*NET communication port. The *Direct*NET Manual provides additional information about communication settings required for network operation.



NOTE: You will only need to use this procedure if you have the bottom port connected to a network. Otherwise, the default settings will work fine.

Use AUX 56 to set the network address and communication parameters. You can also perform this operation from within *Direct*SOFT by using the PLC/Setup sub-menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 57 Set Retentive Ranges

The DL205 CPUs provide certain ranges of retentive memory by default. The default ranges are suitable for many applications, but you can change them if your application requires additional retentive ranges or no retentive ranges at all. The default settings are:

Memory Area	DL230		DL240		DL250-1		DL260	
	Default Range	Available Range	Default Range	Available Range	Default Range	Available Range	Default Range	Available Range
Control Relays	C300-C377	C0-C377	C300-C377	C0-C377	C1000-C1777	C0-C1777	C1000-C1777	C0-C3777
V-Memory	V2000-V7777	V0-V7777	V2000-V7777	V0-V7777	V1400-V3777	V0-V17777	V1400-V3777	V0-V37777
Timers	None by default	T0-T77	None by default	T0-T177	None by default	T0-T377	None by default	T0-T377
Counters	CT0-CT77	CT0-CT77	CT0-CT177	CT0-CT177	CT0-CT177	CT0-CT177	CT0-CT177	CT0-CT377
Stages	None by default	S0-S377	None by default	S0-S777	None by default	S0-S1777	None by default	S0-S1777

Use AUX 57 to change the retentive ranges. You can also perform this operation from within *Direct*SOFT by using the PLC/Setup sub-menu.

WARNING: The DL205 CPUs do not come with a battery. The super capacitor will retain the values in the event of a power loss, but only up to 1 week. The retention time may be less in some conditions. If the retentive ranges are important for your application, make sure you obtain the optional battery.

AUX 58 Test Operations

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, hold its last output state on the transition to TEST-PGM mode. The ability to hold the output states is especially useful, since it allows you to maintain key system I/O points for examination. See Chapter 9 for a description of the Test Modes.

You can use AUX 58 to configure each individual output. You can also perform this operation from within ***Direct***SOFT by using the PLC/Setup sub-menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A

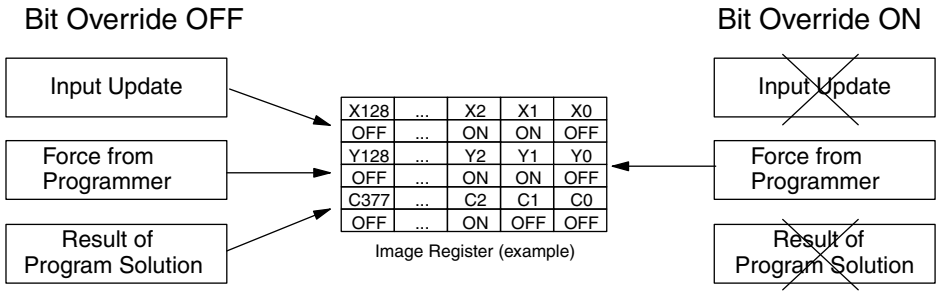
AUX 59 Bit Override

Bit override can be enabled on a point-by-point basis by using AUX 59 from the Handheld Programmer or, by a menu option from within *Direct*SOFT. 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, then the CPU will not change the state of X1. This means that even if X1 comes 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 would always be evaluated as "on".

*NOTE: **Direct**Net protocol does not support single bit write operations.*

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, then the CPU would not change the state of Y0. However, you can still use a programming device to change the status. Now, 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 diagram shows a brief overview of the bit override feature. Notice the CPU does not update the Image Register when bit override is enabled.



AUX 5B Counter Interface Configuration

AUX 5B is used with the DL205 Counter Interface module D2-CTRINT to select the module configuration. You can choose the type of counter, set the counter parameters, etc. See the DL205 Counter Interface Module manual for a complete description of how to select the various counter features.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 5C Display Error History

The DL240, DL250–1 and DL260 CPU will automatically log any system error codes and custom messages created with the FAULT instructions. The CPU logs the error code, date, and 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) of 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
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 use AUX Function 5C to show the error codes or messages. You can also view the errors and messages from within ***Direct***SOFT by using the PLC/Diagnostics sub-menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A

AUX 6* — Handheld Programmer Configuration

AUX 61, 62 and 65

There are several AUX functions available that you can use to setup, view, or change the Handheld Programmer configuration.

- AUX 61 — Show Revision Numbers
- AUX 62 — Beeper On/Off
- AUX 65 — Run Self Diagnostics

AUX 61 Show Revision Numbers

As with most industrial control products, there are cases when additional features and enhancements are made. Sometimes these new features only work with certain releases of firmware. By using AUX 61 you can quickly view the CPU and Handheld Programmer firmware revision numbers. This information (for the CPU) is also available from within *Direct*SOFT from the PLC/Diagnostics sub-menu.

AUX 62 Beeper On/Off

The Handheld has a beeper that provides confirmation of keystrokes. You can use Auxiliary (AUX) Function 62 to turn off the beeper.

AUX 65 Run Self Diagnostics

If you think the Handheld Programmer is not operating correctly, you can use AUX 65 to run a self diagnostics program. You can check the following items.

- Keypad
- Display
- LEDs and Backlight
- Handheld Programmer EEPROM check

AUX 7* - EEPROM Operations

AUX 71 - 76

There are several AUX functions available you can use to move programs between the CPU memory and an optional EEPROM installed in the Handheld Programmer.

- AUX 71 — Read from CPU memory to HPP EEPROM
- AUX 72 — Write HPP EEPROM to CPU
- AUX 73 — Compare CPU to HPP EEPROM
- AUX 74 — Blank Check (HPP EEPROM)
- AUX 75 — Erase HPP EEPROM
- AUX 76 — Show EEPROM Type (CPU and HPP)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Transferrable Memory Areas

Many of these AUX functions allow you to copy different areas of memory to and from the CPU and handheld programmer. The following table shows the areas that may be mentioned.

Option and Memory Type	DL240 Default Range	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	Non-selectable	Non-selectable

AUX 71 CPU to HPP EEPROM

AUX 71 copies information from the CPU memory to an EEPROM installed in the Handheld Programmer. You can copy different types of information from CPU memory as shown in the previous table.

AUX 72 HPP EEPROM to CPU

AUX 72 copies information from an EEPROM installed in the Handheld Programmer to the CPU. You can copy different portions of EEPROM (HP) memory to the CPU memory as shown in the previous table. The amount of data you can copy depends on the CPU.

AUX 73 Compare HPP EEPROM to CPU

AUX 73 compares the program in the Handheld programmer (EEPROM) with the CPU program. You can compare different types of information as shown previously. There is also an option called "etc." that allows you to check all of the areas sequentially without re-executing the AUX function every time.

AUX 74 HPP EEPROM Blank Check

AUX 74 allows you to check the EEPROM in the handheld programmer to make sure it is blank. It's a good idea to use this function anytime you start to copy an entire program to an EEPROM in the handheld programmer.

AUX 75 Erase HPP EEPROM

AUX 75 allows you to clear all data in the EEPROM in the handheld programmer. You should use this AUX function before you copy a program from the CPU.

AUX 76 Show EEPROM Type

You can use AUX 76 to quickly determine what size EEPROM is installed in the CPU and Handheld Programmer. The DL230 and DL240 use different size EEPROMs. See Chapter 3 for additional information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AUX 8* — Password Operations

AUX 81 - 83

There are AUX functions available that you can use to modify or enable the CPU password. You can use these features during on-line communications with the CPU, or, you can also use them with an EEPROM installed in the Handheld Programmer during off-line operation. This will allow you to develop a program in the Handheld Programmer and include password protection.

- AUX 81 — Modify Password
- AUX 82 — Unlock CPU
- AUX 83 — Lock CPU

AUX 81 Modify Password

You can use AUX 81 to provide an extra measure of protection by entering a password that prevents unauthorized machine operations. The password must be an eight-character numeric (0–9) code. Once you've entered a password, you can remove it by entering all zeros (00000000). This is the default from the factory.

Once you've entered a password, you can lock the CPU against access. There are two ways to lock the CPU with the Handheld Programmer.

- 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.

You can also enter or modify a password from within ***Direct***SOFT by using the PLC/Password sub-menu. This feature works slightly differently in ***Direct***SOFT. Once you've entered a password, the CPU is automatically locked when you exit the software package. It will also be locked if the CPU is power cycled.



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. If you do not remember the password, you have to return the CPU to the factory for password removal.



NOTE: The D2–240, DL250–1 and D2–260 CPUs support multi-level password protection of the ladder program. This allows password protection while not locking the communication port to an operator interface. The multi-level password can be invoked by creating a password with an upper case "A" followed by seven numeric characters (e.g. A1234567).

AUX 82 Unlock CPU

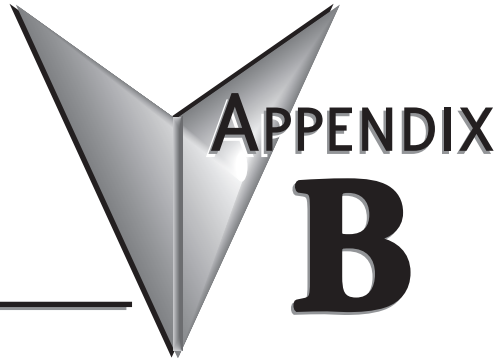
AUX 82 can be used to unlock a CPU that has been password protected. ***Direct***SOFT will automatically ask you to enter the password if you attempt to communicate with a CPU that contains a password.

AUX 83 Lock CPU

AUX 83 can be used to lock a CPU that contains a password. Once the CPU is locked, you will have to enter a password to gain access. Remember, this is not necessary with ***Direct***SOFT since the CPU is automatically locked whenever you exit the software package.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Error Codes



In This Appendix...

Error Code Table B–2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.
E041 CPU BATTERY LOW	The CPU battery is low and should be replaced. SP43 will be on and the error code will be stored in V7757.
E099 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.
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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Error Code	Description
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 insure 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.
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, DL250-1, DL260)	A NEXT instruction does not have the corresponding FOR instruction. SP52 will be on and the error code will be stored in V7755.
E405 MISSING NEXT (DL240/250–1/260)	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 (DL240/250–1/260)	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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Error Code	Description
E413 FOR/NEXT>64 (DL240/250–1/260)	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 allowed 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 (DL240/250–1/260)	Nested loops (programming one FOR/NEXT loop inside of another) is not allowed in the DL240/250–1/260 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 (DL240/250–1/260)	An 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 (DL240/250–1/260)	An 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 (DL240/250–1/260)	An 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.
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 (DL240/250–1/260)	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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Error Codes	Description
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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.
E490 MISSING SG INSTRUCTION	The BLK instruction is not immediately followed by the SG instruction.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Error Code	Description
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.
E499 PRINT INSTRUCTION	Invalid PRINT instruct usage. Quotations and/or spaces were not entered or entered incorrectly.
E501 BAD ENTRY	An invalid keystroke or series of keystrokes was 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B

DL205 Error Code	Description
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.
E525 MODE SWITCH (DL240/250–1/260)	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 key.
E527 ON LINE	The handheld programmer is in the ONLINE mode. To change to the OFFLINE mode use the MODE 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Error Code	Description
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. DV-1000: mismatch between MOVMC and LBL instructions.
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.
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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Instruction Execution Times



In This Appendix...

Introduction C–2
Boolean Instructions C–3
Comparative Boolean Instructions C–4
Bit of Word Boolean Instructions C–13
Immediate Instructions C–14
Timer, Counter and Shift Register Instructions C–15
Accumulator Data Instructions C–16
Logical Instructions C–18
Math Instructions C–20
Differential Instructions C–23
Bit Instructions C–24
Number Conversion Instructions C–25
Table Instructions C–25
CPU Control Instructions C–27
Program Control Instructions C–27
Interrupt Instructions C–28
Network Instructions C–28
Intelligent I/O Instructions C–28
Message Instructions C–29
RLLPLUS Instructions C–29
DRUM Instructions C–29
Clock / Calender Instructions C–30
Modbus Instructions C–30
ASCII Instructions C–30

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

This appendix contains several tables that provide the instruction execution times for the DL205 CPUs. One thing you will notice is that many of the execution times depend on the type of data being used with the instruction. For example, you'll notice that some of the instructions that use V-memory locations are further defined by the following items:

- Data Registers
- Bit Registers

V-Memory Data Registers

Some V-memory locations are considered data registers. For example, the V-memory locations that store the timer or counter current values, or just regular user V-memory would be considered as a V-memory data register. Don't think that you cannot load a bit pattern into these types of registers, you can. It's just that their primary use is as a data register. The following locations are considered as data registers.

Data Registers	DL230	DL240	DL250-1	DL260
Timer Current Values	V0 - V77	V0 - V177	V0 - V377	V0 - V377
Counter Current Values	V1000 - V1077	V1000 - V1177	V1000 - V1177	V1000 - V1377
User Data Words	V2000 - V2377 V4000 - V4177	V2000 - V3777 V4000 - V4377	V1400 - V7377 V10000 - V17777	V400 - V777 V1400 - V7377 V10000 - V35777

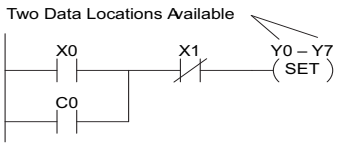
V-Memory Bit Registers

You may recall that some of the discrete points such as X, Y, C, etc. are automatically mapped into V-memory. The following locations that contain this data are considered bit registers.

Bit Registers	DL230	DL240	DL250-1	DL260
Input Points (X)	V40400 - V40407	V40400 - V40423	V40400 - V40437	V40400 - V40477
Output Points (Y)	V40500 - V40507	V40500 - V40523	V40500 - V40537	V40500 - V40577
Control Relays (C)	V40600 - V40617	V40600 - V40617	V40600 - V40677	V40600 - V40777
Timer Status Bits	V41100 - V41103	V41100 - V41107	V41100 - V41117	V41100 - V41177
Counter Status Bits	V41040 - V41143	V41040 - V41147	V41040 - V41147	V41140 - V41157
Stages	V41000 - V41017	V41000 - V41037	V41000 - V41077	V41000 - V41077

How to Read the Tables

Some of the instructions can have more than one parameter. For example, the SET instruction shown in the ladder program to the right can set a single point or a range of points.



In these cases, execution times depend on the amount and type of parameters. The execution time tables list execution times for both situations, as shown below:

SET	1st #: X, Y, C, 2nd #: X, Y, C, S (N pt)	9.2 µs 9.6 µs + 0.9 µs x N
RST	1st #: X, Y, C, 2nd #: X, Y, C, S (N pt)	9.2 µs 9.6 µs + 0.9 µs x N

Execution depends on numbers of locations and types of data used

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Boolean Instructions

Boolean Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
STR	X, Y, C, T, CT, S, SP	3.3 µs	3.3 µs	1.4 µs	1.4 µs	.67 µs	0 µs	.67 µs	0 µs
STRN	X, Y, C, T, CT, S, SP	3.9 µs	3.9 µs	1.6 µs	1.6 µs	.67 µs	0 µs	.67 µs	0 µs
OR	X, Y, C, T, CT, S, SP	2.7 µs	2.7 µs	1.0 µs	1.0 µs	.51 µs	.51 µs	.51 µs	.51 µs
ORN	X, Y, C, T, CT, S, SP	3.3 µs	3.3 µs	1.4 µs	1.4 µs	.55 µs	.55 µs	.55 µs	.55 µs
AND	X, Y, C, T, CT, S, SP	2.1 µs	2.1 µs	0.8 µs	0.8 µs	.42 µs	.42 µs	.42 µs	.42 µs
ANDN	X, Y, C, T, CT, S, SP	2.7 µs	2.7 µs	1.2 µs	1.2 µs	.51 µs	.51 µs	.51 µs	.51 µs
ANDSTR	None	1.2 µs	1.2 µs	0.7 µs	0.7 µs	.37 µs	.37 µs	.37 µs	.37 µs
ORSTR	None	1.2 µs	1.2 µs	0.7 µs	0.7 µs	.37 µs	.37 µs	.37 µs	.37 µs
OUT	X, Y, C	3.4 µs	3.4 µs	7.95 µs	7.65 µs	1.82 µs	1.82 µs	1.82 µs	1.82 µs
OROUT	X, Y, C	8.6 µs	8.6 µs	8.25 µs	8.4 µs	2.09 µs	2.09 µs	2.09 µs	2.09 µs
NOT		-	-	-	-	1.04 µs	1.04 µs	1.04 µs	1.04 µs
SET	1st #: X, Y, C, S, 2nd #: X, Y, C, S (N pt)	17.4 µs 12.0 µs + 5.4 µs x N	6.8 µs 6.8 µs	11.4 µs 11.0 µs + 7.0 µs x N	8.4 µs 8.4 µs	9.2 µs 9.6 µs + 0.9 µs x N	1.0 µs 1.1 µs	9.2 µs 9.6 µs + 0.9 µs x N	1.0 µs 1.1 µs
RST	1st #: X, Y, C, S 2nd #: X, Y, C, S (N pt)	17.7 µs 10.5 µs + 5.2 µs x N	6.8 µs 6.8 µs	11.4 µs 11.0 µs + 7.0 µs x N	8.4 µs 8.4 µs	9.2 µs 9.6 µs + 0.9 µs x N	1.0 µs 1.1 µs	9.2 µs 9.6 µs + 0.9 µs x N	1.0 µs 1.1 µs
	1st #: T, CT 2nd #: T, CT (N pt)	31.6 µs 17 µs + 14.6 µs x N	6.8 µs 6.8 µs	29.0 µs 24.3 µs + 4.7 µs x N	8.4 µs 8.4 µs	25.7 µs 16.8 µs + 2.7 µs x N	1.1 µs 1.4 µs	25.7 µs 16.8 µs + 2.7 µs x N	1.1 µs 1.4 µs
PAUSE	1wd: Y 2wd: Y (N points)	19.0 µs 15 µs + 4 µs x N	19.0 µs 15 µs + 4 µs x N	13.0 µs 11 µs + 3 µs x N	13.0 µs 11 µs + 3 µs x N	5.6 µs 9.2 µs + 0.3 µs x N	5.4 µs 4.8 µs	5.6 µs 9.2 µs + 0.3 µs x N	5.4 µs 4.8 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
STRE	***1st***	***2nd***								
	V: Data Reg.	V: Data Reg.	77 µs	13.8 µs	46 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	57 µs	13.8 µs	46 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	235 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	240 µs	13.8 µs	225 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	139 µs	13.8 µs	135 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	231 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	324 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.7 µs	27.7 µs	27.7 µs	27.7 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.7 µs	27.7 µs	27.7 µs	27.7 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
STRNE	***1st***	***2nd***								
	V: Data Reg.	V: Data Reg.	77 µs	13.8 µs	46 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	13.8 µs	136 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	57 µs	13.8 µs	46 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	235 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	240 µs	13.8 µs	225 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	139 µs	13.8 µs	135 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	231 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	324 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		V: Bit Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		V: Bit Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ORE	***1st***	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	140 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	234 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	239 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	230 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	324 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		V: Bit Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	50.4 µs	50.4 µs	50.4 µs	50.4 µs
		P: Indir. (Bit)	-	-	-	-	50.4 µs	50.4 µs	50.4 µs	50.4 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		V: Bit Reg.	-	-	-	-	30.3 µs	30.3 µs	30.3 µs	30.3 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	50.4 µs	50.4 µs	50.4 µs	50.4 µs
		P: Indir. (Bit)	-	-	-	-	50.4 µs	50.4 µs	50.4 µs	50.4 µs
ORNE	***1st***	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	234 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	239 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	230 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	323 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ANDE	***1st***	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	139 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	233 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	239 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	229 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	322 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
ANDNE	***1st***	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	133 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	139 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	233 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	239 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	229 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	323 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
STR	***1st***	***2nd***								
	T, CT	V: Data Reg.	76 µs	13.8 µs	46 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	57 µs	13.8 µs	46 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	235 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	1st	***2nd***								
	V: Data Reg.	V: Data Reg.	78 µs	13.8 µs	46 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	159 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	57 µs	13.8 µs	46 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	235 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	159 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	241 µs	13.8 µs	225 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	139 µs	13.8 µs	135 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	231 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	324 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
STRN	***1st***	***2nd***								
	T, CT	V: Data Reg.	78 µs	13.8 µs	46 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	13.8 µs	136 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	57 µs	13.8 µs	46 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	235 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	1st	***2nd***								
	V: Data Reg.	V: Data Reg.	78 µs	13.8 µs	46 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	159 µs	13.8 µs	135 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	57 µs	13.8 µs	46 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	235 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	159 µs	13.8 µs	136 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	241 µs	13.8 µs	225 µs	16.2 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	139 µs	13.8 µs	135 µs	16.2 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	231 µs	111.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	324 µs	115.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
OR	***1st***	***2nd***								
	T, CT	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	140 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	234 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	1st	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	140 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	234 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	240 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	230 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	323 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ORN	***1st***	***2nd***								
	T, CT	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	140 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	234 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	1st	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	141 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	234 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	240 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	230 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	324 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
AND	***1st***	***2nd***								
	T, CT	V: Data Reg.	76 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	139 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	233 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	1st	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	140 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	233 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	240 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	229 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	323 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Comparative Boolean Instructions (cont'd)

Comparative Boolean Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ANDN	***1st***	***2nd***								
	T, CT	V: Data Reg.	76 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	139 µs	110.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	233 µs	114.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	1st	***2nd***								
	V: Data Reg.	V: Data Reg.	75 µs	12.0 µs	44 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	55 µs	12.0 µs	44 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	139 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	233 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	V: Bit Reg.	V: Data Reg.	158 µs	12.0 µs	134 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		V: Bit Reg.	240 µs	12.0 µs	223 µs	13.9 µs	7.6 µs	7.6 µs	7.6 µs	7.6 µs
		K: Constant	137 µs	12.0 µs	133 µs	13.9 µs	4.8 µs	4.8 µs	4.8 µs	4.8 µs
		P: Indir. (Data)	-	-	229 µs	109.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	322 µs	113.0 µs	30.2 µs	30.2 µs	30.2 µs	30.2 µs
	P: Indir. (Data)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
	P: Indir. (Bit)	V: Data Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		V: Bit Reg.	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
		K: Constant	-	-	-	-	27.4 µs	27.4 µs	27.4 µs	27.4 µs
		P: Indir. (Data)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs
		P: Indir. (Bit)	-	-	-	-	51.0 µs	51.0 µs	51.0 µs	51.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit of Word Boolean Instructions

Bit of Word Boolean Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
STRB	V: Data Reg.	-	-	-	-	3.1 µs	3.1 µs	3.1 µs	3.1 µs
	V: Bit Reg.	-	-	-	-	3.1 µs	3.1 µs	3.1 µs	3.1 µs
	P: Indir. (Data)	-	-	-	-	30.0 µs	30.0 µs	30.0 µs	30.0 µs
	P: Indir. (Bit)	-	-	-	-	30.0 µs	30.0 µs	30.0 µs	30.0 µs
STRNB	V: Data Reg.	-	-	-	-	3.0 µs	3.0 µs	3.0 µs	3.0 µs
	V: Bit Reg.	-	-	-	-	3.0 µs	3.0 µs	3.0 µs	3.0 µs
	P: Indir. (Data)	-	-	-	-	29.8 µs	29.8 µs	29.8 µs	29.8 µs
	P: Indir. (Bit)	-	-	-	-	29.8 µs	29.8 µs	29.8 µs	29.8 µs
ORB	V: Data Reg.	-	-	-	-	2.9 µs	2.9 µs	2.9 µs	2.9 µs
	V: Bit Reg.	-	-	-	-	2.9 µs	2.9 µs	2.9 µs	2.9 µs
	P: Indir. (Data)	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
	P: Indir. (Bit)	-	-	-	-	29.9 µs	29.9 µs	29.9 µs	29.9 µs
ORNB	V: Data Reg.	-	-	-	-	2.8 µs	2.8 µs	2.8 µs	2.8 µs
	V: Bit Reg.	-	-	-	-	2.8 µs	2.8 µs	2.8 µs	2.8 µs
	P: Indir. (Data)	-	-	-	-	29.6 µs	29.6 µs	29.6 µs	29.6 µs
	P: Indir. (Bit)	-	-	-	-	29.6 µs	29.6 µs	29.6 µs	29.6 µs
ANDB	V: Data Reg.	-	-	-	-	2.8 µs	2.8 µs	2.8 µs	2.8 µs
	V: Bit Reg.	-	-	-	-	2.8 µs	2.8 µs	2.8 µs	2.8 µs
	P: Indir. (Data)	-	-	-	-	29.6 µs	29.6 µs	29.6 µs	29.6 µs
	P: Indir. (Bit)	-	-	-	-	29.6 µs	29.6 µs	29.6 µs	29.6 µs
ANDNB	V: Data Reg.	-	-	-	-	2.7 µs	2.7 µs	2.7 µs	2.7 µs
	V: Bit Reg.	-	-	-	-	2.7 µs	2.7 µs	2.7 µs	2.7 µs
	P: Indir. (Data)	-	-	-	-	29.6 µs	29.6 µs	29.6 µs	29.6 µs
	P: Indir. (Bit)	-	-	-	-	29.6 µs	29.6 µs	29.6 µs	29.6 µs
OUTB	V: Data Reg.	-	-	-	-	3.1 µs	3.4 µs	3.1 µs	3.4 µs
	V: Bit Reg.	-	-	-	-	3.1 µs	3.4 µs	3.1 µs	3.4 µs
	P: Indir. (Data)	-	-	-	-	30.3 µs	30.7 µs	30.3 µs	30.7 µs
	P: Indir. (Bit)	-	-	-	-	30.3 µs	30.7 µs	30.3 µs	30.7 µs
SETB	V: Data Reg.	-	-	-	-	13.4 µs	3.4 µs	13.4 µs	3.4 µs
	V: Bit Reg.	-	-	-	-	13.4 µs	3.4 µs	13.4 µs	3.4 µs
	P: Indir. (Data)	-	-	-	-	41.1 µs	29.1 µs	41.1 µs	29.1 µs
	P: Indir. (Bit)	-	-	-	-	41.1 µs	29.1 µs	41.1 µs	29.1 µs
RSTB	V: Data Reg.	-	-	-	-	13.5 µs	1.4 µs	13.5 µs	1.4 µs
	V: Bit Reg.	-	-	-	-	13.5 µs	1.4 µs	13.5 µs	1.4 µs
	P: Indir. (Data)	-	-	-	-	41.3 µs	29.1 µs	41.3 µs	29.1 µs
	P: Indir. (Bit)	-	-	-	-	41.3 µs	29.1 µs	41.3 µs	29.1 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Immediate Instructions

Immediate Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
LDI	V		-	-	-	-	-	-	20.6 µs	1.1 µs
LDIF	1st#:	X	-	-	-	-	-	-	-	-
	2nd#:	K:Constant	-	-	-	-	-	-	26.6 µs + 0.9 µs x N	1.4 µs
STRI	X		27 µs	9.8 µs	29 µs	10.7 µs	19.3 µs	19.3 µs	19.3 µs	19.3 µs
STRNI	X		26 µs	8.6 µs	29 µs	10.7 µs	19.4 µs	19.4 µs	19.4 µs	19.4 µs
ORI	X		27 µs	9.8 µs	29 µs	8.4 µs	19.1 µs	18.7 µs	19.1 µs	18.7 µs
ORNI	X		26 µs	8.6 µs	29 µs	8.4 µs	19.2 µs	18.9 µs	19.2 µs	18.9 µs
ANDI	X		25 µs	8.0 µs	27 µs	8.4 µs	18.7 µs	18.7 µs	18.7 µs	18.7 µs
ANDNI	X		24 µs	6.8 µs	28 µs	8.4 µs	18.8 µs	18.8 µs	18.8 µs	18.8 µs
OROUTI	Y		45 µs	45 µs	39 µs	40 µs	27.5 µs	27.5 µs	27.5 µs	27.5 µs
OUTI	Y		45 µs	45 µs	39 µs	40 µs	25.5 µs	25.5 µs	25.5 µs	25.5 µs
OUTIF	1st#:	Y	-	-	-	-	-	-	-	-
	2nd#:	K:Constant	-	-	-	-	-	-	66.1 µs + 0.9 µs x N	1.4 µs
SETI	1st#:	Y	25.5 µs	6.8 µs	39.0 µs	8.4 µs	23.1 µs	0.9 µs	23.1 µs	0.9 µs
	2nd#:	Y (N pt)	5.5 µs + 20 µs x N	6.8 µs	44 µs + 25 µs x N	8.4 µs	22.8 µs + 1.4 µs x N	0.9 µs	22.8 µs + 1.4 µs x N	0.9 µs
RSTI	1st#:	Y	25.5 µs	6.8 µs	37 µs	8.4 µs	23.2 µs	0.9 µs	23.2 µs	0.9 µs
	2nd#:	Y (N pt)	5 µs + 20.5 µs x N	6.8 µs	45 µs + 22 µs x N	8.4 µs	22.8 µs + 1.4 µs x N	0.9 µs	22.8 µs + 1.4 µs x N	0.9 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Timer, Counter and Shift Register Instructions

Timer, Counter and Shift Register Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
	1st	*2nd*								
TMR	T	V: Data Reg.	75 µs	31 µs	61 µs	23.5 µs	26.8 µs	7.3 µs	26.8 µs	7.3 µs
		V: Bit Reg.	158 µs	31 µs	158 µs	23.5 µs	26.8 µs	7.3 µs	26.8 µs	7.3 µs
		K: Constant	66 µs	31 µs	70 µs	23.5 µs	20.0 µs	4.8 µs	20.0 µs	4.8 µs
		P: Indir. (Data)	-	-	177 µs	131.0 µs	45.6 µs	30.2 µs	45.6 µs	30.2 µs
		P: Indir. (Bit)	-	-	271 µs	136.0 µs	45.6 µs	30.2 µs	45.6 µs	30.2 µs
TMRF	T	V: Data Reg.	75 µs	31 µs	61 µs	23.5 µs	51.4 µs	7.3 µs	51.4 µs	7.3 µs
		V: Bit Reg.	158 µs	31 µs	158 µs	23.5 µs	51.4 µs	7.3 µs	51.4 µs	7.3 µs
		K: Constant	66 µs	31 µs	70 µs	23.5 µs	48.4 µs	4.6 µs	48.4 µs	4.6 µs
		P: Indir. (Data)	-	-	177 µs	131.0 µs	75.9 µs	30.2 µs	75.9 µs	30.2 µs
		P: Indir. (Bit)	-	-	271 µs	136.0 µs	75.9 µs	30.2 µs	75.9 µs	30.2 µs
TMRA	T	V: Data Reg.	94 µs	56 µs	75 µs	41 µs	48.9 µs	7.3 µs	48.9 µs	7.3 µs
		V: Bit Reg.	304 µs	264 µs	253 µs	219 µs	48.9 µs	7.3 µs	48.9 µs	7.3 µs
		K: Constant	95 µs	45 µs	79 µs	49 µs	45.0 µs	4.6 µs	45.0 µs	4.6 µs
		P: Indir. (Data)	-	-	193 µs	159 µs	75.9 µs	30.2 µs	75.9 µs	30.2 µs
		P: Indir. (Bit)	-	-	366 µs	331 µs	75.9 µs	30.2 µs	75.9 µs	30.2 µs
TMRAF	T	V: Data Reg.	98 µs	54 µs	75 µs	42 µs	54.2 µs	7.3 µs	54.2 µs	7.3 µs
		V: Bit Reg.	304 µs	264 µs	253 µs	218 µs	54.2 µs	7.3 µs	54.2 µs	7.3 µs
		K: Constant	95 µs	49 µs	80 µs	50 µs	50.3 µs	4.6 µs	50.3 µs	4.6 µs
		P: Indir. (Data)	-	-	193 µs	159 µs	81.2 µs	30.2 µs	81.2 µs	30.2 µs
		P: Indir. (Bit)	-	-	366 µs	331 µs	81.2 µs	30.2 µs	81.2 µs	30.2 µs
CNT	CT	V: Data Reg.	68 µs	61 µs	59 µs	38 µs	25.8 µs	7.3 µs	25.8 µs	7.3 µs
		V: Bit Reg.	148 µs	141 µs	157 µs	133 µs	25.8 µs	7.3 µs	25.8 µs	7.3 µs
		K: Constant	56 µs	45 µs	59 µs	45 µs	22.2 µs	4.6 µs	22.2 µs	4.6 µs
		P: Indir. (Data)	-	-	176 µs	152 µs	53.5 µs	30.2 µs	53.5 µs	30.2 µs
		P: Indir. (Bit)	-	-	270 µs	245 µs	53.5 µs	30.2 µs	53.5 µs	30.2 µs
SGCNT	CT	V: Data Reg.	57 µs	64 µs	58 µs	38 µs	27.3 µs	7.3 µs	27.3 µs	7.3 µs
		V: Bit Reg.	140 µs	148 µs	155 µs	133 µs	27.3 µs	7.3 µs	27.3 µs	7.3 µs
		K: Constant	46 µs	53 µs	67 µs	45 µs	23.5 µs	4.6 µs	23.5 µs	4.6 µs
		P: Indir. (Data)	-	-	175 µs	152 µs	54.9 µs	30.2 µs	54.9 µs	30.2 µs
		P: Indir. (Bit)	-	-	268 µs	245 µs	54.9 µs	30.2 µs	54.9 µs	30.2 µs
UDC	CT	V: Data Reg.	103 µs	74 µs	80 µs	56 µs	39.8 µs	7.3 µs	39.8 µs	7.3 µs
		V: Bit Reg.	310 µs	281 µs	261 µs	224 µs	39.8 µs	7.3 µs	39.8 µs	7.3 µs
		K: Constant	102 µs	70 µs	97 µs	60 µs	35.4 µs	4.6 µs	35.4 µs	4.6 µs
		P: Indir. (Data)	-	-	202 µs	165 µs	67.8 µs	30.2 µs	67.8 µs	30.2 µs
		P: Indir. (Bit)	-	-	374 µs	336 µs	67.8 µs	30.2 µs	67.8 µs	30.2 µs
SR	C (N points to shift)		30 µs + 4.6 µs x N	17.2 µs	25 µs + 4 µs x N	19.7 µs	17.8 µs + 0.9 µs x N	9.8 µs	17.8 µs + 0.9 µs x N	9.8 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Accumulator Data Instructions

Accumulator/Stack Load and Output Data Instructions			DL230		DL240		DL250-1		DL260	
Instruction	**Legal Data Types**		**Execute**	**Not Execute**	**Execute**	**Not Execute**	**Execute**	**Not Execute**	**Execute**	**Not Execute**
LD	V: Data Reg.		68 µs	8.4 µs	68 µs	8.4 µs	11.8 µs	1.0 µs	11.8 µs	1.0 µs
	V: Bit Reg.		149 µs	8.4 µs	143 µs	8.4 µs	11.8 µs	1.0 µs	11.8 µs	1.0 µs
	K: Constant		62 µs	8.4 µs	159 µs	8.4 µs	9.0 µs	1.0 µs	9.0 µs	1.0 µs
	P: Indir. (Data)		169 µs	8.4 µs	238 µs	8.4 µs	33.9 µs	0.9 µs	33.9 µs	0.9 µs
	P: Indir. (Bit)		256 µs	8.4 µs	62 µs	8.4 µs	33.9 µs	0.9 µs	33.9 µs	0.9 µs
LDA	O: (Octal constant for address)		58 µs	8.4 µs	56 µs	8.4 µs	10.4 µs	1.0 µs	10.4 µs	1.0 µs
LDD	V: Data Reg.		72 µs	8.4 µs	67 µs	8.4 µs	12.2 µs	1.0 µs	12.2 µs	1.0 µs
	V: Bit Reg.		266 µs	8.4 µs	228 µs	8.4 µs	12.2 µs	1.0 µs	12.2 µs	1.0 µs
	K: Constant		64 µs	8.4 µs	69 µs	8.4 µs	9.0 µs	1.0 µs	9.0 µs	1.0 µs
	P: Indir. (Data)		172 µs	8.4 µs	158 µs	8.4 µs	37.8 µs	0.9 µs	37.8 µs	0.9 µs
	P: Indir. (Bit)		373 µs	8.4 µs	323 µs	8.4 µs	37.8 µs	0.9 µs	37.8 µs	0.9 µs
LDF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP	K: Constant	-	-	86 µs + 5 µs x N	8.4 µs	20.5 µs + 0.9 µs x N	0.9 µs	20.5 µs + 0.9 µs x N	0.9 µs
LDR	V: Data Reg.		-	-	-	-	29.5 µs	1.0 µs	29.5 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	29.5 µs	1.0 µs	29.5 µs	1.0 µs
	K: Constant		-	-	-	-	25.5 µs	1.0 µs	25.5 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	54.9 µs	1.0 µs	54.9 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	54.9 µs	1.0 µs	54.9 µs	1.0 µs
LDSX	K: Constant		-	-	79 µs	8.4 µs	14.6 µs	1.0 µs	14.6 µs	1.0 µs
LDX	V: Data Reg.		-	-	-	-	10.8 µs	1.0 µs	10.8 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	10.8 µs	1.0 µs	10.8 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	45.2 µs	1.0 µs	45.2 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	45.2 µs	1.0 µs	45.2 µs	1.0 µs
OUT	V: Data Reg.		60 µs	8.4 µs	21 µs	8.4 µs	9.3 µs	1.0 µs	9.3 µs	1.0 µs
	V: Bit Reg.		132 µs	8.4 µs	126 µs	8.4 µs	9.3 µs	1.0 µs	9.3 µs	1.0 µs
	P: Indir. (Data)		162 µs	8.4 µs	112 µs	8.4 µs	35.2 µs	0.9 µs	35.2 µs	0.9 µs
	P: Indir. (Bit)		239 µs	8.4 µs	222 µs	8.4 µs	35.2 µs	0.9 µs	35.2 µs	0.9 µs
OUTD	V: Data Reg.		68 µs	8.4 µs	26 µs	8.4 µs	10.2 µs	1.0 µs	10.2 µs	1.0 µs
	V: Bit Reg.		276 µs	8.4 µs	235 µs	8.4 µs	10.2 µs	1.0 µs	10.2 µs	1.0 µs
	P: Indir. (Data)		196 µs	8.4 µs	116 µs	8.4 µs	35.8 µs	0.9 µs	35.8 µs	0.9 µs
	P: Indir. (Bit)		384 µs	8.4 µs	331 µs	8.4 µs	35.8 µs	0.9 µs	35.8 µs	0.9 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Accumulator Data Instructions (cont'd)

Accumulator/Stack Load and Output Data Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
OUTF	1st	2nd								
	X, Y, C	K: Constant	-	-	53 µs + 7 µs x N	8.4 µs	54 µs + 1.0 µs x N	0.9 µs	54 µs + 1.0 µs x N	0.9 µs
OUTL	V: Data Reg.		-	-	-	-	-	-	13.5 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	-	-	13.5 µs	1.0 µs
OUTM	V: Data Reg.		-	-	-	-	-	-	13.7 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	-	-	13.7 µs	1.0 µs
OUTX	V: Data Reg.		-	-	-	-	-	-	17.2 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	-	-	17.2 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	-	-	43.4 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	-	-	43.4 µs	1.0 µs
POP	None		55 µs	7.2 µs	50 µs	8.4 µs	8.4 µs	1.0 µs	8.4 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Logical Instructions

Logical (Accumulator) Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
AND	V: Data Reg.		58 µs	10.4 µs	54 µs	8.4 µs	7.9 µs	1.0 µs	7.9 µs	1.0 µs
	V: Bit Reg.		261 µs	10.4 µs	145 µs	8.4 µs	7.9 µs	1.0 µs	7.9 µs	1.0 µs
	P: Indir. (Data)		-	-	162 µs	8.4 µs	33.4 µs	0.9 µs	33.4 µs	0.9 µs
	P: Indir. (Bit)		-	-	241 µs	8.4 µs	33.4 µs	0.9 µs	33.4 µs	0.9 µs
ANDD	V: Data Reg.		-	-	-	-	8.9 µs	1.0 µs	8.9 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	8.9 µs	1.0 µs	8.9 µs	1.0 µs
	K: Constant		53 µs	8.4 µs	60 µs	8.4 µs	5.7 µs	1.0 µs	5.7 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	34.4 µs	0.9 µs	34.4 µs	0.9 µs
	P: Indir. (Bit)		-	-	-	-	34.4 µs	0.9 µs	34.4 µs	0.9 µs
ANDF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	21.6 µs + 0.9 µs x N	1.0 µs	21.6 µs + 0.9 µs x N	1.0 µs
ANDS	None		-	-	-	-	-	-	10.0 µs	1.0 µs
OR	V: Data Reg.		59 µs	10.4 µs	54 µs	8.4 µs	8.1 µs	1.0 µs	8.1 µs	1.0 µs
	V: Bit Reg.		257 µs	10.4 µs	144 µs	8.4 µs	8.1 µs	1.0 µs	8.1 µs	1.0 µs
	P: Indir. (Data)		-	-	160 µs	8.4 µs	33.8 µs	0.9 µs	33.8 µs	0.9 µs
	P: Indir. (Bit)		-	-	239 µs	8.4 µs	33.8 µs	0.9 µs	33.8 µs	0.9 µs
ORD	V: Data Reg.		-	-	-	-	9.0 µs	1.0 µs	9.0 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	9.0 µs	1.0 µs	9.0 µs	1.0 µs
	K: Constant		49 µs	8.4 µs	60 µs	8.4 µs	5.8 µs	1.0 µs	5.8 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	34.5 µs	0.9 µs	34.5 µs	0.9 µs
	P: Indir. (Bit)		-	-	-	-	34.5 µs	0.9 µs	34.5 µs	0.9 µs
ORF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	20.9 µs + 0.9 µs x N	1.0 µs	20.9 µs + 0.9 µs x N	1.0 µs
ORS	None		-	-	-	-	-	-	10.2 µs	1.0 µs
XOR	V: Data Reg.		60 µs	10.4 µs	69 µs	8.4 µs	8.0 µs	1.0 µs	8.0 µs	1.0 µs
	V: Bit Reg.		257 µs	10.4 µs	144 µs	8.4 µs	8.0 µs	1.0 µs	8.0 µs	1.0 µs
	P: Indir. (Data)		-	-	160 µs	8.4 µs	33.6 µs	0.9 µs	33.6 µs	0.9 µs
	P: Indir. (Bit)		-	-	239 µs	8.4 µs	33.6 µs	0.9 µs	33.6 µs	0.9 µs
XORD	V: Data Reg.		-	-	-	-	9.0 µs	1.0 µs	9.0 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	9.0 µs	1.0 µs	9.0 µs	1.0 µs
	K: Constant		49 µs	8.4 µs	62 µs	8.4 µs	5.4 µs	1.0 µs	5.4 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	34.4 µs	0.9 µs	34.4 µs	0.9 µs
	P: Indir. (Bit)		-	-	-	-	34.4 µs	0.9 µs	34.4 µs	0.9 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Logical Instructions (cont'd)

Logical (Accumulator) Instructions			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
XORF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	20.9 µs + 0.9 µs x N	1.0 µs	20.9 µs + 0.9 µs x N	1.0 µs
XORS	None		-	-	-	-	-	-	10.1 µs	1.0 µs
CMP	V: Data Reg.		59 µs	10.4 µs	69 µs	8.4 µs	9.4 µs	1.0 µs	9.4 µs	1.0 µs
	V: Bit Reg.		259 µs	10.4 µs	115 µs	8.4 µs	9.4 µs	1.0 µs	9.4 µs	1.0 µs
	P: Indir. (Data)		-	-	130 µs	8.4 µs	34.9 µs	0.9 µs	34.9 µs	0.9 µs
	P: Indir. (Bit)		-	-	211 µs	8.4 µs	34.9 µs	0.9 µs	34.9 µs	0.9 µs
CMPD	V: Data Reg.		63 µs	8.4 µs	47 µs	8.4 µs	9.9 µs	1.0 µs	9.9 µs	1.0 µs
	V: Bit Reg.		257 µs	8.4 µs	206 µs	8.4 µs	9.9 µs	1.0 µs	9.9 µs	1.0 µs
	K: Constant		54 µs	8.4 µs	49 µs	8.4 µs	6.7 µs	1.0 µs	6.7 µs	1.0 µs
	P: Indir. (Data)		-	-	133 µs	8.4 µs	35.4 µs	1.0 µs	35.4 µs	1.0 µs
	P: Indir. (Bit)		-	-	303 µs	8.4 µs	35.4 µs	1.0 µs	35.4 µs	1.0 µs
CMPF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	29.2 µs+ 1.0 µs x N	1.0 µs	29.2 µs+ 1.0 µs x N	1.0 µs
CMPR	V: Data Reg.		-	-	-	-	42.8 µs	1.0 µs	42.8 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	42.8 µs	1.0 µs	42.8 µs	1.0 µs
	K: Constant		-	-	-	-	38.4 µs	1.0 µs	38.4 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	69.0 µs	1.0 µs	69.0 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	69.0 µs	1.0 µs	69.0 µs	1.0 µs
CMPS	None		-	-	-	-	-	-	11.2 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Math Instructions

Math Instructions (Accumulator)		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ADD	V: Data Reg.	198 µs	10.6 µs	291 µs	8.4 µs	78.4 µs	0.9 µs	78.4 µs	0.9 µs
	V: Bit Reg.	397 µs	10.6 µs	363 µs	8.4 µs	78.4 µs	0.9 µs	78.4 µs	0.9 µs
	P: Indir. (Data)	-	-	441 µs	8.4 µs	101.2 µs	0.9 µs	101.2 µs	0.9 µs
	P: Indir. (Bit)	-	-	520 µs	8.4 µs	101.2 µs	0.9 µs	101.2 µs	0.9 µs
ADDD	V: Data Reg.	198 µs	8.4 µs	291 µs	8.4 µs	83.3 µs	0.9 µs	83.3 µs	0.9 µs
	V: Bit Reg.	397 µs	8.4 µs	512 µs	8.4 µs	83.3 µs	0.9 µs	83.3 µs	0.9 µs
	K: Constant	188 µs	8.4 µs	298 µs	8.4 µs	67.7 µs	0.9 µs	67.7 µs	0.9 µs
	P: Indir. (Data)	-	-	442 µs	8.4 µs	101.2 µs	0.9 µs	101.2 µs	0.9 µs
	P: Indir. (Bit)	-	-	608 µs	8.4 µs	101.2 µs	0.9 µs	101.2 µs	0.9 µs
SUB	V: Data Reg.	200 µs	10.6 µs	287 µs	8.4 µs	77.4 µs	0.9 µs	77.4 µs	0.9 µs
	V: Bit Reg.	397 µs	10.6 µs	360 µs	8.4 µs	77.4 µs	0.9 µs	77.4 µs	0.9 µs
	P: Indir. (Data)	-	-	434 µs	8.4 µs	95.1 µs	0.9 µs	95.1 µs	0.9 µs
	P: Indir. (Bit)	-	-	513 µs	8.4 µs	95.1 µs	0.9 µs	95.1 µs	0.9 µs
SUBD	V: Data Reg.	198 µs	8.4 µs	288 µs	8.4 µs	82.5 µs	0.9 µs	82.5 µs	0.9 µs
	V: Bit Reg.	392 µs	8.4 µs	504 µs	8.4 µs	82.5 µs	0.9 µs	82.5 µs	0.9 µs
	K: Constant	190 µs	8.4 µs	294 µs	8.4 µs	66.0 µs	0.9 µs	66.0 µs	0.9 µs
	P: Indir. (Data)	-	-	434 µs	8.4 µs	99.7 µs	0.9 µs	99.7 µs	0.9 µs
	P: Indir. (Bit)	-	-	600 µs	8.4 µs	99.7 µs	0.9 µs	99.7 µs	0.9 µs
MUL	V: Data Reg.	497 µs	10.6 µs	311 µs	8.4 µs	266.1 µs	0.9 µs	266.1 µs	0.9 µs
	V: Bit Reg.	483 µs	10.6 µs	385 µs	8.4 µs	266.1 µs	0.9 µs	266.1 µs	0.9 µs
	K: Constant	487 µs	8.4 µs	334 µs	8.4 µs	286.9 µs	0.9 µs	286.9 µs	0.9 µs
	P: Indir. (Data)	-	-	401 µs	8.4 µs	290.0 µs	0.9 µs	290.0 µs	0.9 µs
	P: Indir. (Bit)	-	-	461 µs	8.4 µs	290.0 µs	0.9 µs	290.0 µs	0.9 µs
MULD	V: Data Reg.					839.1 µs	0.9 µs	839.1 µs	0.9 µs
	V: Bit Reg.	-	-	-	-	839.1 µs	0.9 µs	839.1 µs	0.9 µs
	P: Indir. (Data)	-	-	-	-	863.1 µs	0.9 µs	863.1 µs	0.9 µs
	P: Indir. (Bit)					863.1 µs	0.9 µs	863.1 µs	0.9 µs
DIV	V: Data Reg.	909 µs	10.6 µs	601 µs	8.4 µs	363.9 µs	0.9 µs	363.9 µs	0.9 µs
	V: Bit Reg.	1108 µs	10.6 µs	675 µs	8.4 µs	363.9 µs	0.9 µs	363.9 µs	0.9 µs
	K: Constant	699 µs	8.4 µs	573 µs	8.4 µs	384.4 µs	0.9 µs	384.4 µs	0.9 µs
	P: Indir. (Data)	-	-	691 µs	8.4 µs	419.8 µs	0.9 µs	419.8 µs	0.9 µs
	P: Indir. (Bit)	-	-	771 µs	8.4 µs	419.8 µs	0.9 µs	419.8 µs	0.9 µs
DIVD	V: Data Reg.					398.3 µs	0.9 µs	398.3 µs	0.9 µs
	V: Bit Reg.	-	-	-	-	398.3 µs	0.9 µs	398.3 µs	0.9 µs
	P: Indir. (Data)	-	-	-	-	390.9 µs	0.9 µs	390.9 µs	0.9 µs
	P: Indir. (Bit)					390.9 µs	0.9 µs	390.9 µs	0.9 µs
INC	V: Data Reg.					48.5 µs	1.0 µs	48.5 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	48.5 µs	1.0 µs	48.5 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	74.7 µs	1.0 µs	74.7 µs	1.0 µs
	P: Indir. (Bit)					74.7 µs	1.0 µs	74.7 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Math Instructions (cont'd)

Math Instructions (Accumulator)		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
DEC	V: Data Reg.	-	-	-	-	47.5 µs	1.0 µs	47.5 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	47.5 µs	1.0 µs	47.5 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	71.5 µs	1.0 µs	71.5 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	71.5 µs	1.0 µs	71.5 µs	1.0 µs
INCB	V: Data Reg.	88 µs	10.4 µs	35 µs	8.4 µs	13.2 µs	1.0 µs	13.2 µs	1.0 µs
	V: Bit Reg.	349 µs	10.4 µs	211 µs	8.4 µs	13.2 µs	1.0 µs	13.2 µs	1.0 µs
	P: Indir. (Data)	-	-	126 µs	8.4 µs	38.6 µs	0.9 µs	38.6 µs	0.9 µs
	P: Indir. (Bit)	-	-	307 µs	8.4 µs	38.6 µs	0.9 µs	38.6 µs	0.9 µs
DECB	V: Data Reg.	82 µs	10.4 µs	33 µs	8.4 µs	13.2 µs	1.0 µs	13.2 µs	1.0 µs
	V: Bit Reg.	351 µs	10.4 µs	210 µs	8.4 µs	13.2 µs	1.0 µs	13.2 µs	1.0 µs
	P: Indir. (Data)	-	-	123 µs	8.4 µs	38.0 µs	0.9 µs	38.0 µs	0.9 µs
	P: Indir. (Bit)	-	-	304 µs	8.4 µs	38.0 µs	0.9 µs	38.0 µs	0.9 µs
ADDB	V: Data Reg.	-	-	-	-	24.9 µs	1.0 µs	24.9 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	24.9 µs	1.0 µs	24.9 µs	1.0 µs
	K: Constant	-	-	-	-	23.5 µs	1.0 µs	23.5 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	51.1 µs	1.0 µs	51.1 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	51.1 µs	1.0 µs	51.1 µs	1.0 µs
ADDBD	V: Data Reg.	-	-	-	-	-	-	24.4 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	-	-	24.4 µs	1.0 µs
	K: Constant	-	-	-	-	-	-	20.7 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	-	-	50.7 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	-	-	50.7 µs	1.0 µs
SUBB	V: Data Reg.	-	-	-	-	24.7 µs	1.0 µs	24.7 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	24.7 µs	1.0 µs	24.7 µs	1.0 µs
	K: Constant	-	-	-	-	23.3 µs	1.0 µs	23.3 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	50.6 µs	1.0 µs	50.6 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	50.6 µs	1.0 µs	50.6 µs	1.0 µs
SUBBD	V: Data Reg.	-	-	-	-	-	-	24.2 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	-	-	24.2 µs	1.0 µs
	K: Constant	-	-	-	-	-	-	20.2 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	-	-	50.2 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	-	-	50.2 µs	1.0 µs
MULB	V: Data Reg.	-	-	-	-	10.8 µs	1.0 µs	10.8 µs	1.0 µs
	V: Bit Reg.	-	-	-	-	10.8 µs	1.0 µs	10.8 µs	1.0 µs
	K: Constant	-	-	-	-	8.2 µs	1.0 µs	8.2 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	37.1 µs	1.0 µs	37.1 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	37.1 µs	1.0 µs	37.1 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Math Instructions (cont'd)

Math Instructions (Accumulator)			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
DIVB	V: Data Reg.		-	-	-	-	28.7 µs	1.0 µs	28.7 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	28.7 µs	1.0 µs	28.7 µs	1.0 µs
	K: Constant		-	-	-	-	26.1 µs	1.0 µs	26.1 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	54.9 µs	1.0 µs	54.9 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	54.9 µs	1.0 µs	54.9 µs	1.0 µs
ADDR	V: Data Reg.		-	-	-	-	48.1 µs	1.0 µs	48.1 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	48.1 µs	1.0 µs	48.1 µs	1.0 µs
	K: Constant		-	-	-	-	41.7 µs	1.0 µs	41.7 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	74.3 µs	1.0 µs	74.3 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	74.3 µs	1.0 µs	74.3 µs	1.0 µs
SUBR	V: Data Reg.		-	-	-	-	50.1 µs	1.0 µs	50.1 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	50.1 µs	1.0 µs	50.1 µs	1.0 µs
	K: Constant		-	-	-	-	58.7 µs	1.0 µs	58.7 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	76.3 µs	1.0 µs	76.3 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	76.3 µs	1.0 µs	76.3 µs	1.0 µs
MULR	V: Data Reg.		-	-	-	-	54.2 µs	1.0 µs	54.2 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	54.2 µs	1.0 µs	54.2 µs	1.0 µs
	K: Constant		-	-	-	-	42.7 µs	1.0 µs	42.7 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	80.4 µs	1.0 µs	80.4 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	80.4 µs	1.0 µs	80.4 µs	1.0 µs
DIVR	V: Data Reg.		-	-	-	-	50.1 µs	1.0 µs	50.1 µs	1.0 µs
	V: Bit Reg.		-	-	-	-	50.1 µs	1.0 µs	50.1 µs	1.0 µs
	K: Constant		-	-	-	-	58.7 µs	1.0 µs	58.7 µs	1.0 µs
	P: Indir. (Data)		-	-	-	-	76.3 µs	1.0 µs	76.3 µs	1.0 µs
	P: Indir. (Bit)		-	-	-	-	76.3 µs	1.0 µs	76.3 µs	1.0 µs
ADDF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	-	-	109.3 µs + 0.9 µs x N	1.0 µs
SUBF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	-	-	107.3 µs + 0.9 µs x N	1.0 µs
MULF	**1st**	**2nd**								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	-	-	352.5 µs + 0.8 µs x N	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Math Instructions (cont'd)

Math Instructions Accumulator			DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
DIVF	1st	2nd								
	X, Y, C, S, T, CT, SP GX, GY	K: Constant	-	-	-	-	-	-	477.3 µs + 0.8 µs x N	1.0 µs
ADDS	None		-	-	-	-	-	-	99.5 µs	1.0 µs
SUBS	None		-	-	-	-	-	-	97.5 µs	1.0 µs
MULS	None		-	-	-	-	-	-	342.5 µs	1.0 µs
DIVS	None		-	-	-	-	-	-	467.3 µs	1.0 µs
ADDBS	None		-	-	-	-	-	-	24.3 µs	1.0 µs
SUBBS	None		-	-	-	-	-	-	23.7 µs	1.0 µs
MULBS	None		-	-	-	-	-	-	11.7 µs	1.0 µs
DIVBS	None		-	-	-	-	-	-	29.7 µs	1.0 µs
SQRTR	None		-	-	-	-	-	-	87.9 µs	1.0 µs
SINR	None		-	-	-	-	-	-	226.8 µs	1.0 µs
COSR	None		-	-	-	-	-	-	213.1 µs	1.0 µs
TANR	None		-	-	-	-	-	-	285.5 µs	1.0 µs
ASINR	None		-	-	-	-	-	-	489.8 µs	1.0 µs
ACOSR	None		-	-	-	-	-	-	508.3 µs	1.0 µs
ATANR	None		-	-	-	-	-	-	317.1 µs	1.0 µs

Differential Instructions

Differential Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
PD	X, Y, C	13.5 µs	13.5 µs	15.9 µs	14.6 µs	14.4 µs	14.4 µs	14.4 µs	14.4 µs
STRPD	X, Y, C, S, T, CT	-	-	-	-	5.4 µs	5.4 µs	5.4 µs	5.4 µs
STRND	X, Y, C, S, T, CT	-	-	-	-	7.3 µs	7.3 µs	7.3 µs	7.3 µs
ORPD	X, Y, C, S, T, CT	-	-	-	-	6.8 µs	5.2 µs	6.8 µs	5.2 µs
ORND	X, Y, C, S, T, CT	-	-	-	-	7.1 µs	4.9 µs	7.1 µs	4.9 µs
ANDPD	X, Y, C, S, T, CT	-	-	-	-	6.8 µs	5.2 µs	6.8 µs	5.2 µs
ANDND	X, Y, C, S, T, CT	-	-	-	-	7.1 µs	4.9 µs	7.1 µs	4.9 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bit Instructions

Bit Instructions (Accumulator)		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
SUM	None	-	-	-	-	-	-	6.7 µs	1.0 µs
SHFR	V: Data Reg. (N bits)	44 µs + 14 .6 µs x N	10.4 µs	35 µs + 6 µs x N	8.4 µs	12.1 µs + 0.1 µs x N	0.9 µs	12.1 µs + 0.1 µs x N	0.9 µs
	V: Bit Reg (N bits)	243 µs + 14.6 µs x N	8.4 µs	110 µs + 6 µs x N	8.4 µs	12.1 µs + 0.1 µs x N	0.9 µs	12.1 µs + 0.1 µs x N	0.9 µs
	K: Constant (N bits)	34 µs + 14.6 µs x N	8.4 µs	35 µs + 6 µs x N	8.4 µs	8.4 µs + 0.1 µs x N	0.9 µs	8.4 µs + 0.1 µs x N	0.9 µs
SHFL	V: Data Reg. (N bits)	44 µs + 14.6 µs x N	10.4 µs	33 µs + 6 µs x N	8.4 µs	12.1 µs + 0.1 µs x N	0.9 µs	12.1 µs + 0.1 µs x N	0.9 µs
	V: Bit Reg (N bits)	243 µs + 14.6 µs x N	8.4 µs	107 µs + 6 µs x N	8.4 µs	12.1 µs + 0.1 x N	0.9 µs	12.1 µs + 0.1 µs x N	0.9 µs
	K: Constant (N bits)	34 µs + 14.6 µs x N	8.4 µs	33 µs + 6 µs x N	8.4 µs	8.4 µs + 0.1 µs x N	0.9 µs	8.4 µs + 0.1 µs x N	0.9 µs
ROTR	V: Data Reg. (N bits)	-	-	-	-	-	-	16.4 µs	1.0 µs
	V: Bit Reg (N bits)	-	-	-	-	-	-	16.4 µs	1.0 µs
	K: Constant (N bits)	-	-	-	-	-	-	12.9 µs	1.0 µs
ROTL	V: Data Reg. (N bits)	-	-	-	-	-	-	16.4 µs	1.0 µs
	V: Bit Reg (N bits)	-	-	-	-	-	-	16.4 µs	1.0 µs
	K: Constant (N bits)	-	-	-	-	-	-	12.7 µs	1.0 µs
ENCO	None	62 µs	7.2 µs	98 µs	8.4 µs	33.9 µs	0.9 µs	33.9 µs	0.9 µs
DECO	None	34 µs	7.2 µs	28 µs	8.4 µs	5.7 µs	1.0 µs	5.7 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Number Conversion Instructions

Number Conversion Instructions (Accumulator)		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
BIN	None	359 µs	7.2 µs	267 µs	8.4 µs	100.2 µs	0.9 µs	100.2 µs	0.9 µs
BCD	None	403 µs	7.2 µs	383 µs	8.4 µs	95.2 µs	0.9 µs	95.2 µs	0.9 µs
INV	None	27 µs	5.0 µs	12 µs	8.4 µs	2.5 µs	1.0 µs	2.5 µs	1.0 µs
BCDPL	None	296 µs	7.2 µs	69 µs	8.4 µs	75.6 µs	1.0 µs	75.6 µs	1.0 µs
ATH	V	-	-	-	-	-	-	25.4 µs	1.0 µs
HTA	V	-	-	-	-	-	-	25.4 µs	1.0 µs
GRAY	None	-	-	227 µs	9.0 µs	110.8 µs	1.0 µs	110.8 µs	1.0 µs
SFLDGT	None	-	-	258 µs	9.0 µs	23.1 µs	1.0 µs	23.1 µs	1.0 µs
BTOR	None	-	-	-	-	18.6 µs	1.0 µs	18.6 µs	1.0 µs
RTOB	None	-	-	-	-	8.6 µs	1.0 µs	8.6 µs	1.0 µs
RADR	None	-	-	-	-	-	-	51.4 µs	1.0 µs
DEGR	None	-	-	-	-	-	-	81.5 µs	1.0 µs

Table Instructions

Table Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
FILL	V: Data Reg.	-	-	-	-	-	-	29.4 µs + 8.0 µs x N	1.0 µs
	V: Bit Reg.	-	-	-	-	-	-		-
	K: Constant	-	-	-	-	-	-	26.2 µs+ 8.0 µs x N	1.0 µs
	P: Indir. (Data)	-	-	-	-	-	-	55.1 µs + 8.0 µs x N	1.0 µs
	P: Indir. (Bit)	-	-	-	-	-	-		-
FIND	V: Data Reg. (N bits)	-	-	-	-	-	-	66.8 µs	1.0 µs
	V: Bit Reg. (N bits)	-	-	-	-	-	-	66.8 µs	1.0 µs
	K: Constant (N bits)	-	-	-	-	-	-	64.0 µs	1.0 µs
FDGT	V: Data Reg. (N bits)	-	-	-	-	-	-	66.1 µs	1.0 µs
	V: Bit Reg. (N bits)	-	-	-	-	-	-	66.1 µs	1.0 µs
	K: Constant (N bits)	-	-	-	-	-	-	55.2 µs	1.0 µs
FINDB	V: Data Reg. (N bits)	-	-	-	-	-	-	210.8 µs	1.0 µs
	V: Bit Reg. (N bits)	-	-	-	-	-	-	210.8 µs	1.0 µs
	P: Indir. (Data)	-	-	-	-	-	-	237.0 µs	1.0 µs
	P: Indir. (Bit)	-	-	-	-	-	-	237.0 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table Instructions (cont'd)

Table Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
MOV	Move V: data reg. to V: data reg	450 µs + 17 µs x N	6.2 µs	586 µs + 8 µs x N	8.4 µs	60.2 µs + 9.5 µs x N	0.9 µs	60.2 µs + 9.5 µs x N	0.9 µs
	Move V: bit reg. to V: data reg.	430 µs + 244 µs x N	6.2 µs	629 µs + 114.7 µs x N	8.4 µs				
	Move V: data reg to V: bit reg.	460 µs + 215 µs x N	6.2 µs	569 µs + 94.4 µs x N	8.4 µs				
	Move V: bit reg. to V:bit reg. N = #of words	490 µs + 448 µs x N	6.2 µs	693 µs + 198 µs x N	8.4 µs				
TTD	V: Data Reg.	-	-	-	-	-	-	66.9 µs	1.0 µs
	V: Bit Reg.							66.9 µs	1.0 µs
RFB	V: Data Reg.	-	-	-	-	-	-	66.8 µs	1.0 µs
	V: Bit Reg.							66.8 µs	1.0 µs
STT	V: Data Reg.	-	-	-	-	-	-	67.8 µs	1.0 µs
	V: Bit Reg.							67.8 µs	1.0 µs
	K: Constant							65.0 µs	1.0 µs
RFT	V: Data Reg.	-	-	-	-	-	-	51.1 µs	1.0 µs
	V: Bit Reg.							51.1 µs	1.0 µs
ATT	V: Data Reg.	-	-	-	-	-	-	53.5 µs	1.0 µs
	V: Bit Reg.							53.5 µs	1.0 µs
	K: Constant							50.8 µs	1.0 µs
TSHFL	V: Data Reg.	-	-	-	-	-	-	134.0 µs	1.0 µs
	V: Bit Reg.							134.0 µs	1.0 µs
TSHFR	V: Data Reg.	-	-	-	-	-	-	133.9 µs	1.0 µs
	V: Bit Reg.							133.9 µs	1.0 µs
ANDMOV	V: Data Reg.	-	-	-	-	-	-	80.2 µs	1.0 µs
	V: Bit Reg.							80.2 µs	1.0 µs
ORMOV	V: Data Reg.	-	-	-	-	-	-	80.4 µs	1.0 µs
	V: Bit Reg.							80.4 µs	1.0 µs
XORMOV	V: Data Reg.	-	-	-	-	-	-	80.4 µs	1.0 µs
	V: Bit Reg.							80.4 µs	1.0 µs
SWAP	V: Data Reg.	-	-	-	-	-	-	84.1 µs	1.0 µs
	V: Bit Reg.							84.1 µs	1.0 µs
SETBIT	V: Data Reg. (N bits)	-	-	-	-	-	-	59.5 µs	1.0 µs
	V: Bit Reg. (N bits)							59.5 µs	1.0 µs
RSTBIT	V: Data Reg. (N bits)	-	-	-	-	-	-	59.5 µs	1.0 µs
	V: Bit Reg. (N bits)							59.5 µs	1.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table Instructions (cont'd)

Table Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
MOVMC	Move V: Data Reg. to EEPROM	-	-	356 µs + 7689 µs x N	8.4 µs	-	-	33.5 µs + 10.4 µs x N	0.9 µs
	Move V: Bit Reg. to EEPROM	-	-	392 µs + 7843 µs x N	8.4 µs	-	-	33.5 µs + 10.4 µs x N	0.9 µs
	Move from EEPROM to V: Data Reg.	250 µs + 201 µs x N	6.2 µs	520 µs + 181 µs x N	8.4 µs	50 µs + 15 µs x N	1.2 µs	33.5 µs + 10.4 µs x N	0.9 µs
	Move from EEPROM to V: Bit Reg. N=#of words	-	-	565 µs + 344 µs x N	8.4 µs	50 µs + 15 µs x N	1.2 µs	33.5 µs + 10.4 µs x N	0.9 µs
LDLBL	K	58 µs	8.4 µs	56 µs	8.4 µs	7.4 µs	1.5 µs	6.4 µs	1.3 µs

CPU Control Instructions

CPU Control Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
NOP	None	0 µs	0 µs	0 µs	0 µs	0.5 µs	0.5 µs	0.5 µs	0.5 µs
END	None	27 µs	27 µs	16 µs	16 µs	12.8 µs	0 µs	12.8 µs	0 µs
STOP	None	16 µs	5 µs	15 µs	7.4 µs	0 µs	0.9 µs	0 µs	0.9 µs
RSTWT	None	-	-	19 µs	8.4 µs	4.7 µs	0.9 µs	4.7 µs	0.9 µs

Program Control Instructions

Program Control Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
GOTO	K	-	-	14 µs	8.4 µs	5.1 µs	4.8 µs	5.1 µs	4.8 µs
LBL	K	-	-	0.6 µs	0.6 µs	5.7 µs	0.0 µs	5.7 µs	0.0 µs
FOR	V, K	-	-	32 µs	16.4 µs	85.8 µs	5.8 µs	85.8 µs	5.8 µs
NEXT	None	-	-	19 µs	0 µs	10.2 µs	0.0 µs	10.2 µs	0.0 µs
GTS	K	-	-	37 µs	11.4 µs	10.9 µs	5.5 µs	10.9 µs	5.5 µs
SBR	K	-	-	0.6 µs	0 µs	0.5 µs	0.0 µs	0.5 µs	0.0 µs
RT	None	-	-	35 µs	0 µs	9.9 µs	0.0 µs	9.9 µs	0.0 µs
RTC	None	-	-	-	-	-	-	11.4 µs	5.9 µs
MLS	K (1-7)	12 µs	12 µs	11.5 µs	11.5 µs	3.7 µs	3.7 µs	3.7 µs	3.7 µs
MLR	K (0-7) N=1 to 7	13 µs + 2.4 µs x N	13 µs + 2.4 µs x N	12.7 µs + 2.3 µs x N	12.7 µs + 2.3 µs x n	3.5 µs	3.5 µs	3.5 µs	3.5 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C

Interrupt Instructions

Interrupt Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ENI	None	9 µs	5 µs	10.5 µs	8.4 µs	5.0 µs	1.0 µs	5.0 µs	1.0 µs
DISI	None	8 µs	5 µs	11 µs	8.4 µs	5.7 µs	0.9 µs	5.7 µs	0.9 µs
INT	O	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs
IRT	None	1.6 µs	0 µs	8 µs	0 µs	1.3 µs	0 µs	1.3 µs	0 µs
IRTC	None	-	-	-	-	-	-	0.5 µs	0 µs

Network Instructions

Network Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
RX	X, Y, C, T, CT, SP, S	-	-	TBD	TBD	251.3 µs	1.1 µs	251.3 µs	1.1 µs
	V: Data Reg.					251.3 µs	1.1 µs	251.3 µs	1.1 µs
	V: Bit Reg.					251.3 µs	1.1 µs	251.3 µs	1.1 µs
	P: Indir. (Data)					270.3 µs	1.9 µs	270.3 µs	1.9 µs
	P: Indir. (Bit)					270.3 µs	1.9 µs	270.3 µs	1.9 µs
WX	X, Y, C, T, CT, SP, S	-	-	TBD	TBD	252.0 µs	2.7 µs	252.0 µs	2.7 µs
	V: Data Reg.					252.0 µs	2.7 µs	252.0 µs	2.7 µs
	V: Bit Reg.					252.0 µs	2.7 µs	252.0 µs	2.7 µs
	P: Indir. (Data)					271.3 µs	3.4 µs	271.3 µs	3.4 µs
	P: Indir. (Bit)					271.3 µs	3.4 µs	271.3 µs	3.4 µs

Intelligent I/O Instructions

Intelligent I/O Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
RD	V: Data Reg.	TBD	TBD	TBD	TBD	385.7 µs	1.2 µs	385.7 µs	1.2 µs
	V: Bit Reg.					385.7 µs	1.2 µs	385.7 µs	1.2 µs
WT	V: Data Reg.	TBD	TBD	TBD	TBD	385.6 µs	1.2 µs	385.6 µs	1.2 µs
	V: Bit Reg.					385.6 µs	1.2 µs	385.6 µs	1.2 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Message Instructions

Message Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
FAULT	V: Data Reg.	171 µs	8.4 µs	23176 µs	8.4 µs	84.9 µs	1.1 µs	84.9 µs	1.1 µs
	V: Bit Reg.	253 µs	8.4 µs	23206 µs	8.4 µs	84.9 µs	1.1 µs	84.9 µs	1.1 µs
	K: Constant	2798 µs	8.4 µs	29108 µs	8.4 µs	80.8 µs	1.2 µs	80.8 µs	1.2 µs
DLBL	K	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs
NCON	K	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs
ACON	K	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs	0 µs
PRINT	Text Data	-	-	-	-	36.3 µs	1.1 µs	36.3 µs	1.1 µs

RLL*PLUS* Instructions

RLL*PLUS* Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
ISG	S	31 µs	32 µs	28 µs	27 µs	20.9 µs	9.2 µs	20.9 µs	9.2 µs
SG	S	31 µs	32 µs	28 µs	27 µs	20.9 µs	9.2 µs	20.9 µs	9.2 µs
JMP	S	14 µs	8 µs	14.3 µs	8.4 µs	20.9 µs	3.7 µs	20.9 µs	3.7 µs
NJMP	S	14 µs	8 µs	13.3 µs	8.4 µs	21.0 µs	4.0 µs	21.0 µs	4.0 µs
CV	S	43 µs	27 µs	20 µs	20 µs	12.1 µs	12.1 µs	12.1 µs	12.1 µs
CVJMP	S (N stages, 1 to 16)	33 µs + 14.5 µs x N	23 µs	22.9 µs + 6.1 µs x N	10 µs	11.0 µs	11.0 µs	11.0 µs	11.0 µs
BCALL	C	18 µs	17 µs	17 µs	18 µs	22.1 µs	22.6 µs	22.1 µs	22.6 µs
BLK	C	32 µs	30 µs	17 µs	13 µs	17.1 µs	14.6 µs	17.1 µs	14.6 µs
BEND	None	17 µs	17 µs	9 µs	9 µs	8.7 µs	0.0 µs	8.7 µs	0.0 µs

DRUM Instructions

Drum Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
DRUM	CT	-	-	-	-	265.2 µs	48.8 µs	265.2 µs	48.8 µs
EDRUM	CT	-	-	-	-	189.5 µs	78.0 µs	189.5 µs	78.0 µs
MDRMD	CT	-	-	-	-	411.3 µs	216.4 µs	411.3 µs	216.4 µs
MDRMW	CT	-	-	-	-	378.6 µs	147.0 µs	378.6 µs	147.0 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C

Clock / Calender Instructions

Clock / Calender Instructions		DL230		DL240		DL250-1		DL260	
Instruction		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
DATE	V: Data Reg.	-	-	-	-	24.0 µs	1.2 µs	24.0 µs	1.2 µs
	V: Bit Reg.								
TIME	V: Data Reg.	-	-	-	-	50.8 µs	1.2 µs	50.8 µs	1.2 µs
	V: Bit Reg.								

Modbus Instructions

Modbus Instructions		DL230		DL240		DL250-1		DL260	
Instruction		Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
MRX	Input, Input Register Coil, Holding Register	-	-	-	-	-	-	120.2 µs	1.3 µs
MWX	Input, Input Register Coil, Holding Register	-	-	-	-	-	-	21.3 µs	1.3 µs

ASCII Instructions

ASCII Instructions		DL230		DL240		DL250-1		DL260	
Instruction	Legal Data Types	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute	Execute	Not Execute
AIN	V	-	-	-	-	-	-	13.9 µs	12.0 µs
AFIND	V	-	-	-	-	-	-	111.5 µs	1.3 µs
AEX	V	-	-	-	-	-	-	111.7 µs	1.3 µs
CMPV	V	-	-	-	-	-	-	12.2 µs	1.3 µs
SWAPB	V	-	-	-	-	-	-	109.8 µs	1.3 µs
VPRINT	Text Data	-	-	-	-	-	-	161.6 µs	1.3 µs
PRINTV	V	-	-	-	-	-	-	163.3 µs	1.3 µs
ACRB	V	-	-	-	-	-	-	3.9 µs	1.1 µs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Special Relays



In This Appendix...

DL230 CPU Special Relays D-2
DL240/DL250-1/DL260 CPU Special Relays D-5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.
SP2	Always OFF	Provides a contact that is always off
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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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	On 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.
SP74	Underflow	On anytime a math operation results in an underflow error.
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	XO is on	XO - On when corresponding input is on.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Equal Relays for Multi-step Presets with Up/Down Counter #1 / DL230 (for use with a Counter Interface Module)

SP540	Current = target value	On when the counter current value equals the value in V3630.
SP541	Current = target value	On when the counter current value equals the value in V3632.
SP542	Current = target value	On when the counter current value equals the value in V3634.
SP543	Current = target value	On when the counter current value equals the value in V3636.
SP544	Current = target value	On when the counter current value equals the value in V3640.
SP545	Current = target value	On when the counter current value equals the value in V3642.
SP546	Current = target value	On when the counter current value equals the value in V3644.
SP547	Current = target value	On when the counter current value equals the value in V3646.
SP550	Current = target value	On when the counter current value equals the value in V3650.
SP551	Current = target value	On when the counter current value equals the value in V3652.
SP552	Current = target value	On when the counter current value equals the value in V3654.
SP553	Current = target value	On when the counter current value equals the value in V3656.
SP554	Current = target value	On when the counter current value equals the value in V3660.
SP555	Current = target value	On when the counter current value equals the value in V3662.
SP556	Current = target value	On when the counter current value equals the value in V3664.
SP557	Current = target value	On when the counter current value equals the value in V3666.
SP560	Current = target value	On when the counter current value equals the value in V3670.
SP561	Current = target value	On when the counter current value equals the value in V3672.
SP562	Current = target value	On when the counter current value equals the value in V3674.
SP563	Current = target value	On when the counter current value equals the value in V3676.
SP564	Current = target value	On when the counter current value equals the value in V3700.
SP565	Current = target value	On when the counter current value equals the value in V3702.
SP566	Current = target value	On when the counter current value equals the value in V3704.
SP567	Current = target value	On when the counter current value equals the value in V3706.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL240/DL250-1/DL260 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.
SP2	Always OFF	Provides a contact that is always off.
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.
SP14	Break Relay 1 (DL250-1/260)	On when the BREAK instruction is executed. It is OFF when the CPU is in any other 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.
SP17	Forced stop mode relay (DL250-1/260)	On anytime the CPU keyswitch is in the STOP position.
SP20	Forced stop mode	On when the STOP instruction is executed.
SP21	Break Relay 2 (DL250-1/260)	On when the BREAK instruction is executed. It is OFF when the CPU mode is changed to RUN.
SP22	Interrupt enabled	On when interrupts have been enabled using the ENI instruction.
SP25	CPU battery disabled relay (DL250-1/260)	On when the CPU battery is disabled by special V-memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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/dead	On when the CPU battery voltage is low or dead. Note: The CPU must have a battery installed.
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 occured on any of the CPU ports.
SP47	I/O configuration error	On if an I/O configuration error has occured. 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.
SP54	Intelligent I/O error	On when communications with an intelligent module has occurred.
SP56	Table instruction overrun	On if a table instruction with a pointer is executed and the pointer value is outside the table boundary

Accumulator Status Relays

SP53	Math/Table pointer error	On if there is math execution error or a table pointer error.
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	On 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.
SP72	Floating Point	On when the numerical value in the accumulator is a floating point number.
SP73	Overflow	On if overflow occurs in the accumulator when a signed addition or subtraction results in an incorrect sign bit.
SP74	Under flow	On when a floating point math operation results in an underflow error.
SP75	Data error	On if data is not a numerical value.
SP76	Load zero	On when any instruction loads a value of zero into the accumulator.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Communications Monitoring Relays

SP116	DL240 CPU communication	On when the CPU is communicating with another device
SP116	DL250-1/260 communication	On when Port 2 is communicating with another device
SP117	Comm error Port 2 (DL250-1/260)	On when Port 2 has encountered a communication error.
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	Comm 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	Comm 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	Comm 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	Comm 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	Comm 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	Comm 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	Comm 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	Comm error Slot 7	On when the communication module in slot 7 of the local base has encountered a communication error.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 V3630.
SP541	Current = target value	On when the counter current value equals the value in V3632.
SP542	Current = target value	On when the counter current value equals the value in V3634.
SP543	Current = target value	On when the counter current value equals the value in V3636.
SP544	Current = target value	On when the counter current value equals the value in V3640.
SP545	Current = target value	On when the counter current value equals the value in V3642.
SP546	Current = target value	On when the counter current value equals the value in V3644.
SP547	Current = target value	On when the counter current value equals the value in V3646.
SP550	Current = target value	On when the counter current value equals the value in V3650.
SP551	Current = target value	On when the counter current value equals the value in V3652.
SP552	Current = target value	On when the counter current value equals the value in V3654.
SP553	Current = target value	On when the counter current value equals the value in V3656.
SP554	Current = target value	On when the counter current value equals the value in V3660.
SP555	Current = target value	On when the counter current value equals the value in V3662.
SP556	Current = target value	On when the counter current value equals the value in V3664.
SP557	Current = target value	On when the counter current value equals the value in V3666.
SP560	Current = target value	On when the counter current value equals the value in V3670.
SP561	Current = target value	On when the counter current value equals the value in V3672.
SP562	Current = target value	On when the counter current value equals the value in V3674.
SP563	Current = target value	On when the counter current value equals the value in V3676.
SP564	Current = target value	On when the counter current value equals the value in V3700.
SP565	Current = target value	On when the counter current value equals the value in V3702.
SP566	Current = target value	On when the counter current value equals the value in V3704.
SP567	Current = target value	On when the counter current value equals the value in V3706.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 V3710
SP571	Current = target value	On when the counter current value equals the value in V3712
SP572	Current = target value	On when the counter current value equals the value in V3714
SP573	Current = target value	On when the counter current value equals the value in V3716
SP574	Current = target value	On when the counter current value equals the value in V3720
SP575	Current = target value	On when the counter current value equals the value in V3722
SP576	Current = target value	On when the counter current value equals the value in V3724
SP577	Current = target value	On when the counter current value equals the value in V3726
SP600	Current = target value	On when the counter current value equals the value in V3730
SP601	Current = target value	On when the counter current value equals the value in V3732
SP602	Current = target value	On when the counter current value equals the value in V3734
SP603	Current = target value	On when the counter current value equals the value in V3736
SP604	Current = target value	On when the counter current value equals the value in V3740
SP605	Current = target value	On when the counter current value equals the value in V3742
SP606	Current = target value	On when the counter current value equals the value in V3744
SP607	Current = target value	On when the counter current value equals the value in V3746
SP610	Current = target value	On when the counter current value equals the value in V3750
SP611	Current = target value	On when the counter current value equals the value in V3752
SP612	Current = target value	On when the counter current value equals the value in V3754
SP613	Current = target value	On when the counter current value equals the value in V3756
SP614	Current = target value	On when the counter current value equals the value in V3760
SP615	Current = target value	On when the counter current value equals the value in V3762
SP616	Current = target value	On when the counter current value equals the value in V3764
SP617	Current = target value	On when the counter current value equals the value in V3766

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PLC Memory



In This Appendix...

DL205 PLC Memory E-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 PLC Memory

When designing a PLC application, it is important for the PLC user to understand the different types of memory in the PLC. The DL205 CPUs use two types of memory: RAM and EEPROM. RAM is Random Access Memory and EEPROM is Electrically Erasable Programmable Read Only Memory. The PLC program is stored in EEPROM, and the PLC V-memory data is stored in RAM. There is also a small range of V-memory that can be copied to EEPROM which will be explained later.

The V-memory in RAM can be configured as either retentive or non-retentive.

Retentive memory is memory that is configured by the user to maintain values through a power cycle or a PROGRAM to RUN transition. Non-retentive memory is memory that is configured by the PLC user to clear data after a power cycle or a PROGRAM to RUN transition. The retentive ranges can be configured with either the handheld programmer using AUX57 or ***Direct***SOFT (PLC Setup).

The contents of RAM memory can be written to and read from an infinite number of times, but RAM requires a power source to maintain the contents of memory. The contents of RAM are maintained by the internal power supply (5VDC) only while the PLC is powered by an external source, normally 120VAC. When power to the PLC is turned off, the contents of RAM are maintained by a"Super-Capacitor." If the Super-Capacitor ever discharges, the contents of RAM will be lost. The data retention time of the Super-Capacitor backed RAM is 3 weeks maximum, and 4 1/2 days minimum (at 60° C). An optional battery, D2-BAT, can be added to maintain RAM retentive memory if the DL230 or DL240 is ever without external power (see Volume I, page 3-14 for a detailed explanation).

The contents of EEPROM memory can be read from an infinite number of times, but there is a limit to the number of times it can be written to (typical specification is 100,000 writes). EEPROM does not require a power source to maintain the memory contents. It will retain the contents of memory indefinitely.

PLC user V-memory is stored in both volatile RAM and non-volatile EEPROM memory. The table below shows the memory areas for each DL205 CPU.

DL205 Memory Area				
PLC Type	**DL230**	**DL240**	**DL250-1**	**DL260**
Volatile RAM	V2000 - V2377	V2000 - V3777	V1400 - V7377 V10000 - V17777	V400 - V777 V1400 - V7377 V10000 - V35777
Non-volatile	V4000 - V4177	V4000 - V4377	-	-

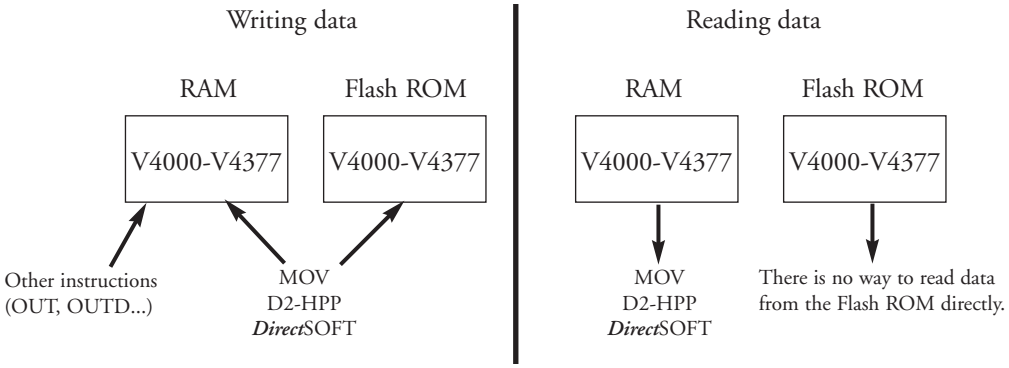
Data values that must be retained for long periods of time, when the PLC is powered off, should be stored in EEPROM based V-memory. Since EEPROM is limited to the number of times it can be written to, it is suggested that transitional logic, such as a one-shot, be used to write the data one time, instead on each CPU scan.

Data values that are continually changing or which can be initialized with program logic should be stored in RAM based V-memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

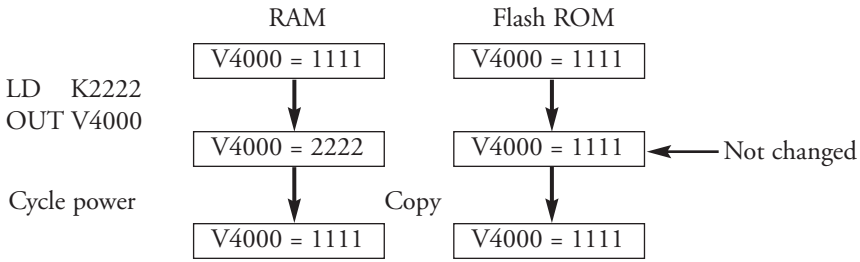
Non-volatile V-memory in the DL205

There are two types of memory assigned for the non-volatile V-memory area. They are RAM and flash ROM (EEPROM). They are sharing the same V-memory addresses; however, **you can only use the MOV instruction, D2-HPP and *Direct*SOFT to write data to the flash ROM.** When you write data to the flash ROM, the same data is also written to RAM. If you use other instructions, you can only write data to RAM. When you read data from the non-volatile V-memory area, the data is always read from RAM. The following explanation uses the DL240 CPU as an example.



After a power cycle, the PLC always copies the data in the flash ROM to the RAM.

If you use the instructions except for the MOV instruction to write data into the non-volatile V-memory area, you only update the data in RAM. After a power cycle, the PLC copies the previous data from the flash memory to the RAM, so you may think the data you changed has disappeared. To avoid trouble such as this, we recommend that you use the MOV instruction.



This appears to be previous data returning.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Product Weight Table



In This Appendix...

DL205 Product Weight Table F-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL205 Product Weight Table

CPUs	Weight
D2-230	2.8 oz. (80g)
D2-240	2.8 oz. (80g)
D2-250-1	2.5 oz. (70g)
D2-260	2.5 oz. (70g)
I/O Bases	
D2-03B-1	12.3oz. (350g)
D2-03BDC1-1	11.4oz. (322g)
D2-03BDC-2	10.1oz. (285g)
D2-04B-1	13.4 oz. (381g)
D2-04BDC1-1	12.5 oz. (354g)
D2-04BDC-2	11.2 oz. (317g)
D2-06B-1	14.4 oz. (410g)
D2-06BDC1-1	13.8 oz. (392g)
D2-06BDC2-1	13.8 oz. (392g)
D2-09B-1	18.6 oz. (530g)
D2-09BDC1-1	18.3 oz. (522g)
D2-09BDC2-1	19 oz. (530g)
DC Input Modules	
D2-08ND3	2.3 oz. (65g)
D2-16ND3-2	2.3 oz. (65g)
D2-32ND3	2.1oz. (60g)
D2-32ND3-2	2.1oz. (60g)
AC Input Modules Weight	
D2-08NA-1	2.5 oz. (70g)
D2-08NA-2	2.5 oz. (70g)
D2-16NA	2.4 oz. (68g)
DC Input/Relay Output Module	
D2-08CDR	3.5 oz. (100g)

DC Output Modules	
D2-04TD1	2.8 oz. (80g)
D2-08TD1	2.3 oz. (65g)
D2-08TD2	2.1 oz. (60g)
D2-16TD1-2	2.3 oz. (65g)
D2-16TD2-2	2.8 oz. (80g)
F2-16TD1P	2.0 oz. (56g)
F2-16TD2P	2.0 oz. (56g)
D2-32TD1	2.1oz. (60g)
D2-32TD2	2.1oz. (60g)
AC Output Modules	
D2-08TA	2.8 oz. (80g)
F2-08TA	3.5 oz. (99g)
D2-12TA	2.8 oz. (80g)
Relay Output Modules	
D2-04TRS	2.8 oz. (80g)
D2-08TR	3.9 oz. (114g)
D2-12TR	4.6 oz. (130g)
F2-08TR	5.5 oz. (156g)
F2-08TRS	5.5 oz. (156g)
CPU-Slot Controllers	
H2-EBC	1.6 oz. (45g)
H2-EBC-F	2.1 oz. (60g)
F2-SDS-1	2.8 oz. (80g)
H2-PBC	2.1 oz. (60g)
F2-DEVNETS-1	3.0 oz. (86g)

Analog Modules Weight	
F2-04AD-1	3.0 oz (86g)
F2-04AD-2	3.0 oz (86g)
F2-04AD-1L	3.0 oz (86g)
F2-04AD-2L	3.0 oz (86g)
F2-08AD-1	3.0 oz (86g)
F2-08AD-2	4.2 oz (118g)
F2-02DA-1	2.8 oz. (80g)
F2-02DA-2	2.8 oz. (80g)
F2-02DA-1L	2.8 oz. (80g)
F2-02DA-2L	2.8 oz. (80g)
F2-08DA-1	2.8 oz. (80g)
F2-08DA-2	3.8 oz. (109g)
F2-02DAS-1	3.8 oz. (109g)
F2-02DAS-2	3.8 oz. (109g)
F2-4AD2DA	4.2 oz. (118g)
F2-8AD4DA-1	4.2 oz. (118g)
F2-8AD4DA-2	4.2 oz. (118g)
F2-04RTD	3.0 oz (86g)
F2-04THM	3.0 oz (86g)
Specialty Modules	
H2-CTRIO	2.3 oz. (65g)
D2-CTRINT	2.3 oz. (65g)
H2-ECOM	1.6 oz. (45g)
H2-ECOM100	1.5 oz. (43g)
H2-ECOM-F	5.5 oz. (156g)
H2-ERM	1.6 oz. (45g)
H2-ERM-F	5.5 oz. (156g)
H2-SERIO	1.5 oz. (43g)
D2-DCM	3.8 oz. (109g)
F2-CP128	3.9 oz. (111g)
D2-EM	2.3 oz. (65g)
D2-CM	1.8 oz. (50g)
F2-08SIM	2.5 oz. (70g)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ASCII Table



In This Appendix...

ASCII Conversion Table G-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ASCII Conversion Table

DECIMAL TO HEX TO ASCII CONVERTER											
DEC	HEX	ASCII	DEC	HEX	ASCII	DEC	HEX	ASCII	DEC	HEX	ASCII
0	00	NUL	32	20	space	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	“	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	‘	71	47	G	103	67	g
8	08	BS	40	28	(	72	48	H	104	68	h
9	09	TAB	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Numbering Systems



In This Appendix...

Introduction H–2
Binary Numbering System H–2
Hexadecimal Numbering System H–3
Octal Numbering System H–4
Binary Coded Decimal (BCD) Numbering System H–5
Real (Floating Point) Numbering System H–5
BCD/Binary/Decimal/Hex/Octal -What is the Difference? H–6
Data Type Mismatch H–7
Signed vs. Unsigned Integers H–8
AutomationDirect.com Products and Data Types H–9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

As almost anyone who uses a computer is somewhat aware, the actual operations of a computer are done with a binary number system. Traditionally, the two possible states for a binary system are represented by the digits "zero" (0) and "one" (1), although "off" and "on" or sometimes "no" and "yes" are closer to what is actually involved. Most of the time a typical PC user has no need to think about this aspect of computers, but every now and then one gets confronted with the underlying nature of the binary system.

A PLC user, specifically the PLC programmer, should be more aware of the binary system. This appendix will provide an explanation of the numbering systems most commonly used by a PLC.

Binary Numbering System

Computers, including PLCs, use the Base 2 numbering system, which is called Binary and often called Decimal. Much the same, a PLC relies on only two valid digits , zero and one, or off and on respectively. You would think that it would be hard to have a numbering system built on Base 2 with only two possible values, but the secret is by encoding using several digits.

Each digit in the base 2 system when referenced by a computer is called a bit. When four bits are grouped together, they form what is known as a nibble. Eight bits or two nibbles would be a byte. Sixteen bits or two bytes would be a word (Table 1). Thirty-two bits or two words is a double word.

Word															
Byte								Byte							
Nibble				Nibble				Nibble				Nibble			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 1

Binary is not “natural” for us to use since we grow up using the base 10 system. Base 10 uses the numbers 0-9, as we are all well aware. From now on, the different bases will be shown as a subscripted number following the number. Example; 10 decimal would be 10_{10}.

Table 2 shows how base 2 numbers relate to their decimal equivalents.

A nibble of 1001_2 would be equal to a decimal number 9 ($1*2^3 + 1*2^0$ or $8_{10} + 1_{10}$). A byte of 11010101_2 would be equal to 213 ($1*2^7 + 1*2^6 + 1*2^4 + 1*2^2 + 1*2^0$ or $128_{10} + 64_{10} + 16_{10} + 4_{10} + 1_{10}$).

Binary/Decimal Bit Pattern																
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Power	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Bit Value	32678	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Max Value	65535_{10}															

Table 2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hexadecimal Numbering System

The binary numbering system can be difficult and cumbersome to interpret for some users. Therefore, the hexadecimal numbering system was developed as a convenience for humans since the PLC (computer) only understands pure binary. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits. It is easier for us to read hexadecimal numbers than binary numbers.

The hexadecimal numbering system uses 16 characters (base 16) to represent values. The first ten characters are the same as our decimal system, 0-9, and the first six letters of the alphabet, A-F. Table 3 lists the first eighteen decimal numbers; 0-17 in the left column and the equivalent hexadecimal numbers are shown in the right column.

Decimal	Hex	Decimal	Hex
0	0	9	9
1	1	10	A
2	2	11	B
3	3	12	C
4	4	13	D
5	5	14	E
6	6	15	F
7	7	16	10
8	8	17	11

Table 3

Note that “10” and “11” in hex are not the same as “10” and “11” in decimal. Only the first ten numbers 0-9 are the same in the two representations. For example, consider the hex number “D8AF”. To evaluate this hex number use the same method used to write decimal numbers. Each digit in a decimal number represents a multiple of a power of ten (base 10). Powers of ten increase from right to left. For example, the decimal number 365 means $3x10^2$ + 6x10 + 5. In hex each digit represents a multiple of a power of sixteen (base 16). Therefore, the hex number D8AF translated to decimal means $13x16^3 + 8x16^2 + 10x16 + 15 = 55471$. However, going through the arithmetic for hex numbers in order to evaluate them is not really necessary. The easier way is to use the calculator that comes as an accessory in Windows. It can convert between decimal and hex when in “Scientific” view.

Note that a hex number such as “365” is not the same as the decimal number “365”. Its actual value in decimal terms is $3x16^2 + 6x16 + 5 = 869$. To avoid confusion, hex numbers are often labeled or tagged so that their meaning is clear. One method of tagging hex numbers is to append a lower case “h” at the end. Another method of labeling is to precede the number with 0x. Thus, the hex number “D8AF” can also be written “D8AFh”, where the lower case “h” at the end is just a label to make sure we know that it is a hex number. Also, D8AF can be written with a labeling prefix as “0xD8AF”.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Octal Numbering System

Many of the early computers used the octal numbering system for compiled printouts. Today, the PLC is about the only device that uses the Octal numbering system. The octal numbering system uses 8 values to represent numbers. The values are 0-7 being Base 8. Table 4 shows the first 31 decimal digits in octal. Note that the octal values are 0-7, 10-17, 20-27, and 30-37.

Octal	Decimal	Octal	Decimal
0	0	20	16
1	1	21	17
2	2	22	18
3	3	23	19
4	4	24	20
5	5	25	21
6	6	26	22
7	7	27	23
10	8	30	24
11	9	31	25
12	10	32	26
13	11	33	27
14	12	34	28
15	13	35	29
16	14	36	30
17	15	37	31

Table 4

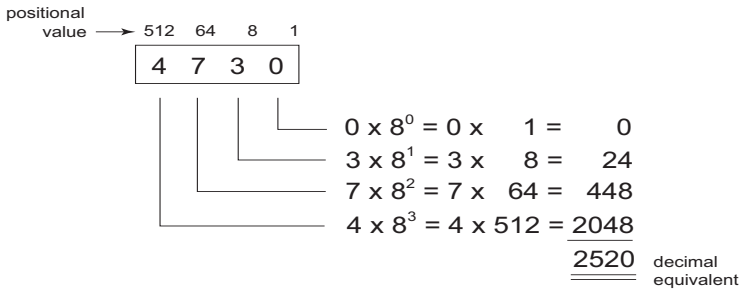
This follows the *Direct*LOGIC PLCs. Refer to Chapter 3 bit maps and notice that the memory addresses are numbered in octal, as well as each bit. The octal system is much like counting in the decimal system without the digits 8 and 9 being available.

The general format for four digits of the octal number is:

$$(d \times 8^0) + (d \times 8^1) + (d \times 8^2) + (d \times 8^3)$$

where "d" means digit. This is the same format used in the binary, decimal, or hexadecimal systems except that the base number for octal is 8.

Using the powers of expansion, the example below shows octal 4730 converted to decimal.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Binary Coded Decimal (BCD) Numbering System

BCD is a numbering system where four bits are used to represent each decimal digit. The binary codes corresponding to the hexadecimal digits A-F are not used in the BCD system. For this reason numbers cannot be coded as efficiently using the BCD system. For example, a byte can represent a maximum of 256 different numbers (i.e. 0-255) using normal binary, whereas only 100 distinct numbers (i.e. 0-99) could be coded using BCD. Also, note that BCD is a subset of hexadecimal and neither one does negative numbers.

	BCD Bit Pattern															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Power	10^3				10^2				10^1				10^0			
Bit Value	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
Max Value	9				9				9				9			

Table 5

One plus for BCD is that it reads like a decimal number, whereas 867 in BCD would mean 867 decimal. No conversion is needed; however, within the PLC, BCD calculations can be performed if numbers are adjusted to BCD after normal binary arithmetic.

Real (Floating Point) Numbering System

The terms Real and floating-point both describe IEEE-754 floating point arithmetic. This standard specifies how single precision (32-bit) and double precision (64-bit) floating point numbers are to be represented as well as how arithmetic should be carried out on them. Most PLCs use the 32-bit format for floating point (or Real) numbers which will be discussed here.

	Real (Floating Point 32) Bit Pattern															
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Sign	Exponent								Mantissa						
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Mantissa (continues from above)															

Table 6

Floating point numbers which *Direct*LOGIC PLCs use have three basic components: sign, exponent and mantissa. The 32-bit word required for the IEEE standard floating point numbers is shown in Table 6. It is represented as a number from 0 to 31, left to right. The first bit (31) is the sign bit, the next eight bits (30-23) are the exponent bits and the final 23 bits (22-0) are the fraction bits.

In summary:

The sign bit is either "0" for positive or "1" for negative;

The exponent uses base 2;

The first bit of the mantissa is typically assumed to be "1.*fff*", where "*f*" is the field of fraction bits.

The Internet can provide a more in-depth explanation of the floating point numbering system. One website to look at is:

http://www.psc.edu/general/software/packages/ieee/ieee.html

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCD/Binary/Decimal/Hex/Octal - What is the Difference?

Sometimes there is confusion about the differences between the data types used in a PLC. The PLC's native data format is BCD, while the I/O numbering system is octal. Other numbering formats used are binary and Real. Although data is stored in the same manner (0's and 1's), there are differences in the way that the PLC interprets it.

While all of the formats rely on the base 2 numbering system and bit-coded data, the format of the data is dissimilar. Table 7 below shows the bit patterns and values for various formats.

Binary/Decimal Bit Pattern																
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Decimal Bit Value	32678	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Max Value	65535															
	Hexadecimal Bit Pattern															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Decimal Bit Value	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
Max Value	F				F				F				F			
	BCD Bit Pattern															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Decimal Bit Value	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
Max Value	9				9				9				9			
	Octal Bit Pattern															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Value	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1
Max Value	1	7			7			7			7			7		
	Real (Floating Point 32) Bit Pattern															
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Sign	Exponent								Mantissa						
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Mantissa (continued from above)															

Table 7

As seen in Table 7, the BCD and hexadecimal formats are similar, although the maximum number for each grouping is different (9 for BCD and F for hexadecimal). This allows both formats to use the same display method. The unfortunate side effect is that unless the data type is documented, it's difficult to know what the data type is unless it contains the letters A-F.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Data Type Mismatch

Data type mismatching is a common problem when using an operator interface. Diagnosing it can be a challenge until you identify the symptoms. Since the PLC uses BCD as the native format, many people tend to think it is interchangeable with binary (unsigned integer) format. This is true to some extent, but not in this case. Table 8 shows how BCD and binary numbers differ.

Data Type Mismatch												
Decimal	0	1	2	3	4	5	6	7	8	9	10	11
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	0001 0000	0001 0001
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	0000 1010	0000 1011

Table 8

As the table shows, BCD and binary share the same bit pattern up until you get to the decimal number 10. Once you get past 10, the bit pattern changes. The BCD bit pattern for the decimal 10 is actually equal to a value of 16 in binary, causing the number to jump six digits when viewing it as BCD. With larger numbers, the error multiplies. Binary values from 10 to 15 Decimal are actually invalid for the BCD data type.

Looking at a larger number, such as the value shown in Table 9, both the BCD bit pattern and the decimal bit pattern correspond to a base 10 value of 4095_{10}. If bit patterns are read, or interpreted, in a different format than what is used to write them, the data will not be correct. For instance, if the BCD bit pattern is interpreted as a decimal (binary) bit pattern, the result is a base 10 value of 16533_{10}. Similarly, if you try to view the decimal (binary) bit pattern as a BCD value, it is not a valid BCD value at all, but could be represented in hexadecimal as 0xFFF.

Base 10 Value	BCD Bit Pattern	Binary Bit Pattern
4095	0100 0000 1001 0101	1111 1111 1111

Table 9

Look at the following example and note the same value represented by the different numbering systems.

0100	0011	Binary	0001 0010 0011 0100		Binary
6	7	Decimal	4	6 6 0	Decimal
4	3	Hex	1	2 3 4	Hex
0110	0111	BCD	0100 0110 0110 0000		BCD
1 0	3	Octal	1 1 0 6 4		Octal

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Signed vs. Unsigned Integers

So far, we have dealt with unsigned data types only. Now we will deal with signed data types (negative numbers). The BCD and hexadecimal numbering systems do not use signed data types.

In order to signify that a number is negative or positive, we must assign a bit to it. Usually, this is the Most Significant Bit (MSB) as shown in Table 10. For a 16-bit number, this is bit 15. This means that for 16-bit numbers we have a range of -32767 to 32767.

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSB															LSB

Table 10

There are two ways to encode a negative number: two's complement and Magnitude Plus sign. The two methods are not compatible.

The simplest method to represent a negative number is to use one bit of the PLC word as the sign of a number while the remainder of the word gives its magnitude. It is general convention to use the most significant bit (MSD) as the sign bit: a 1 will indicate a negative, and a 0 a positive number. Thus, a 16-bit word allows numbers in the range ±32767. Table 11 shows a representation of 100 and a representation of -100 in this format.

Magnitude Plus Sign	
Decimal	**Binary**
100	0000 0000 0110 0100
-100	1000 0000 0110 0100

Table 11

Two's complement is a bit more complicated. A simple formula for two's complement is to invert the binary and add one (see Table 12). Basically, 1's are being changed to 0's and all 0's are being changed to 1.

Two's Complement	
Decimal	**Binary**
100	0000 0000 0110 0100
-100	1111 1111 1001 1011

Table 12

More information about 2's complement can be found on the Internet at the following websites:

http://www.evergreen.edu/biophysics/technotes/program/2s_comp.htm

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AutomationDirect.com Products and Data Types

*Direct*LOGIC PLCs

The *Direct*LOGIC PLC family uses the octal numbering system for all addressing which includes: inputs, outputs, internal V-memory locations, timers, counters, internal control relays (bits), etc. Most data in the PLC, including timer and counter current values, is in BCD format by default. User data in V-memory locations may be stored in other data types if it is changed by the programmer, or comes from some external source, such as an operator interface. Any manipulation of data must use instructions appropriate for that data type which includes: Load instructions, Math instructions, Out box instructions, comparison instructions, etc. In many cases, the data can be changed from one data type to another, but be aware of the limitations of the various data types when doing so. For example, to change a value from BCD to decimal (binary), use a BIN instruction box. To change from BCD to a real number, use a BIN and a BTOR instruction box. When using Math instructions, ***the data types must match.*** For example, a BCD or decimal (binary) number cannot be added to a real number, and a BCD number cannot be added to a decimal (binary) number. If the data types are mismatched, the results of any math operation will be meaningless.

To simplify number conversions, Intelligent Box (IBox) Instructions are available with *Direct*SOFT. These instruction descriptions are located in Volume 1, page 5-230, in the Math IBox group.

Most *Direct*LOGIC analog modules can be set up to give the raw data in decimal (binary) format or in BCD format, so it is necessary to know how the module is being used. *Direct*LOGIC PID is another area where not all values are in BCD. In fact, nearly all of the PID parameters are stored in the PLC memory as decimal (binary) numbers.



NOTE: *The PID algorithm uses magnitude plus sign for negative decimal (binary) numbers, whereas the standard math functions use two's complement. This can cause confusion while debugging a PID loop.*

When using the Data View in *Direct*SOFT, be certain that the proper format is selected for the element to be viewed. The data type and length is selected using the drop-down boxes at the top of the Data View window. Also notice that BCD is called BCD/Hex. Remember that BCD is a subset of hexadecimal so they share a display format even though the values may be different. This is where good documentation of the data type stored in memory is crucial.

C-more/C-more Micro-Graphic Panels

In the C-more and C-more Micro-Graphic HMI operator panels, the 16-bit BCD format is listed as "BCD int 16". Binary format is either "Unsigned int 16" or "Signed int 16" depending on whether or not the value can be negative. Real number format is "Floating PT 32".

More available formats are, "BCD int 32", "Unsigned int 32" and "Signed int 32".

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

European Union Directives (CE)



In This Appendix...

European Union (EU) Directives I-2
Basic EMC Installation Guidelines I-4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

European Union (EU) Directives



NOTE: *The information contained in this section is intended as a guideline and is based on our interpretation of the various standards and requirements. Since the actual standards are issued by other parties, and in some cases governmental agencies, the requirements can change over time without advance warning or notice. Changes or additions to the standards can possibly invalidate any part of the information provided in this section.*

This area of certification and approval is absolutely vital to anyone who wants to do business in Europe. One of the key tasks that faced the EU member countries and the European Economic Area (EEA) was the requirement to bring several similar yet distinct standards together into one common standard for all members. The primary purpose of a single standard was to make it easier to sell and transport goods between the various countries and to maintain a safe working and living environment. The Directives that resulted from this merging of standards are now legal requirements for doing business in Europe. Products that meet these Directives are required to have a CE mark to signify compliance.

Member Countries

As of January 1, 2007, the members of the EU are Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithonia, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, and United Kingdom. Iceland, Liechtenstein, and Norway together with the EU members make up the European Economic Area (EEA) and all are covered by the Directives.

Applicable Directives

There are several Directives that apply to our products. Directives may be amended, or added, as required.

- **Electromagnetic Compatibility Directive (EMC)** — this Directive attempts to ensure that devices, equipment, and systems have the ability to function satisfactorily in an electromagnetic environment without introducing intolerable electromagnetic disturbance to anything in that environment.
- **Machinery Safety Directive** — this Directive covers the safety aspects of the equipment, installation, etc. There are several areas involved, including testing standards covering both electrical noise immunity and noise generation.
- **Low Voltage Directive** — this Directive is also safety related and covers electrical equipment that has voltage ranges of 50–1000VAC and/or 75–1500VDC.
- **Battery Directive** — this Directive covers the production, recycling, and disposal of batteries.

Compliance

Certain standards within each Directive already require mandatory compliance. The EMC Directive, which has gained the most attention, became mandatory as of January 1, 1996. The Low Voltage Directive became mandatory as of January 1, 1997.

Ultimately, we are all responsible for our various pieces of the puzzle. As manufacturers, we must test our products and document any test results and/or installation procedures that are necessary to comply with the Directives. As a machine builder, you are responsible for

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

installing the products in a manner which will ensure compliance is maintained. You are also responsible for testing any combinations of products that may (or may not) comply with the Directives when used together. The end user of the products must comply with any Directives that may cover maintenance, disposal, etc. of equipment or various components. *Although we strive to provide the best assistance available, it is impossible for us to test all possible configurations of our products with respect to any specific Directive. Because of this, it is ultimately your responsibility to ensure that your machinery (as a whole) complies with these Directives and to keep up with applicable Directives and/or practices that are required for compliance.*

As of January 1, 1999, the DL05, DL06, DL205, DL305, and DL405 PLC systems manufactured by Koyo Electronics Industries, FACTS Engineering or HOST Engineering, when properly installed and used, conform to the Electromagnetic Compatibility (EMC), Low Voltage Directive, and Machinery Directive requirements of the following standards.

- **EMC Directive Standards Relevant to PLCs**
  EN50081–1 Generic emission standard for residential, commercial, and light industry
  EN50081–2 Generic emission standard for industrial environment.
  EN50082–1 Generic immunity standard for residential, commercial, and light industry
  EN50082–2 Generic immunity standard for industrial environment.
- **Low Voltage Directive Standards Applicable to PLCs**
  EN61010–1 Safety requirements for electrical equipment for measurement, control, and laboratory use.
- **Product Specific Standard for PLCs**
  EN61131–2 Programmable controllers, equipment requirements and tests. This standard replaces the above generic standards for immunity and safety. However, the generic emissions standards must still be used in conjunction with the following standards:
  -EN 61000-3-2 Harmonics
  -EN 61000-3-2 Fluctuations
- **Warning on Electrostatic Discharge (ESD)**
  We recommend that all personnel take necessary precautions to avoid the risk of transferring static charges to inside the control cabinet, and clear warnings and instructions should be provided on the cabinet exterior. Such precautions may include, the use of earth straps, similar devices or the powering off of the equipment inside the enclosure before the door is opened.
- **Warning on Radio Interference (RFI)**
  This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

General Safety

- External switches, circuit breaker or external fusing, are required for these devices.
- The switch or circuit breaker should be mounted near the PLC equipment.

AutomationDirect is currently in the process of changing their testing procedures from the generic standards to the product specific standards.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Special Installation Manual

The installation requirements to comply with the requirements of the Machinery Directive, EMC Directive and Low Voltage Directive are slightly more complex than the normal installation requirements found in the United States. To help with this, we have published a special manual which you can order:

- **DA–EU–M** – EU Installation Manual that covers special installation requirements to meet the EU Directive requirements. Order this manual to obtain the most up-to-date information.

Other Sources of Information

Although the EMC Directive gets the most attention, other basic Directives, such as the Machinery Directive and the Low Voltage Directive, also place restrictions on the control panel builder. Because of these additional requirements it is recommended that the following publications be purchased and used as guidelines:

- BSI publication TH 42073: February 1996 – covers the safety and electrical aspects of the Machinery Directive
- EN 60204–1:1992 – General electrical requirements for machinery, including Low Voltage and EMC considerations
- IEC 1000–5–2: EMC earth grounding and cabling requirements
- IEC 1000–5–1: EMC general considerations

It may be possible for you to obtain this information locally; however, the official source of applicable Directives and related standards is:

The Office for Official Publications of the European Communities L–2985 Luxembourg; quickest contact is via the World Wide Web at http://publications.europa.eu/index_en.htm

Other sources are:

British Standards Institution – Sales Department
Linford Wood
Milton Keynes
MK14 6LE
United Kingdom; the quickest contact is via the World Wide Web at http://www.bsi.org.uk
A commercial provider of Standards at www.ihs.com

Basic EMC Installation Guidelines

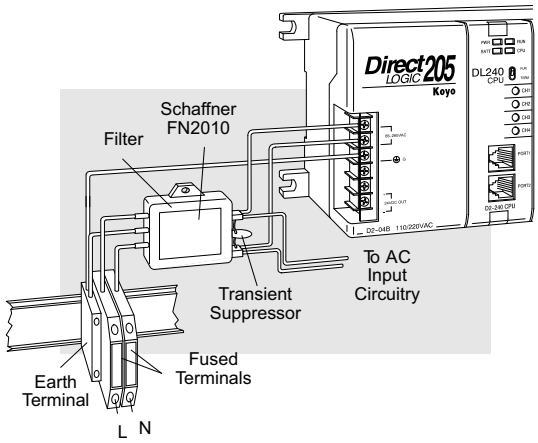
Enclosures

The simplest way to meet the safety requirements of the Machinery and Low Voltage Directives is to house all control equipment in an industry standard lockable steel enclosure. This normally has an added benefit because it will also help ensure that the EMC characteristics are well within the requirements of the EMC Directive. Although the RF emissions from the PLC equipment, when measured in the open air, are well below the EMC Directive limits, certain configurations can increase emission levels. Holes in the enclosure, for the passage of cables or to mount operator interfaces, will often increase emissions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AC Mains Filters

The DL305 AC powered base power supplies require extra mains filtering to comply with the EMC Directive on conducted RF emissions. All PLC equipment has been tested with filters from Schaffner, which reduce emissions levels if the filters are properly grounded (earth ground). A filter with a current rating suitable to supply all PLC power supplies and AC input modules should be selected. We suggest the FN2080 for DL305 systems.



NOTE: *Very few mains filters can reduce problem emissions to negligible levels. In some cases, filters may increase conducted emissions if not properly matched to the problem emissions.*

Suppression and Fusing

In order to comply with the fire risk requirements of the Low Voltage and Machinery Directive electrical standards (EN 61010–1 and EN 60204–1), by limiting the power into "unlimited" mains circuits with power leads reversed, it is necessary to fuse both AC and DC supply inputs. You should also install a transient voltage suppressor across the power input connections of the PLC. Choose a suppressor such as a metal oxide varistor, with a rating of 275VAC working voltage for 230V nominal supplies (150VAC working voltage for 115V supplies) and high energy capacity (eg. 140 joules).

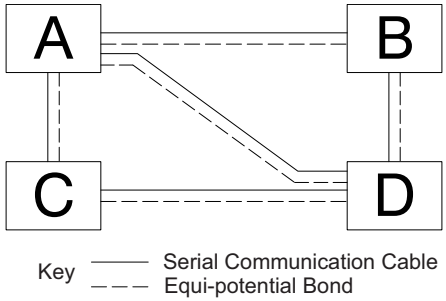
Transient suppressors must be protected by fuses and the capacity of the transient suppressor must be greater than the blow characteristics of the fuses or circuit breakers to avoid a fire risk. A recommended AC supply input arrangement for Koyo PLCs is to use twin 3 amp TT fused terminals with fuse blown indication, such as DINnectors DN–F10L terminals, or twin circuit breakers, wired to a Schaffner FN2010 filter or equivalent, with high energy transient suppressor soldered directly across the output terminals of the filter. PLC system inputs should also be protected from voltage impulses by deriving their power from the same fused, filtered, and surge-suppressed supply.

Internal Enclosure Grounding

A heavy-duty star earth terminal block should be provided in every cubicle for the connection of all earth ground straps, protective earth ground connections, mains filter earth ground wires, and mechanical assembly earth ground connections. This should be installed to comply with safety and EMC requirements, local standards, and the requirements found in IEC 1000–5–2.The Machinery Directive also requires that the common terminals of PLC input modules, and common supply side of loads driven from PLC output modules should be connected to the protective earth ground terminal.

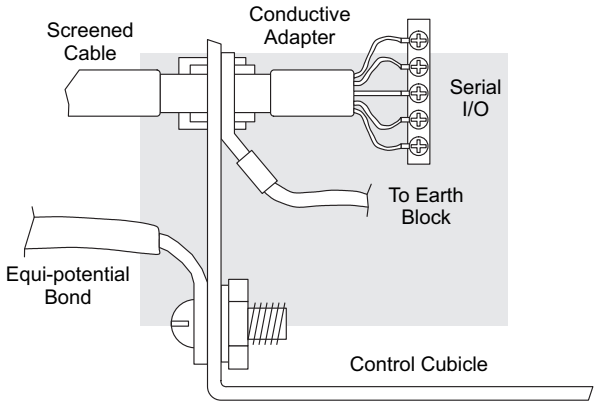
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Equi–potential Grounding



Adequate site earth grounding must be provided for equipment containing modern electronic circuitry. The use of isolated earth electrodes for electronic systems is forbidden in some countries. Make sure you check any requirements for your particular destination. IEC 1000–5–2 covers equi-potential bonding of earth grids adequately, but special attention should be given to apparatus and control cubicles that contain I/O devices, remote I/O racks, or have inter-system communications with the primary PLC system enclosure. An equi-potential bond wire must be provided alongside all serial communications cables, and to any separate items of the plant which contain I/O devices connected to the PLC. The diagram shows an example of four physical locations connected by a communications cable.

Communications and Shielded Cables



Good quality 24 AWG minimum twisted-pair shielded cables, with overall foil and braid shields are recommended for analog cabling and communications cabling outside of the PLC enclosure. To date it has been a common practice to only provide an earth ground for one end of the cable shield in order to minimize the risk of noise caused by earth ground loop currents between apparatus. The procedure of only grounding one end, which primarily originated as a result of trying to reduce hum in audio systems, is no longer applicable to the complex industrial environment. Shielded cables are also efficient emitters of RF noise from the PLC system, and can interact in a parasitic manner in networks and between multiple sources of interference.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

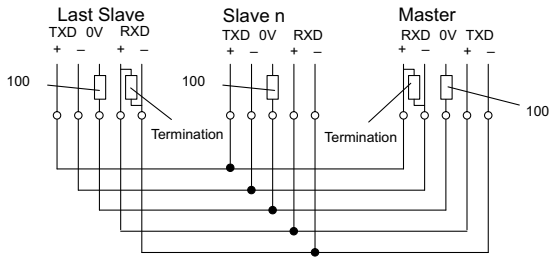
The recommendation is to use shielded cables as electrostatic "pipes" between apparatus and systems, and to run heavy gauge equi-potential bond wires alongside all shielded cables. When a shielded cable runs through the metallic wall of an enclosure or machine, it is recommended in IEC 1000–5–2 that the shield should be connected over its full perimeter to the wall, preferably using a conducting adapter, and not via a pigtail wire connection to an earth ground bolt. Shields must be connected to every enclosure wall or machine cover that they pass through.

Analog and RS232 Cables

Providing an earth ground for both ends of the shield for analog circuits provides the perfect electrical environment for the twisted pair cable as the loop consists of signal and return, in a perfectly balanced circuit arrangement, with connection to the common of the input circuitry made at the module terminals. RS232 cables are handled in the same way.

Multidrop Cables

RS422 twin twisted pair, and RS485 single twisted pair cables also require a 0V link, which has often been provided in the past by the cable shield. It is now recommended that you use triple twisted pair cabling for RS422 links, and twin twisted pair cable for RS485 links. This is because the extra pair can be used as the 0V inter-system link. With loop DC power supplies earth grounded in both systems, earth loops are created in this manner via the inter-system 0v link. The installation guides encourage earth loops, which are maintained at a low impedance by using heavy equi-potential bond wires. **To account for non–European installations using single-end earth grounds, and sites with far from ideal earth ground characteristics, we recommend the addition of 100 ohm resistors at each 0V link connection in network and communications cables.**



Shielded Cables within Enclosures

When you run cables between PLC items within an enclosure which also contains susceptible electronic equipment from other manufacturers, remember that these cables may be a source of RF emissions. There are ways to minimize this risk. Standard data cables connecting PLCs and/or operator interfaces should be routed well away from other equipment and their associated cabling. You can make special serial cables where the cable shield is connected to the enclosure's earth ground at both ends, the same way as external cables are connected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Analog Modules and RF Interference

All Automationdirect products are tested to withstand field strength levels up to 10V/m, which is the maximum required by the relevant EU standards. While all products pass this test, analog modules will typically exhibit deviations of their readings. This is quite normal, however, systems designers should be aware of this and plan accordingly.

When assembling a control system using analog modules, these issues must be adhered to and should be integrated into the system design. This is the responsibility of the system builder/commissioner.

Network Isolation

For safety reasons, it is a specific requirement of the Machinery Directive that a keyswitch must be provided that isolates any network input signal during maintenance, so that remote commands cannot be received that could result in the operation of the machinery. The FA–ISOCON does not have a keyswitch! Use a keylock and switch on your enclosure which when open removes power from the FA–ISOCON. To avoid the introduction of noise into the system, any keyswitch assembly should be housed in its own earth grounded steel box and the integrity of the shielded cable must be maintained.

Again, for further information on EU directives we recommend that you get a copy of our EU Installation Manual (DA–EU–M). Also, if you are connected to the World Wide Web, you can check the EU Commission's official site at: http://ec.europa.eu/index_en.htm.

DC Powered Versions

Due to slightly higher emissions radiated by the DC powered versions of the DL205, and the differing emissions performance for different DC supply voltages, the following stipulations must be met:

- The PLC must be housed within a metallic enclosure with a minimum amount of orifices.
- I/O and communications cabling exiting the cabinet must be contained within metallic conduit/trunking.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Items Specific to the DL205

- The rating between all circuits in this product are rated **as basic insulation only**, as appropriate for single fault conditions.
- There is no isolation offered between the PLC and the analog inputs of this product.
- It is the responsibility of the system designer to earth one side of all control and power circuits, and to earth the braid of screened cables.
- This equipment must be properly installed while adhering to the guidelines of the installation manual DA–EU–M (available for download at AutomationDirect Technical Support Manuals), and the installation standards IEC 1000–5–1, IEC 1000–5–2 and IEC 1131–4.
- It is a requirement that all PLC equipment must be housed in a protective steel enclosure, which limits access to operators by a lock and power breaker. If access is required by operators or untrained personnel, the equipment must be installed inside an internal cover or secondary enclosure. A warning label must be used on the front door of the installation cabinet as follows:
  Warning: Exposed terminals and hazardous voltages inside.
- It should be noted that the safety requirements of the machinery directive standard EN60204–1 state that all equipment power circuits must be wired through isolation transformers or isolating power supplies, and that one side of all AC or DC control circuits must be earthed.
- Both power input connections to the PLC must be separately fused using 3 amp T-type anti–surge fuses, and a transient suppressor fitted to limit supply overvoltages.
- If the user is made aware by notice in the documentation that if the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INDEX



A

Accumulator/Stack Instructions, 5–53
  Pointers, 5–57
Analog Potentiometer, 3–19
ASCII Conversion Table, G-2
ASCII Instructions, 5–211
Auto tuning error, 8–48
Auto Tuning Procedure, 8–45
Auxiliary Functions, A–2
  Accessing, A–3
    *Direct*SOFT, A–3
    Handheld Programmer, A–3

B

Basic EMC Installation Guidelines, I-4
Binary Numbering System, H–2
Bit Map, 3–57
  Control Relay, 3–59
  Remote I/O, 3–66
  Stage Control/Status, 3–63
  Timer and Counter Status, 3–65
  X Input/Y Output, 3–57
Bit Operations, 5–123
Boolean Instructions, 5–5, 5–10
Bumpless Transfer, 8–13, 8–27, 8–75, 8–76

C

Cascade Control, 8–64
  Tuning, 8–66
Clock and Calendar, 3–16
Clock/Calendar Instructions, 5–175
Comparative Boolean, 5–27
Converge Jump instruction, 7–25
Converge Stage instruction, 7–25
Convergence Jump instruction, 7–20
Convergence Stages, 7–19
Counter Instructions, 5–46
CPU Control Instructions, 5–177
CPU Error Indicators, 9–10
  PWR Indicator, 9–11
CPU Features, 3–2
  Mode Switch, 3–12
  Program Storage Media, 3–9
  Setting the CPU Network Address, 3–17
  Setup, 3–6
  Specifications, 3–4
  Status Indicators, 3–12
CPU Operation, 3–21
  I/O Response Time, 3–27
  Scan Time Considerations, 3–29

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D

Data Type Mismatch, H–7
DL205 Aliases, 3–52
DL205 PLC Memory, E-2
Drum Instruction (DRUM), 6–12
  chart representation, 6–3
  counter assignments, 6–6
  drum control techniques, 6–10
  event drum (EDRUM), 6–14
  handheld programmer mnemonics, 6–16
  masked event drum (MDRMD)(MDRMW), 6–19, 6–21
  Operation, 6–8
  powerup state, 6–9
  self-resetting, 6–11
  step transitions, 6–11
  Terminology, 6–2

E

Environmental Specifications, 2–8
Error Codes, 9–6, B–2
Error Term Selection, 8–34
European Union Directives, I-2
Expanding DL205 I/O, 4–17
  Ethernet Base Controller, 4–22
  Ethernet Remote Master, 4–17
  Serial Remote I/O Master/Slave, 4–26

F

Feedforward Control, 8–69
Freeze Bias, 8–11, 8–35

H

Hardware Maintenance, 9–2
  Communications, 9–13
  Diagnostics, 9–3
Hexadecimal Numbering System, H–3
Hysteresis, 8–13, 8–37, 8–38, 8–39

I

I/O Module Troubleshooting, 9–14
I/O System Configurations, 4–2
Immediate Instructions, 5–33
Initial Stage (ISG), 7–24
Installation, 2–10
  Base Wiring, 2–13
  Connecting DC I/O, 2–19
  I/O Modules Position, 2–25
  Surge Suppression, 2–21
Instruction Execution Times, C–2
Instruction Table, 5–2
Instructions
  stage, 7–23
  stage programming, 7–2
Intelligent Box (IBox) Table, 5–230
Intelligent I/O Instructions, 5–191
Interrupt Instructions, 5–187

J

Jump instruction, 7–7

L

Local Expansion I/O, 4–11
Loop Mode, 8–28
Loop Table Word Definitions, 8–20

M

Machine Startup, 9–18
Memory Map, 3–37, 3–53
  DL230, 3–53

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DL240, 3–54
DL250–1, 3–55
DL260, 3–56
Message Instructions, 5–197
Modbus RTU Instructions, 5–205
Mounting Guidelines, 2–5
Component Dimensions, 2–10

N

Network Connections, 4–32
Configuring Port 2, 4–32
Network Instructions, 5–193
Network Master Operation, 4–41
Network Modbus RTU Master Operation, 4–45
Network Slave Operation, 4–35
Modbus, 4–35
Noise Troubleshooting, 9–17
Non-volatile V-memory, E-3
Non–Sequence Protocol, 4–54
Configure the DL250-1 Port 2, 4–56
Not Jump, 7–24
Number Conversion Instructions, 5–130

O

Octal Numbering System, H–4

P

Parallel Processing, 7–19
PID
Analog Filter, 8–54
*Direct*SOFT Filter Intelligent Box Instruction, 8–56
Error Flags, 8–18
Example Program, 8–71
Loop Modes, 8–28, 8–52, 8–53, 8–65
Parameters, 8–33
Setup Alarms, 8–36
Special Features, 8–52
PID Alarms, 8–36
Calculation Overflow/Underflow Error, 8–39
Hysteresis, 8–39
Mode/Alarm Bit Description, 8–23
Monitor Limit, 8–36
Programming Error, 8–39
PV Deviation, 8–37
Rate-of-Change, 8–38
PID Loop
Alarms, 8–13
Auto Tuning, 8–41
Configure, 8–26
Manual Tuning, 8–41, 8–42, 8–44, 8–47, 8–54
Mode, 8–28
Operating Modes, 8–14
Operation, 8–9
Program Setup, 8–71
Reverse Acting, 8–12, 8–14, 8–27, 8–41
Setup, 8–18
Terminology, 8–76
Troubleshooting Tips, 8–74
PID Mode 2 Word Description, 8–22
PID Mode Setting 1 Description, 8–21
PLC Numbering Systems, 3–35
Position Algorithm, 8–9, 8–15, 8–77
Position Form, 8–9
Power Budget, 4–7
Product Weight Table, F-2
Products and Data Types, H–9
Program Control Instructions, 5–179

Q

Quick Start, 1–10

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

R

Ramp/Soak Generator, 8–57
Controls, 8–60
*Direct*SOFT Ramp/Soak Example, 8–62
Profile Monitoring, 8–61
Ramp/Soak Flag Bit Description, 8–23
Ramp/Soak Table Location, 8–24
Relay Ladder, 8–62
Table, 8–58
Table Flags, 8–60
Test Profile with PID View, 8–63
Testing, 8–61
Rate-of-Change, 8–13, 8–14, 8–38
Real Numbering System, H–5
Reset Windup, 8–10, 8–35, 8–77

S

Safety Guidelines, 2–2
Emergency Stops, 2–3
Shift Register Instruction, 5–52
Signed vs. Unsigned Integers, H–8
Special Relays, D–2
Square Root, 8–14
Stage instructions, 7–23
Stage Programming, 7–2, 7–15
convergence, 7–19
emergency stop, 7–14
Example, 7–10
four steps to stage programming, 7–9
introduction, 7–2
jump instruction, 7–7
mutually exclusive transitions, 7–14
parallel processing concepts, 7–19
program organization, 7–15
questions and answers, 7–29
stage instruction characteristics, 7–6
state transition diagrams, 7–3
timer inside stage, 7–13
State Diagram, 7–11
Step Transitions, 6–4
System Components, 1–4
*Direct*LOGIC™ Part Numbering System, 1–8
System V-memory, 3–41
DL230, 3–41
DL240, 3–43
DL250–1, 3–46
DL260, 3–49

T

Table Instructions, 5–144
Technical Support, 1–2
Time-Proportioning Control, 8–67
On/Off Control Program, 8–68
Timer Instructions, 5–41
Transcendental Functions, 5–121
Transfer Mode, 8–27

U

Using a Password, 3–18
Using Battery Backup, 3–14

V

Velocity Algorithm, 8–9, 8–15, 8–77
Velocity Form, 8–12

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP