WindLDR™ Tutorial

© 2001, IDEC Corporation. All rights reserved. Produced and printed in the United States of America.

**Important Information**

While every effort has been made to ensure that the information contained within this document is accurate, under no circumstances shall IDEC Corporation be held liable or responsible for indirect or consequential damages resulting from the use of or the application of IDEC programmable logic controller components, individually or in combination with other equipment.

In line with our policy of continuous improvement, we reserve the right to make product changes without notice.

All persons using these components must be willing to accept responsibility for choosing the correct component to suit their application and for choosing an application appropriate for the component, individually or in combination with other equipment.

All diagrams and examples in this manual are for illustrative purposes only. In no way does the inclusion of these diagrams and examples in this manual constitute a guarantee as to their suitability for any specific application. To test and approve all programs prior to installation is the responsibility of the end user.
By using IDEC Corporation's software product, you are consenting to be bound by the IDEC Corporation software end user license agreement included as a part of this package. If you do not agree to all of the terms of this agreement, you must erase all installed IDEC Corporation software files from your system.

**User Comments**

IDEC Corporation is eager to receive comments about its products and documentation. If you have any comments, please phone, fax, or mail them:

IDEC Corporation
1175 Elko Drive
Sunnyvale, CA 94289

PH: 800-262-4332
FAX: 800-635-6246

www.idec.com
# TABLE OF CONTENTS

## Chapter 1
- Document Conventions ........................................ 1-1
- On-Line Help ................................................. 1-3
- Technical Support ............................................ 1-3

## Chapter 2
- What is a PLC .................................................. 2-1
- Elements of a PLC System .................................... 2-2
  - Programming Device ........................................ 2-3
  - Power Supply ................................................ 2-3
  - Central Processing Unit (CPU) .............................. 2-3
  - Memory ....................................................... 2-3
  - Input Interface .............................................. 2-4
  - Output Interface ............................................ 2-4
- How PLCs Work .................................................. 2-5
- PLC Scan .......................................................... 2-6
  - Read Inputs .................................................. 2-7
  - Execute Programmed Logic ................................. 2-7
  - Update Outputs .............................................. 2-8
  - Communications and Housekeeping ...................... 2-8
- Defining Ladder Logic Input Conditions .................... 2-8
  - Examine-On Contact ....................................... 2-8
  - Examine-Off Contact ..................................... 2-8

© 2001 IDEC CORPORATION
Chapter 3

Getting Started ......................................................... 3-1
   Installing WindLDR  ........................................ 3-1
   Starting WindLDR .............................................. 3-2
   Program Editing Features ...................................... 3-3
   Deleting an Entry ............................................... 3-4
   Selecting Your PLC Type ........................................ 3-5
   Configuring PLC Function and Communication Settings .......... 3-8
   Choosing Ladder Preferences .................................. 3-12

Chapter 4

Introduction to Basic Programming ................................. 4-1
   Input Instructions ............................................. 4-1
   Output Instructions ........................................... 4-2
Exercise 1: Creating a Start/Stop Program ......................... 4-3
   Closing a Program ............................................ 4-10
   Configuring the Communication Port .......................... 4-11
   Transferring a Program ....................................... 4-12
   Monitoring a Program ......................................... 4-13
Online ................................................................. 4-14
   1. PLC Status .................................................. 4-14
   2. Batch Monitor ............................................... 4-15
   3. Communication Error ...................................... 4-16
   4. Point Write .................................................. 4-17
NO and NC Conventions .............................................. 4-18
   NC Inputs .......................................................... 4-19
Exercise 2: Creating a Multiple Start/Stop Program .................. 4-20
Chapter 5

Introduction to Timers (TML, TIM, TMH, and TMS) ................................ 5-1
- Micro-1, FA1J/2J, FA3S-CP11/-CP11T Timers ........................................ 5-1
- FA3S-CP12/13 Timers ...................................................................... 5-1
- Micro³ and Micro³C Timers ............................................................. 5-3
- ONC Timers .................................................................................. 5-4
- MicroSmart Timers ........................................................................ 5-5
Exercise 3: Creating an On-Delay Timer Program .......................... 5-7
Exercise 4: Timer Comparison Instruction (TC= and TC≥) ............. 5-15

Chapter 6

Introduction to Counters (CNT) ......................................................... 6-1
- Micro-1, FA1J/2J, FA3S-CP11/-CP11T Counters ............................... 6-1
- Micro3, Micro3C, ONC and MicroSmart Counters ......................... 6-2
Introduction to Counter Comparisons .............................................. 6-6
(CC= and CC>) ............................................................................. 6-6
Exercise 5: Creating a Counter Program ......................................... 6-8

Chapter 7

SET and RST (Reset) ........................................................................ 7-1
Exercise 6: Creating a Set/Reset Program ........................................ 7-2

Chapter 8

Micro³, Micro³C, ONC, and MicroSmart Single Output Up (SOTU) and
Down (SOTD) Instructions ................................................................. 8-1
- Micro-1, FA-Js, and FA3Ss: Single Output (SOT) Instruction ........... 8-2
Exercise 7: Creating an SOTU/SOTD Program  
(Micro³, Micro³C, ONC, and MicroSmart) 8-3

Exercise 8: Creating an SOT Program  
(Micro-1 and FA Series PLCs) 8-8

Chapter 9

Data Move (Micro³, Micro³C, ONC and MicroSmart) . . . . . . . . . . . . . . . . . . . . . . 9-1
Structure of an Advanced Instruction 9-1
Input Condition for Advanced Instructions 9-2
Source and Destination Operands 9-3
Using a Timer or Counter as the Source Operand 9-3
Using a Timer or Counter as the Destination Operand 9-4
Using an Input or Output as the Source or Destination Operand 9-4
Exercise 9: Creating a Data Move Program (Micro³, Micro³C, ONC and MicroSmart series) 9-6
Data Move (FA series) 9-10
Operand Designation 9-10
Exercise 10: Creating a Data Move Program (FA series) 9-11

Chapter 10

Binary Arithmetic Instructions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10-1
Addition (ADD) 10-1
Subtraction (SUB) 10-2
Multiplication (MUL) 10-2
Division (DIV) 10-3
Root (ROOT) — ONC and MicroSmart Only 10-3
Sum (SUM) — ONC Only 10-4
Valid Operands (Standard Processing) 10-5
Comparison Instructions

(Micro\(^3\), Micro\(^3\)C, ONC and MicroSmart only) 10-7
Compare Equal To (CMP=) 10-7
Compare Not Equal To (CMP≠) 10-8
Compare Less Than (CMP<) 10-8
Compare Greater Than (CMP>) 10-9
Compare Less Than or Equal To (CMP≤) 10-9
Compare Greater Than or Equal To (CMP≥) 10-10
Interval Compare Greater Than or Equal to (ICMP≥) — ONC and MicroSmart Only 10-10
Valid Operands (Standard Processing) 10-11

Exercise 11: Creating a Binary Arithmetic/Comparison Program 10-12

Binary Arithmetic Instructions: FA Series PLCs 10-19
Add with Carry (JADDC) 10-19
Subtraction with Borrow (JSUBB) 10-20
Multiplication (JMUL) 10-21
Division (JDIV) 10-22
10,000 Addition 3 Operand (TADD3) 10-23
10,000 Subtraction 3 Operand (TSUB3) 10-23
10,000 Multiplication (TMUL) 10-23
10,000 Division (TDIV) 10-24

Comparison Instructions: FA Series PLCs 10-24
Word Greater Than or Equal To (WGE) 10-24
Word Less Than or Equal To (WLE) 10-25
Word Equal To (WEQ) 10-26

Exercise 12: Creating an FA Series Advanced Instruction Program 10-27
Chapter 11

Introduction to Analog Read Instructions

- Analog Read 0 (ANR0) ........................................ 11-1
- Analog Read 1 (ANR1) ........................................ 11-2
- Valid Operands (Standard Processing) ...................... 11-2
- Analog Potentiometer Setting for Micro3 and Micro3C .... 11-3
- Example: ANR0 ............................................. 11-3

Exercise 13: Creating an Analog Read Program

- Analog Potentiometer Setting for MicroSmart ................ 11-9
- Example: Changing Counter Present Value Using Analog Potentiometer ........................................ 11-10

Chapter 12

Introduction to Analog/Digital Instructions

- Analog/Digital (A/D) Conversion .................................. 12-1
- Valid Operands (Standard Processing) ...................... 12-2

Exercise 14: Creating an A/D Conversion Program ................. 12-3

Chapter 13

Calendar and Clock

Calendar/Clock Instructions:

- (Micro3 and Micro3C) ........................................ 13-1
- Calendar Read (CALR) ........................................ 13-2
- Example: CALR ............................................. 13-2
- Clock Read (CLKR) ........................................ 13-3
- Example: CLKR ............................................. 13-3
- Valid Operands (Standard Processing) ...................... 13-4
- Repeat Operation in the Move Instruction ................... 13-4
WindLDR™ Tutorial

Exercise 15: Creating a Calendar/Clock Program 13-6
Real-Time Clock and Calendar (ONC) 13-20
  The Week Comparison Instruction 13-20
  Week Compare ON (WKCMP ON) 13-21
  Week Compare OFF (WKCMP OFF) 13-21
  Week Table (WKTBL) 13-22
  Examples: WKCMP ON/OFF 13-23
Setting Calendar/Clock Using WindLDR 13-25
Setting Calendar/Clock Using a User Program 13-27
Real-Time Clock and Calendar (FA3S-CP11T) 13-30
  Clock Set (CLKST) 13-31
  Assignment Codes 13-32
  Clock Read (CLKRD) 13-33
  Clock Adjust (CLKAJ) 13-34
  Clock Comparison Format (CLKCF) 13-35
Clock Function (MicroSmart) 13-35
  Setting Calendar/Clock 13-36
  Setting Calendar/Clock Using a User Program 13-37
  Example: Setting Calendar/Clock Data 13-38
  Adjusting Clock Using a User Program 13-39
  Example: Adjusting Calendar/Clock Data to 0 Seconds 13-39

Chapter 14

Transmit and Receive ................................. 14-1
  Transmit (TXD) 14-1
  Valid Operands (Standard Processing) 14-2
  Data Registers 14-5
Exercise 16: Creating a Transmit Program 14-6
Transmit and Receive Instructions for
Chapter 15

XY Conversion ................................................................. 15-1
  XY Format Set (XYFS) .................................................. 15-1
  Valid Operands (Standard Processing) ......................... 15-2
  Valid Data Types ....................................................... 15-2
  Valid Operands (Standard Processing) ......................... 15-3
  Valid Operands (Standard Processing) ......................... 15-4
  Valid Operands (Standard Processing) ......................... 15-5
Exercise 17: Creating an X/Y Linear Relationship .......... 15-6

FA3S Series XY Conversion ............................................ 15-17
  XY Format Set (XYFS) Linear Conversion .................... 15-17
  Convert X to Y (CVXTY) ............................................. 15-17

Index .............................................................................. Index I
The WindLDR™ Tutorial is an introduction to IDEC’s third release of WindLDR™ ladder logic programming and monitoring software for all of programmable logic controllers (PLCs): the Micro series, the FA series, the ONC series and the MicroSmart. This tutorial is a good starting point for first time users of IDEC’s WindLDR and PLCs. Once you are familiar with WindLDR, the WindLDR™ Reference Manual and the user’s manual specific to your PLC will be your primary reference guides.

This tutorial provides the basics to get you up and running. You will accomplish the following tasks by completing the exercises in this manual.

- Create/edit ladder logic programs using WindLDR’s easy-to-use program editing features
- Print your ladder logic programs for hardcopy review
- Download programs to your PLC and monitor their operation
- Monitor the program for testing and troubleshooting purposes

**Document Conventions**

This tutorial is divided into two sections: basic and advanced instructions. Basic instructions are discussed in Chapters 4 through 8. Advanced instructions are discussed in Chapters 9 through 15.
Basic instructions are applicable to all PLC types. However, differing advanced instructions are applicable to each of the Micro³, Micro³C, FA series, ONC series and MicroSmart PLCs. Unless otherwise noted at the beginning of each chapter, instructions are applicable for all PLC series.

Please refer to the user’s manual that accompanied your PLC for a more detailed explanation of the basic and advanced instructions.

The following conventions are applicable to this manual:

Action items are denoted by the following arrow graphic.

This symbol appears before each set of instructions. Follow these steps to easily complete each lesson.

Action items requiring a specific entry appear bold and underlined. For example, enter 1234.

Menu instructions are separated by an arrow (➔). For example, “Click Configure ➔ PLC Selection” means to click Configure from the Menu Commands toolbar and then to click PLC Selection from the drop down menu.

While all keyboards are set up uniquely, the ENTER, RETURN, or NEXT key is always written as [ENTER] in this manual.

The majority of the WindLDR screens and dialog boxes are applicable to all PLC types. However, some screens and dialog boxes differ depending on the PLC type. Unless there is a difference among PLC types, all screen and dialog box examples in this tutorial will reflect the ONC PLC.
On-Line Help

Help screens are available from most menus and screen options. These Help screens provide on-line assistance to guide you through a process or procedure.

To access On-Line Help, click the ? icon in the Menu Commands toolbar at any time.

Technical Support

Registered users of IDEC’s WindLDR software are entitled to telephone support. To qualify, complete the registration card that came with your WindLDR software, and return it to IDEC. For assistance, call 1-800-262-IDEC (4332). Press 2 and then 1 to contact the PLC Technical Engineering Group. To contact us through E-mail, write: support@idec.com via the Internet.
**What is a PLC**

Before you start this tutorial, you should have a basic understanding of how your PLC works. If you are unfamiliar with PLCs, this chapter provides a good starting point for this tutorial. If you already have a general knowledge of PLCs, you can skip this chapter and go directly to *Chapter 3, Getting Started*.

Functionally, a PLC examines the status of input interfaces and, in response, controls something through output interfaces. Combinations of input and output data are referred to as **logic**. Several logic combinations are usually needed to carry out a control plan or program, as it is commonly called. This program is stored in memory using a programming device. All logic combinations stored in memory are periodically evaluated by the processor in a predetermined order.
Elements of a PLC System

All PLC systems consist of five basic components and a programming device to enter instructions. With larger PLCs, these components are usually modular, and a system is constructed by placing the appropriate modules in a rack. In the larger I/O PLCs, like the FA series, the ONC, and the 24-I/O MicroSmart, the power supply, the CPU, and the I/O cards are all separate modules; they are then combined to complete the PLC. In the smaller “brick type” controllers like the Micro-1, Micro³, Micro³C, and the 10-,16-I/O MicroSmart, all of the components are usually contained in the same housing.
Programming Device
The programming device allows the programmer to enter instructions into the PLC's memory. This is typically accomplished either by a hand-held programmer or by a PC with programming software such as IDEC’s WindLDR. IDEC offers both methods of programming.

Power Supply
The power supply takes line voltage (typically 120V AC) and converts it to 12V or 24V DC which is required by the CPU and the internal circuits of the input/output interface.

Central Processing Unit (CPU)
The CPU is the brain of the PLC. It receives input data, performs logical decisions based on the program, and drives the outputs accordingly.

Memory
The processor memory stores the user's program, I/O data, timer and counter values, analog values, and any control constants that the programmer creates. IDEC PLC user programs are stored permanently in nonvolatile Electrically Erasable Programmable Read Only Memory (EEPROM). Programs stored in EEPROM are safe during power outages and can be reprogrammed with the proper equipment (that is, a program loader).

Unlike EEPROM, Random Access Memory (RAM) temporarily stores register, timer, and counter values in the PLC. Although RAM is volatile, it is backed up by a 30 day capacitor.
**Input Interface**

The input interface provides an electrical interface between the input devices and the processor's internal logic control circuits. Input devices include such items as pushbuttons, float and limit switches, operator selector switches, pressure switches, and sensors.

![Input Circuit Program](image)

**Input Circuit Program**

In the illustration above, when the pushbutton is closed, the light emitting diode (LED) turns on. This light in turn switches on a phototransistor connected to the internal circuitry. This means of keeping the input wiring and internal circuitry electrically separate is known as *optical isolation*.

**Output Interface**

The output interface provides the means of using the processor's low voltage signals (12V DC) to switch power to a field output device. This is done either with an electromechanical relay (see figure on the next page) or phototransistor.

Output devices include such items as pilot lights, relays, alarms, digital displays, motor starters, valves, and solenoids.

**NOTE:** The output voltage to the device is only *switched* by the processor's power supply. The output voltage is not *supplied* by the processor's power supply.
How PLCs Work

PLCs are generally programmed in what is known as ladder logic. This method of programming was established because it could be closely related to the hardwired relay logic (see figure below) that PLCs replaced.

Typical Relay Logic Diagram
The hardwired systems operated just as the previous diagram shows. The power was supplied to the circuit; and, if there was a path for current to travel through, then the output would be energized. The concept of ladder logic programming is identical to this. The entire program is solved each scan — starting from the first element on the top left of the diagram, then solving left to right, top to bottom, just as current would flow. In the following ladder logic diagram, the order of solution would start at A and proceed sequentially through J.

![Ladder Logic Sequential Flow Diagram](image)

**PLC Scan**

The operations carried out in a PLC are accomplished by a sequential process referred to as *scanning*. The scan time is most often the criterion when PLC speed is questioned. It is essentially how quickly the PLC can look at its inputs, solve the user’s programmed logic, and then update the outputs accordingly. A more elaborate description is given on the following page.
**Read Inputs**

Input(s) status is (are) read into the input table. The input table does not change until the next cycle of the scan when the input(s) status is (are) read in again.

**Execute Programmed Logic**

The program that is stored in the PLC's memory is executed or solved. Ladder logic is solved sequentially left to right and then top to bottom one rung (ladder logic line) at a time. Logic results of the outputs are set to an internal output table.
Update Outputs
The output(s) status in the internal output table is (are) sent to the physical output(s) (e.g., relay, starter, pilot light).

Communications and Housekeeping
If required, information is exchanged with other devices. Single scan transitions are logged, timers and counters are updated, and a system diagnostics check is performed in preparation for the next scan.

NOTE: Steps 1, 3, and 4 are consolidated in the PLC’s END instruction.

Defining Ladder Logic Input Conditions
A typical line or rung in a ladder logic program contains several input conditions. Each condition is like an “if” statement. “If” a condition is “true,” then a particular action will take place by the PLC. Likewise, “if” the same condition is “not true,” then the action will not take place. The program is solved in this manner by the PLC determining if the defined conditions are true or not. Conditions are defined by the programmer as contacts—either an Examine-On contact or an Examine-Off contact.

Examine-On Contact
Examine-On contact is true when the input terminal is hot (the LED light on the input is ON).

Examine-Off Contact
Examine-Off contact is true when the input terminal is not hot (the LED light on the input is OFF), regardless of whether the input device (pushbutton, etc.) is considered a normally open (NO) or normally closed (NC) device.
3. Getting Started

Getting Started
This tutorial walks you through the process of creating basic programs using the IDEC PLC instruction sets. Each chapter introduces you to a new instruction and includes a programming exercise using that instruction.

Before you begin this tutorial, you must first complete the following steps.

Installing WindLDR

The WindLDR software is supplied on a CD and is accessed from the PC after installing the files onto the hard disk. When copying the software files to the PC, the application creates a directory in which the files will reside.

The WindLDR destination directory name is configurable but defaults to WindLDR. This directory holds all of the executable files.

To install the WindLDR software:

1. Insert the installation CD into the PC’s CD Rom drive.

2. In Windows 95, 98, 2000 and NT4.0 click the Start ➔ Run from the task bar.

3. Enter D:setup.exe in the Run dialog box. The “D” indicates the CD drive. Your PC’s CD drive may be designated by another letter.
3. Getting Started © 2001 IDEC Corporation

A series of screens displays to walk you through the installation process.

4. When prompted for the destination directory, you can use the default directory, C:\Program Files\IDEC\WindLDR, or enter another destination directory for the installation files.

**Starting WindLDR**

When you install WindLDR, use Windows Explorer to create an icon for WindLDR on the PC screen.

To start WindLDR, double-click the WindLDR icon.

The WindLDR screen displays.
NOTE: You can enlarge your screen by clicking the maximize button in the upper right corner of the WindLDR screen and then clicking the maximize button in the upper right corner of the ladder program screen.

Program Editing Features

There are four ways to create/edit WindLDR programs:

1. Advanced and Basic Instruction Toolbars

Each icon in these toolbars represents a programming instruction. To insert an instruction in your program, simply click on the instruction icon and then click in the area of your program screen where you wish to place your instruction. This “point and click,” easy-to-use method is the convention for all exercises in this tutorial.

2. Hot Keys

To use a “hot key” to access an instruction, type the first letter of the instruction you wish to insert (e.g., “O” for Output). The Coil Selection Dialog box lists all instructions beginning with “O.” You can then scroll down to the desired instruction, and click the OK button or press the [ENTER] key to confirm your selection.

3. Cut and Paste

Select the coil, ladder line, or rung using your mouse. Then, either click the Copy icon or use [CNTR] + [c]. Finally, select the location on your program screen where you wish to place your instruction, and either click the Paste icon or use [CNTR] + [v].
4. Right Mouse Button

You can also access instructions by clicking your right mouse button and selecting the desired instruction from a series of drop down menus. For a more detailed description of functions accessible through the right mouse button, please refer to the *WindLDR™ Reference Manual*.

**Deleting an Entry**

To delete an instruction from your program screen, click on the instruction and press the [DELETE] key, or click your right mouse button to access the Delete command.

To delete a rung, column, or ladder line, click on the appropriate area, and then click your right mouse button to access the Delete command.

**NOTE:** To delete a line (vertical or horizontal), use the Eraser icon in the Basic Instructions Toolbar. Or, use the keyboard commands: [CNTR] + [ → ], [ ← ], [ ↑ ], or [ ↓ ] as desired.

To create a line (vertical or horizontal), use the Pencil icon in the Basic Instructions Toolbar. Or, use the keyboard commands: [SHIFT] + [ → ], [ ← ], [ ↑ ], or [ ↓ ] as desired.
Selecting Your PLC Type

You can use WindLDR to program all of IDEC’s PLCs. However, you must choose your specific PLC type before you begin programming.

To select the correct PLC:

1. Click **Configure → PLC selection…** from the Menu Commands toolbar, or click the **Select PLC** icon.

   The **PLC Selection** dialog box displays.

2. Click the correct PLC type.

3. Click the **Configure** button.

   The **PLC Configuration** dialog box displays.

4. For the Micro-1, click the expansion unit (if needed) for your PLC. For the Micro³, Micro³C or Micro Smart, click the specific base unit and expansion unit (if needed) for your PLC. For the FA and ONC series, there is no need for additional selections.
3. Getting Started

3-6 WindLDR™ Tutorial
5. Click the OK button or press the [ENTER] key to confirm. Notice that you return to the PLC selection dialog box.

6. Click the Use as Default button at the bottom of the PLC selection dialog box. The selected PLC type displays in the Default: field.

7. Click the OK button or press the [ENTER] key to confirm. The selected PLC type appears in the status bar at the bottom of the WindLDR screen.

**IMPORTANT:** When installed, WindLDR defaults to the ONC. If you select another PLC, a message appears: “Changing the PLC will cause certain changes to be made. These changes cannot be undone. Proceed?” Click Yes to accept this.

The correct toolbars for your PLC display.
Configuring PLC Function and Communication Settings

After you choose the correct PLC type, make sure that the PLC function settings are configured properly. For the purpose of this tutorial, choose the default function settings.

To configure the function settings:

1. Click Configure ➔ Function area settings… from the Menu Commands toolbar.

2. The Function Area Settings dialog box displays.

3. Click the Default button for every selection to accept it. Otherwise, perform the necessary changes, and click OK.

4. Click the OK button or press the [ENTER] key to confirm.

**NOTE:** The Function Area Setting dialog box for the ONC differs from the dialog box for the Micro, FA series and MicroSmart, as shown on the following page.

Please refer to the user’s manual included with your PLC for a detailed explanation of function settings.

**TIP:** The Stop Input field in the Function Setting window allows you to physically bring your PLC to the stop or run mode with a designated input switch.
Micro-1 Function Area and Communications Settings Dialog Boxes

Micro³ and Micro³C Function Area and Communications Settings Dialog Boxes
ONC Function Area and Communications Settings Dialog Boxes

FA Series Function Area and Communications Settings Dialog Boxes
MicroSmart Function Area and Communications Settings Dialog Boxes
Choosing Ladder Preferences

To customize your WindLDR ladder logic program screen, click Configure ➔ Ladder preferences… from the Menu Commands toolbar. For the purposes of this tutorial, choose the settings shown below.

To select the font type and size, click the Font button. The recommended font type and size is New Courier, Bold, 10 pt.

You are now ready to start the tutorial.

**NOTE:** The text size of the rung comments is not restricted. In the Tag Comment Lines: field, you can enter any number up to 100. Ideally, IDEC recommends that you use between 1 and 5 tag comment lines.
4. Basic Programming Instructions

Introduction to Basic Programming

As discussed in Chapter 2, PLC Overview, a PLC examines the status of input interfaces and, in response, controls something through output interfaces. In the following exercises, you will use the most basic programming instructions to create simple input/output programs.

Input Instructions

When programming your PLC, a load (LOD) or load not (LODN) instruction is used before an operand starting at the left bus of the ladder program. The LOD instruction starts the logical operation with a normally open (NO) contact. The LODN instruction starts the logical operation with a normally closed (NC) contact.

LOD/LODN Ladder Diagram
4-2 Basic Instructions

Output Instructions

The output (OUT) instruction outputs the result of a bit logical operation to the specified operand. The output not (OUTN) instruction outputs the inverted result of a bit logical operation to the specified operand.

OUT/OUTN Ladder Diagram

```
I0 ----> Q0
|     |
|     |
I1 ----> Q1
```
Exercise 1: Creating a Start/Stop Program

In Chapter 3, Getting Started, you opened WindLDR, selected your PLC type, set the appropriate function settings, and set your ladder preferences. You are now ready to create a ladder logic program using the blank program displayed on your screen.

In this exercise, you will write a basic ladder logic program that uses a PLC input to turn on/off an output. The figure below illustrates a simple input/output scenario in which an normally open (NO) pushbutton is used to turn on/off a pump. You will use the elements from this figure to create your program.

Start 1: NO alternate action pushbutton wired to I1

Pump A: Wired to Q0

After you create your program, you will transfer it to your PLC and monitor its operation.
Follow the steps below to create a start/stop program. When completed, your program screen should look like this.

**STEP 1: Enter the Start 1 Contact**

To enter the Start 1 contact:

1. Click the **Normally Open** icon.

Notice that as you move the pointer away from the icon, the icon symbol remains next to the pointer.
2. Click on Column 1 of Rung 1.

The Normally Open (LOD) dialog box displays.

You can move from field to field within a dialog box by pressing the [TAB] key or by clicking in the desired field.

3. Enter **Start 1** in the Tag Name: field of the dialog box.

4. Enter **I1** in the Allocation Number: field.

5. Enter **N.O. P/B is wired to I1** in the Tag Name Comment: field.

Your completed dialog box should look like this:

![Dialog Box Image]

6. Click the **OK** button or press the [ENTER] key to confirm.

The **Start 1** contact displays at the beginning of Rung 1.
Step 2: Enter the Pump A Output Coil

To enter the Pump A output coil:

1. Click the Output icon.
2. Click on Column 2 of Rung 1.

The Output (OUT) dialog box displays.

3. Enter Pump A in the Tag Name: field of the dialog box.
4. Enter Q0 in the Allocation Number: field.
5. Enter Pump A is wired to Q0 in the Tag Name Comment: field.

Your completed dialog box should look like this:

![Output Dialog Box](image)

**NOTE:** Output Not does not display in the Type: field of the ONC Output dialog box.

6. Click the OK button or press the [ENTER] key to confirm.

The Pump A output coil displays at the end of Rung 1.
Step 3: Enter a Rung Comment

To enter a rung comment:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar, or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter:
   This program shows a normally open contact I1 turning on an output Q0.

   IMPORTANT: WindLDR allows you to enter as many characters as desired. It is recommended that you keep to the length of the line when writing rung comments. You can use as many lines as you want. Press [ENTER] to transition to the next line.

4. Click OK to save the comments.

The rung comments display above Rung 1.
4-8 Basic Instructions

**Saving a Program**

To name and save this program:

1. Click **File ➔ Save as…** from the Menu Commands toolbar.
   
   The **Save As** dialog box displays.

2. Enter **START1** in the **File Name:** field of the dialog box.

3. Click the **OK** button or press the **[ENTER]** key to confirm.

**Ladder Preview**

To print preview this program:

1. Click **File ➔ Listings…** from the Menu Commands toolbar.
   
   The **Ladder Preview** dialog box displays.

2. Click your left mouse button or press the **[ENTER]** key to preview the whole ladder program.
Printing a Program

To print this program:

1. Click File ➔ Listing… from the Menu Commands toolbar.
   The Ladder dialog box displays. Select Ladder with left mouse
   button or hit [ENTER] key to print.

   The Print dialog box displays.

2. Enter the correct print options, and then click the OK button.
   or press the [ENTER] key to confirm.

Cross Reference

To cross reference the entire ladder program:

1. Click File ➔ Listings… from the Menu Commands toolbar.

   The Cross Reference dialog box displays.

2. Click your left mouse button or press the [ENTER] key to view
   the cross reference file.

NOTE: You may save or print the cross reference file which is presented in
the Notepad file format.
Closing a Program

To close a program:

1. Click the icon button in the upper left corner of the program screen.

   A drop down menu displays.

2. Click Close from the drop down menu.
Configuring the Communication Port

Before you can transfer your START1 program to the PLC, you must set the proper serial port settings. For this exercise, you will choose the default settings for your PLC.

To configure the communication port:

1. Click Configure ➔ Communication settings… from Menu Commands toolbar.

The Communication Settings dialog box displays asking you to confirm the communication settings.

2. Click the Default button.

WindLDR displays the programmed function settings for your PLC.

**NOTE:** Make sure that the Port: field is set to your computer's serial communication port.

3. Click the OK button or press the [ENTER] key to confirm.
Transferring a Program

After setting the communication ports, you are ready to transfer the START1 program to the PLC.

To transfer your program to the PLC:

1. Click **Compile ➔ Convert Ladder** from the Menu Commands toolbar.
2. Click **Online ➔ Download program…** from the Menu Commands toolbar.

If the program is not already compiled, a warning box displays asking you to recompile the ladder.

3. Click the **Yes** button.

A second dialog box displays stating that the conversion to mnemonics succeeded. If WindLDR encounters an error in your ladder program, an error message will display at this point.

4. Click the **OK** button.

The **Download Dialog** box displays.

5. Click the **OK** button or press the [ENTER] key to confirm.

If the download is successful, a dialog box displays confirming the download. If the download is unsuccessful, an error message dialog box displays.
**Monitoring a Program**

On-line monitoring allows you to monitor or view the programmed (ladder logic) instructions. Monitoring allows you to watch inputs and outputs turn on and off and to view the operation of timers, counters, and other operands.

To monitor your START1 program:

1. Click **Online ➔ Monitor** from the Menu Commands toolbar.
2. Notice that the toolbar icons disappear and the monitor screen displays.
3. If the input is powered ON (energized), a red highlight displays in both the NO contact and the output. If the input is OFF, the red highlight disappears from both the NO contact and the output.

**NOTE:** To exit the monitor screen, click **Online ➔ Monitor** again.
Online

The online option offers you many features which allow for powerful monitoring and manipulating of the program. These features are:

1. PLC Status

To monitor PLC status:

1. Click Online ➔ Monitor from the Menu Commands toolbar.
2. Click Online ➔ PLC Status from the Menu Commands toolbar.
2. Batch Monitor

Batch monitor allows you to monitor an array of inputs, outputs, timers, counters, data registers, and so forth by simply selecting the operand.

To monitor your programs in batches:

1. Click Online ➔ Monitor from the Menu Commands toolbar.

2. Click Online ➔ Batch Monitor from the Menu Commands toolbar.

3. Type the desired operand and starting address.

4. Scroll forward and backward by clicking the ← and → buttons.
3. Communication Error

To view a communication error:

1. Click Online ➔ Monitor from the Menu Commands toolbar.
2. Click Online ➔ Communication Error from the Menu Commands toolbar.
4. Point Write

To manipulate your program with the Point Write command:

1. Click Online ➔ Monitor from the Menu Commands toolbar.

2. Click Online ➔ Point Write from the Menu Commands toolbar.

3. Type the desired operand in the left box, and press the Tab key to display its current value.

4. Overwrite the current value by typing a new one into the right box on the same line. Press Enter.

In the above example, D0 is being changed from 0 to 500 and the preset value of T1 is being changed from 0 to 300.
NO and NC Conventions

It is important to understand the correlation between an external field device that is wired to a PLC, the device's operation conditions, and the PLC's input logic programming. The illustrations below show how you can program any switch wired to a PLC as either NO or NC and how this affects the (controlled) output.

External Switch is Open (OFF)

External Switch is Closed (ON)
The following figure illustrates the position of the external field device and its affect on the programmed ladder logic.

<table>
<thead>
<tr>
<th>Rung #</th>
<th>External Input Device</th>
<th>Programmed Logic</th>
<th>On/Off Status</th>
<th>Monitor Screen Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NO</td>
<td>OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(OFF) OPEN</td>
<td>NC</td>
<td>ON (CLOSED)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>NO</td>
<td>ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(ON) CLOSED</td>
<td>NC</td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

**NC Inputs**

In the first programming example, the input is a NO contact. When a NO contact is ON the output is also ON. Likewise, when the NO contact is OFF, the output is also OFF. In the following exercise, the E-Stop (emergency stop) is a NC external device. You must check to see if the E-Stop is off or if the NC contact becomes open.
Exercise 2: Creating a Multiple Start/Stop Program

In this exercise, you will add an additional LOD and OUT instruction and an E-Stop to the Start/Stop program from Exercise 1. The figure below illustrates a multiple input/output scenario in which either of two NO pushbuttons can be used to turn on/off two pumps simultaneously. An emergency NC pushbutton is used to simultaneously turn off both pumps. You will use the elements from this figure to create your program.

Start 1: NO alternate action pushbutton wired to I1
Start 2: NO alternate action pushbutton wired to I2
E-Stop: NC alternate action emergency pushbutton wired to I3
Pump A: Wired to Q0
Pump B: Wired to Q1
Follow the steps below to create a multiple Start/Stop program. When completed, your program screen should look like this:

**Step 1: Open the START1 Program from Exercise 1**
Step 2: Enter the E-Stop on Rung 1

To enter the E-Stop:

1. Click the Normally Closed \(\checkmark\) icon.

2. Click on Column 2 of Rung 1.

The Normally Closed (LODN) dialog box displays.

3. Enter E-Stop in the Tag Name: field of the dialog box.

4. Enter I3 in the Allocation Number: field.

5. Enter N.C. P/B is wired to I3 in the Tag Name Comment: field.

Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The E-Stop contact displays next to Start1 on Rung 1.
Step 3: Enter the Start 2 Contact on Line 2

To enter the Start 2 contact:

1. Click the Draw Line icon.

   Notice that the pointer changes from an arrow to a pencil.

2. Click the crossbar (+) to the right of the Start 1 contact on Rung 1, and drag the pointer down to Line 2 of Rung 1.

   A line connecting Line 1 with Line 2 on Rung 1 displays.

3. Click the Normally Open icon.

   Notice that the pointer returns to an arrow.

4. Click on the Column 1 of Line 2.

   The Normally Open (LOD) dialog box displays.

5. Enter Start 2 in the Tag Name: field of the dialog box.

6. Enter I2 in the Allocation Number: field.

7. Enter N.O. P/B is wired to I2 in the Tag Name Comment: field.
Your completed dialog box should look like this:

8. Click the **OK** button or press the **[ENTER]** key to confirm.

Notice that Rung 1 now has two lines. The *Start 2* contact displays at the beginning of Line 2.

**Step 4: Enter the Pump B Output Coil on Line 2**

To enter the Pump B output coil:

1. Click the **Draw Line** icon.

2. Click the crossbar to the left of the *Pump A* output coil on the Line 1, and drag the pointer down to the corresponding crossbar on Line 2.

A line connecting Line 1 with Line 2 displays.
3. Click the **Output** icon.

4. Click in the last column of Line 2.

The **Output** (OUT) dialog box displays.

5. Enter **Pump B** in the **Tag Name:** field of the dialog box.

6. Enter **Q1** in the **Allocation Number:** field of the dialog box.

7. Enter **Pump B is wired to Q1** in the **Tag Name** Comment: field.

Your completed dialog box should look like this:

![Output dialog box](image)

8. Click the **OK** button or press the [ENTER] key to confirm.

The **Pump B** output coil displays at the end of Line 2.
Step 5: Enter a Rung Comment

To enter a rung comment:

1. Click anywhere on Line 1 or 2.

2. Click Edit ➔ Edit Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Delete the existing comments in the dialog screen and enter:
   This program shows a Normally Open contact I1 turning on an Output Q0 of Pump A and Q1 of Pump B. Since the E-Stop Switch is an external NC contact, User must determine when it is off (NC contact is open).

4. Click OK to save the comments.

The rung comments display above Rung 1.
Step 6: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.

The Save As dialog box displays.

2. Enter **START2** in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 7: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
5. Timer Instructions

Introduction to Timers (TML, TIM, TMH, and TMS)

Timers are used for time-delay functions in PLC programs, just as relay timers are used in relay logic control systems. When power is placed to the control terminals, the timing circuit starts timing. When the preset time setting is reached, it allows current to flow to the coil of the relay, thus operating the controlled contacts.

Micro-1, FA1J/2J, FA3S-CP11/-CP11T Timers

These PLCs have 80 built-in timers (T0 – T79). All of the timers are 100msec timedown timers (TIM). Each timer must be programmed with an input to control the timer and may be programmed with or without an output after the timer. These timers can be used in on-delay or off-delay applications.

FA3S-CP12/13 Timers

The FA3S-CP12 and -CP13 CPUs have 256 timers, 100msec (T0 – T255) and 10msec (H0 – H79). Each timer must be programmed with an input to control the timer and may be programmed with or without an output after the timer.

NOTE: For details on high-speed timers, refer to the FA3S-CP12 User Manual.
## 5.2 Basic Instructions

### WindLDR™ Tutorial

<table>
<thead>
<tr>
<th>Micro-1 and FA Series Timer</th>
<th>Allocation Numbers</th>
<th>Range</th>
<th>Increase</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-1 TIM</td>
<td>T0 – T79</td>
<td>0 – 999.9 sec</td>
<td>100 msec</td>
<td>Constant: 0 – 9999</td>
</tr>
<tr>
<td>FA1J, FA2J, FA3S-CP11, FA3S-CP11T TIM</td>
<td>T0 – T79</td>
<td>0 – 999.9 sec</td>
<td>100 msec</td>
<td>Constant: 0 – 9999</td>
</tr>
<tr>
<td>FA3S-CP12</td>
<td>T0 – T255</td>
<td>0 – 999.9 sec</td>
<td>100 msec</td>
<td>Constant: 0 – 9999</td>
</tr>
<tr>
<td>FA3S-CP13</td>
<td>T10MS H0 – H79</td>
<td>0 – 99.99 sec</td>
<td>10 msec</td>
<td>Constant: 0 – 9999</td>
</tr>
</tbody>
</table>

**Micro-1 and FA Series Timer**

- **TIM**
- **CP11, CP11T**
- **CP12**
- **CP13**

**Allocation Numbers**

- **T0 – T79**
- **T0 – T255**
- **T10MS H0 – H79**

**Range**

- **0 – 999.9 sec**
- **0 – 999.9 sec**
- **0 – 99.99 sec**

**Increase**

- **100 msec**
- **100 msec**
- **10 msec**

**Preset Value**

- **Constant: 0 – 9999**
- **Constant: 0 – 9999**
- **Constant: 0 – 9999**
Micro\(^3\) and Micro\(^3\)C Timers

Three types of timers are available for the Micro\(^3\) and Micro\(^3\)C PLCs: a 100msec timedown timer (TIM), a 10msec timedown timer (TMH), and a 1msec time-
down timer (TMS). A total of 32 timers and counters can be programmed in the
standard processing mode. Each timer must be allocated to a unique number (0
through 31), and the same number cannot be used for counters. In the high-
speed processing mode, timer numbers 0 through 15 are available.

<table>
<thead>
<tr>
<th>Micro(^3) and Micro(^3)C, Timers</th>
<th>Allocation Numbers</th>
<th>Range</th>
<th>Increase</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIM (100msec timer)</td>
<td>T0 – T31</td>
<td>0 – 999.9 sec</td>
<td>100msec</td>
<td>Constant: 0 – 9999</td>
</tr>
<tr>
<td>TMH (10msec timer)</td>
<td>T0 – T31</td>
<td>0 – 99.99 sec</td>
<td>10msec</td>
<td>Data registers: D0 – D99 (standard mode)</td>
</tr>
<tr>
<td>TMS (1msec timer)</td>
<td>T0 – T31</td>
<td>0 – 9.999 sec</td>
<td>1msec</td>
<td>D0 to D31 (high-speed mode)</td>
</tr>
</tbody>
</table>
**ONC Timers**

There are four types of timers available for the ONC controllers:

- 1 second timer (TML)
- 100msec timer (TIM)
- 10msec timer (TMH)
- 1msec timer (TMS)

A total of 256 timers can be programmed in a user program. All timers are the timedown type. Each timer must be allocated to a unique number (T0 – T255).

**NOTE:** All ONC timers are 16-bit register timers. This allows a preset value of up to 65535.

<table>
<thead>
<tr>
<th>ONCTimer</th>
<th>Allocation Numbers</th>
<th>Range</th>
<th>Increase</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TML</td>
<td>T0 – T255</td>
<td>0 – 65535 sec</td>
<td>1 sec</td>
<td>Constant: 0 – 65535</td>
</tr>
<tr>
<td>TIM</td>
<td>T0 – T255</td>
<td>0 – 6553.5 sec</td>
<td>100msec</td>
<td>Data registers: D0 – D7999</td>
</tr>
<tr>
<td>TMH</td>
<td>T0 – T255</td>
<td>0 – 655.35 sec</td>
<td>10msec</td>
<td></td>
</tr>
<tr>
<td>TMS</td>
<td>T0 – T255</td>
<td>0 – 65.535 sec</td>
<td>1msec</td>
<td></td>
</tr>
</tbody>
</table>
**MicroSmart Timers**

There are four types of timers available for the MicroSmart:

- 1 second timer (TML)
- 100msec timer (TIM)
- 10msec timer (TMH)
- 1msec timer (TMS)

All-In-One type CPU module has a total of 32 timers for the 10 I/O type CPU, or 100 timers for the 16 I/O or 24 I/O type CPU. Slim Type CPU Module has a total of 100 timers for 20I/O type CPU or 40 I/O type CPU. This timers can be programmed in a user program. Each timer must be allocated to a unique number (0 – 255).

<table>
<thead>
<tr>
<th>MicroSmart Timer</th>
<th>All-In-One Type CPU</th>
<th>Slim Type CPU</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 I/O Type CPU</td>
<td>20 I/O Type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16 I/O Type CPU</td>
<td>24 I/O Type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24 I/O Type CPU</td>
<td>40 I/O Type</td>
<td></td>
</tr>
<tr>
<td>TML</td>
<td>T0 – T31</td>
<td>T0-T99</td>
<td>Constant: 0-65535</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data Registers:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 I/O Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-D0-D399</td>
</tr>
<tr>
<td>TIM</td>
<td>T0-T99</td>
<td>T0-T99</td>
<td>Other Types:</td>
</tr>
<tr>
<td>TMH</td>
<td>T0-T99</td>
<td>T0-T99</td>
<td>-D0-D1299</td>
</tr>
<tr>
<td>TMS</td>
<td>T0-T99</td>
<td>T0-T99</td>
<td>-D2000-D7999</td>
</tr>
</tbody>
</table>
5-6 Basic Instructions

Ladder Diagram (TIM)

Program List

<table>
<thead>
<tr>
<th>Prgm Ads</th>
<th>Instruction</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LOD</td>
<td>I0</td>
</tr>
<tr>
<td>1</td>
<td>TIM</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>LOD</td>
<td>I1</td>
</tr>
<tr>
<td>4</td>
<td>AND</td>
<td>T0</td>
</tr>
<tr>
<td>5</td>
<td>OUT</td>
<td>Q0</td>
</tr>
</tbody>
</table>

Timing Chart
Exercise 3: Creating an On-Delay Timer Program

In this exercise, you will write an on-delay timer program that uses a PLC input to turn on an output after a time delay.

The figure below illustrates an input/output scenario in which a normally open (NO) pushbutton is used to turn on a pump after a 10 second delay. You will use the elements from this figure to create your program.

Start 1: NO alternate action pushbutton wired to I1
Pump A: Wired to Q0
Follow the steps below to create an on-delay timer program. When completed, your program screen should look like this:

**Step 1: Click the New Icon to Open a New File**

A blank ladder logic program screen displays.
Step 2: Enter the Start 1 Contact

To enter the Start 1 contact:

1. Click the Normally Open icon.

   Notice that as you move the pointer away from the icon, the icon symbol remains next to the pointer.

2. Click on Column 1 of Rung 1.

   The Normally Open (LOD) dialog box displays.

   You can move from field to field within the dialog box by pressing the [TAB] key or by clicking in the desired field.

3. Enter Start 1 in the Tag Name: field of the dialog box.

4. Enter allocation number I1 in the Allocation Number: field.

5. Enter N.O. P/B is wired to I1 in the Tag Name Comment: field.
Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The Start 1 contact displays at the beginning of Rung 1.
Step 3: Enter the Timer

To enter the timer:

1. Click the Timer icon.

2. Click on Column 2 of Rung 1.

The 100 msec Timer (TIM) dialog box displays.

3. Enter T000 in the Tag Name: field of the dialog box.

4. Enter T0 in the Allocation Number: field.

5. Enter 100 in the Preset Value: field for a 10 second delay.

6. Enter 10-Sec Timer On-Delay in the Tag Name Comment: field.

Your completed dialog box should look like this:

7. Click the OK button or press the [ENTER] key to confirm.

The Timer instruction displays next to Start 1 on Rung 1.
Step 4: Enter a Comment for Rung 1

To enter a rung comment:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter:

   This program shows an On-Delay Timer for Pump A.
   The Pump will turn on 10 sec after Start 1 input is switched on.

   NOTE: WindLDR does not restrict the number of characters in the tag name editor. However, you are advised to stay below 10 characters per line, so that the width of the rung will be appropriate.

4. Click OK to save the comments.

The rung comment displays above Rung 1.
Step 5: Enter an NO Contact on Rung 2

To enter an NO contact:

1. Press the [ENTER] key to move to Rung 2.

2. Click the Normally Open icon.

3. Click on Column 1 of Rung 2.

The Normally Open (LOD) dialog box displays.

4. Enter T0 in the Tag Name: field of the dialog box.

   Notice that the properties for the Timer contact automatically display in the Allocation Number: and Tag Name Comment: fields.

5. Enter 10-Sec Timer on Delay in the Tag Name Comment: field.

   Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

   The Timer contact displays at the beginning of Rung 2.
5-14 Basic Instructions

Step 6: Enter the Pump A Output Coil on Rung 2

To enter the Pump A output coil:

1. Click the Output icon.

2. Click on Column 2 of Rung 2.

The Output (OUT) dialog box displays.

3. Enter Pump A in the Tag Name field of the dialog box.

4. Enter Q0 in the Allocation Number field.

5. Enter Pump A is wired to Q0 in the Tag Name Comment field.

Your completed dialog box should look like this:

NOTE: Output Not does not display in the Type field of the Micro-1 and FA series Output dialog box.

6. Click the OK button or press the [ENTER] key to confirm.

The Pump A output coil displays at the end of Rung 2.
Step 7: Save the Program

To name and save this program:

1. Click File → Save as… from the Menu Commands toolbar.

   The Save As dialog box displays.

2. Enter TIMERS in File Name: field of the dialog box.

3. Click OK or press [ENTER] to confirm.

Step 8: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.

Exercise 4: Timer Comparison Instruction
(TC= and TC≥)

**NOTE:** This option is only available with the ONC PLC. All four types of timers on the ONC PLC can be used with the Timer Comparison Instruction.

This program shows how the timer comparison instruction is used to turn on outputs. The preset value of the timer and timer comparison can be a constant (of values 0 to 65535) or a data register (D0 to D7999).
The TC= and TC≥ instructions are equivalent comparison instructions for timer current values. This instruction will constantly compare current values to the value that has been programmed in. When the timer value is greater than or equal to the given value, the desired output will be initiated.

Enter the value of the timers (from T0 to T255) followed by the preset value to compare (from 0 to 65535) or a data register to compare (from D0 to D7999).

**Ladder Diagram (TC=)**

```
<table>
<thead>
<tr>
<th>Prgm Adrs</th>
<th>Instruction</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TC=</td>
<td>T2</td>
</tr>
<tr>
<td>1</td>
<td>OUT</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Q0</td>
</tr>
</tbody>
</table>
```

**Program List**

**Ladder Diagram (TC=)**

```
<table>
<thead>
<tr>
<th>Prgm Adrs</th>
<th>Instruction</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TC≥</td>
<td>T3</td>
</tr>
<tr>
<td>1</td>
<td>OUT</td>
<td>D15</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Q1</td>
</tr>
</tbody>
</table>
```

© 2001 IDEC Corporation
Follow the steps below to create a timer comparison program. When completed your program screen should look like this:

Step 1: Click the New Icon to Open a New File

A blank ladder logic program screen displays.
Step 2: Enter the T000 Contact

To enter the T000 contact:

1. Click the **Normally Closed** icon.
2. Click on Column 1 of Rung 1.

The **Normally Closed** dialog box displays.

You can move from field to field within the dialog box by pressing the [TAB] key or by clicking in the desired field.

3. Enter **T000** in the **Tag Name** field of the dialog box.
4. Enter allocation number **T000** in the **Allocation Number** field.
5. Enter **1-Sec Timer from 10 - 0** in the **Tag Name Comment** field.

Your completed dialog box should look like this:

6. Click the **OK** button or press the [ENTER] key to confirm.

The **T000** contact displays at the beginning of Rung 1.
Step 3: Enter the Timer

To enter the timer:

1. Click the Timer icon.
2. Click on Column 3 of Rung 1.

The 1 sec Timer (TML) dialog box displays.

3. In the Type field, click the 1-sec Timer option.
4. Enter T000 in the Tag Name: field of the dialog box.
5. Enter T000 in the Allocation Number: field.
7. Enter 1-Sec Timer from 10 - 0 in the Tag Name Comment: field.

Your completed dialog box should look like this:

7. Click the OK button or press the [ENTER] key to confirm.

The T000 instruction displays next to T000 on Rung 1.
Step 4: Enter the Timer Comparison

To enter the timer comparison:

1. Click the Timer Comparison icon.
2. Click on column 1 of rung 2.

The Timer Comparison (TC=) dialog box displays.

3. In the Type field, click the Timer Comparison (TC=) option.
4. Enter T000 in the Tag Name: field of the dialog box.
5. Enter T000 in the Allocation Number: field.
6. Enter 0007 in the Preset Value: field for a 7 second comparison.
7. The Tag Name Comment: field automatically displays 1-Sec Timer from 10 - 0.

Your completed dialog box should look like this:

7. Click the OK button or press the [ENTER] key to confirm.
Step 5: Enter the Output Coil

To enter the output coil:

1. Click the Output icon.

2. Click on column 2 of rung 2.

The Output (OUT) dialog box displays.

3. Enter Q0002 in the Tag Name: field of the dialog box.

4. Enter Q0002 in the Allocation Number: field.

5. Enter When T0 = 7 Q2 is on in the Tag Name Comment: field.

Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The Q0002 output coil displays at the end of Rung 2.
Step 6: Enter the Timer Comparison

1. Click the Timer Comparison icon.
2. Click on the first column of rung 3.

The Timer Comparison (TC=) dialog box displays.

3. In the Type field, click the Timer Comparison (TC>=) option.
4. Enter T000 in the Tag Name: field of the dialog box.
5. Enter T000 in the Allocation Number: field.
6. Enter 0003 in the Preset Value: field for a 3 second comparison.
7. The Tag Name Comment: field automatically displays 1-Sec Timer from 10 - 0.

Your completed dialog box should look like this:

7. Click the OK button or press the [ENTER] key to confirm.

The TC= instruction displays at the beginning of rung 3.
Step 7: Enter the Output Coil to which to Compare

To enter the output coil:

1. Click the Output icon.

2. Click on column 2 of rung 3.

   The Output (OUT) dialog box displays.

3. Enter Q0003 in the Tag Name: field of the dialog box.

4. Enter Q0003 in the Allocation Number: field.

5. Enter When T0>= 3 Q3 is On in the Tag Name Comment: field.

   Your completed dialog box should look like this:

   ![Output dialog box image]

6. Click the OK button or press the [ENTER] key to confirm.

   The Q0003 output coil displays at the end of Rung 3.
Step 8: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.

The Save As dialog box displays.

2. Enter TIMCOMP in File Name: field of the dialog box.

3. Click OK or press [ENTER] to confirm.

Step 9: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
6. Counter Instructions

Introduction to Counters (CNT)

Counters, like the name implies, count pulses. These pulses are usually made in response to an object passing by a sensor that turns the input ON and then back OFF. Pulses can represent bottles, cans, parts, or the number of times a process operation has been performed. Counters can count up, down, or in some instances, in both directions. Applications for counters are almost unlimited.

IDEC PLCs have three types of counters: adding (up) counters, dual-pulse reversible counters, and up/down selection reversible counters.

**NOTE:** Reversible counters operate differently than adding counters. Adding counters start at 0 and turn ON when the preset value is reached. Reversible counters start at a preset value and turn ON when 0 is reached.

Micro-1, FA1J/2J, FA3S-CP11/-CP11T Counters

These PLCs have 45 adding (up) counters (CNT0–CNT44), one dual-pulse reversible counter (CNT45), and one up/down selection reversible counter (CNT46).

Micro-1, FA1J, FA2J, FA3S-CP11, and FA3S-CP11T Counter Table

<table>
<thead>
<tr>
<th>Type of Counter</th>
<th>Value</th>
<th>Preset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Up Counters</td>
<td>C0 – C44</td>
<td>0 – 9999</td>
</tr>
<tr>
<td>Dual Pulse Counters</td>
<td>C45</td>
<td>0 – 9999</td>
</tr>
<tr>
<td>Reversible Counters (Up/Down)</td>
<td>C46</td>
<td>0 – 9999</td>
</tr>
</tbody>
</table>
PF3S-CP12/-CP13 Counters

These high-performance PLCs have 100 adding up counters (CNT0 – CNT44, CNT48 – CNT102) and two reversible counters (CNT45 and CNT46).

FA3S-CP12, -CP13 Counter Table

<table>
<thead>
<tr>
<th>Type of Counter</th>
<th>Allocation Number</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Up Counters</td>
<td>C0 – C44</td>
<td>0 – 9999</td>
</tr>
<tr>
<td></td>
<td>C48 – C102</td>
<td></td>
</tr>
<tr>
<td>Dual Pulse Counters</td>
<td>C45</td>
<td>0 – 9999</td>
</tr>
<tr>
<td>Reversible Counters (Up/Down)</td>
<td>C46</td>
<td>0 – 9999</td>
</tr>
</tbody>
</table>

Micro³, Micro³C, ONC and MicroSmart Counters

The Micro³ and Micro³C PLCs have 30 adding (up) counters (CNT2 – CNT31), one dual-pulse reversible counter (CNT0), and one up/down selection reversible counter (CNT1). The ONC has 256 counters (CNT0 - CNT255). The MicroSmart has a total of 32 (10 point type) or 100 (other type) counters.
## Micro\(^3\) and Micro\(^3\)C Counter Table

<table>
<thead>
<tr>
<th>Type of Counter</th>
<th>Allocation Number</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Up Counters</td>
<td>C2 – C31 (standard mode)</td>
<td>0 – 9999</td>
</tr>
<tr>
<td></td>
<td>C2 – C15 (high-speed mode)</td>
<td>D0 – D99</td>
</tr>
<tr>
<td>Dual Pulse Counters</td>
<td>C0</td>
<td>D0 – D31</td>
</tr>
<tr>
<td>Reversible Counters (Up/Down)</td>
<td>C1</td>
<td></td>
</tr>
</tbody>
</table>

## ONC Counter Table

<table>
<thead>
<tr>
<th>Type of Counter</th>
<th>Allocation Number</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Up Counters</td>
<td>C0 – C255</td>
<td>0 – 65535</td>
</tr>
<tr>
<td>Dual Pulse Counters</td>
<td></td>
<td>D0 – D7999</td>
</tr>
<tr>
<td>Reversible Counters (Up/Down)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## MicroSmart Counter Table

<table>
<thead>
<tr>
<th>Type of Counter</th>
<th>Allocation Number</th>
<th>Preset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Up Counters</td>
<td>C0 – C31 (10-I/O type CPU)</td>
<td>0 – 65535 (10 I/O type only)</td>
</tr>
<tr>
<td>Dual Pulse Counters</td>
<td>C0 – C99 (20, 40 I/O type CPU)</td>
<td>D0 – D399 (Other CPU Modules)</td>
</tr>
<tr>
<td>Reversible Counters (Up/Down)</td>
<td>C0 – C99 (16-,24-I/O type CPU)</td>
<td>D0 – D1299 D2000-D7999</td>
</tr>
</tbody>
</table>
Adding (Up) Counters

An adding counter requires two inputs and may be programmed with or without an output.

Ladder Diagram of Adding Up Counter

The first line of the counter is the reset input. When the reset input is ON, the accumulated value of the counter resets to zero. The input must be OFF to allow the counter to operate. The second line is the pulse input. The counter will increment its accumulated value every time the pulse line makes the transition from OFF to ON.

**NOTE:** The reset line must be OFF to begin counting. The accumulated value of the counter only increments once each time the pulse line turns ON.

Dual-Pulse Type Reversible Counter

The dual-pulse counter has three lines of logic to control its operation. The top line is for the control of the counter's preset operation. When this line turns ON, the value assigned to the counter's preset is loaded, NOT 0. As with the up counters, this input must be in the OFF state for the counter to count. The sec-
ond line is for the UP pulses, and the third line is for the DOWN pulses. If both the up and down pulses turn ON simultaneously, the dual-pulse counter will not count. When the counted value reaches 0, the output will turn ON.

**Ladder Diagram of Reversible Counter**

**Up/Down Selection Type Reversible Counter**

The up/down selection counter looks similar to the dual-pulse counter, but there is a major difference. There is only one line for the input pulses. The third line of logic controls the direction of the count (up or down). When the up/down selection line is ON, the counter counts up; when it is off, the counter counts down. Again, the top line loads the counter’s preset value and must be OFF for the counter to operate. When the counted value reaches 0, the output will turn ON.
Introduction to Counter Comparisons
(CC= and CC>)

The CC= instruction is an equivalent comparison instruction for counted values. This instruction will constantly compare counted values to the value that has been programmed. When the counter value equals the given value, the desired output will be initiated.

The CC≥ instruction is an equal to or greater than comparison instruction for counted values. This instruction will constantly compare counted values to the value that has been programmed. When the counter value is equal to or greater than the given value, the desired output will be initiated.

Counter Comparisons

When a counter comparison instruction is programmed, two addresses are required. The circuit for a counter comparison instruction must be programmed in the following order: the CC= or CC≥ instruction, a counter number 0 through 255 (depending upon the PLC), followed by the preset value to compare from 0 to 65535 for ONC and MicroSmart and 0 to 9999 for the remaining PLCs.

The preset value can be designated using a decimal constant or a data register only. When a data register is used, the data of the data register becomes the preset value. If the data register designated as a counter comparison preset value holds a value over 9999 (Micro³ and Micro³C) or 65535 (ONC and MicroSmart), then a program execution error will result; next, error indicator ERR1 is lit, and special internal relay M304 (Micro³ and Micro³C) or M8004 (ONC and MicroSmart) turns on.
**NOTE:** A data register for the preset value of the counter is applicable to the Micro³, Micro³C, ONC and MicroSmart only.

**Ladder Diagram (CC=)**

Counter # to Compare

```
  =C2
  10
  Q0
```

Preset Value to Compare

**Ladder Diagram (CC≥)**

```
  ≥C3
  15
  Q1
```
Exercise 5: Creating a Counter Program

In this exercise, you will write a counter program that uses a counter and counter comparison instruction to switch on two outputs when the counter value reaches a predetermined level.

The figure below illustrates an input/output scenario in which an NO pushbutton is used to alternately turn on/off two pumps. You will use the elements from this figure to create your program.

Start 1: NO alternate action pushbutton wired to I1

Pump A: Wired to Q0

Pump B: Wired to Q1
Follow the steps beginning on the next page to create a counter/counter comparison program. When completed, your program screen should look like this.

**Step 1: Click the New Icon to Open a New File**

A blank ladder logic program screen displays.
Step 2: Enter a Reset Contact

To enter a reset contact:

1. Click the **Normally Open** icon.

   Notice that as you move the pointer away from the icon, the icon symbol remains next to the pointer.

2. Click on Column 1 of Rung 1.

   The **Normally Open** (LOD) dialog box displays.

   You can move from field to field within a dialog box by pressing the [TAB] key or by clicking in the desired field.

3. Enter **C000** in the **Tag Name:** field of the dialog box.

4. Enter **C000** in the **Allocation Number:** field.

5. Enter **Counter 3 Reset** in the **Tag Name Comment:** field.
Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The Counter contact displays at the beginning of Rung 1.

Step 3: Enter a Counter

To enter the counter:

1. Click the Counter icon.

2. Click on Column 2 of Rung 1.

The Up Counter (CNT) dialog box displays.

3. Enter C000 in the Tag Name: field of the dialog box.
Notice that the properties for the *Counter* instruction automatically display in the *Allocation Number:* and *Tag Name Comment:* fields.

4. Enter 3 in the *Preset Value:* field.

Your completed dialog box should look like this:

![Dialog box with 3 entered in Preset Value]

5. Click the **OK** button or press the [ENTER] key to confirm.

Notice that Rung 1 now has two lines. The *Counter* instruction displays next to the *Counter* contact.
Step 4: Enter the Start 2 Pulse Contact on Line 2

To enter the Start 2 pulse contact:

1. Click the Normally Open icon.
2. Click on Column 1 of Line 2.
   The Normally Open (LOD) dialog box displays.
3. Enter Start 2 in the Tag Name: field of the dialog box.
4. Enter I2 in the Allocation Number: field.
5. Enter N.O. P/B to count up in the Tag Name Comment: field.
   Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

   The Start 2 contact displays at the beginning of Line 2.
Step 5: Enter a Rung Comment

To enter a rung comment:

1. Click Edit → Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

2. Click in the white area of the dialog screen and enter:
   This is a counter program using comparison instructions to turn on Pump A and pump B wired to outputs Q0 and Q1.

   Starting with counter C0 with preset value of 3. When counter 0 receives 3 counts it resets itself.

3. Click OK to save the comments.

The rung comment displays above Rung 1.
Step 6: Enter a Counter Comparison on Rung 2

To enter a counter comparison:

1. Press the [ENTER] key to move to Rung 2.
2. Click the Counter Comparison icon.
3. Click on Column 1 of Rung 2.

The Counter Comparison (CC=) dialog box displays.

4. Enter C000 in the Tag Name: field of the dialog box.

   Notice that the properties for the Counter instruction automatically display in the Allocation Number: and Tag Name Comment: fields.

5. Enter preset value 1 in the Preset Value: field.

Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The counter comparison instruction displays at the beginning of Rung 2.
Step 7: Enter the Pump A Output Coil on Rung 2

To enter the Pump A output coil:

1. Click the Output icon.

2. Click on Column 2 of Rung 2.

The Output (OUT) dialog box displays.

3. Enter Pump A in the Tag Name: field.

4. Enter Q0 in Allocation Number: field of the dialog box.

5. Enter Pump A is Wired to Q0 in the Tag Name Comment: field.

Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The Pump A output coil displays at the end of Rung 2.
Steps 8 and 9: Enter a Counter Comparison and the Pump B Output Coil on Rung 3

Follow the instructions in Steps 6 and 7 to create a counter comparison and the Pump B output coil on Rung 3 using the following parameters:

For the counter comparison, enter **Counter** in the **Tag Name:** field and **2** in the **Preset Value:** field of the **Counter Comparison** dialog box.

For the Pump B output coil, enter **Pump B** in the **Tag Name:** field, **Q1** in the **Allocation Number:** field, and **Pump B is Wired to Q1** in the **Tag Name Comment:** field of the **Output** dialog box.

Step 10: Save the Program

To name and save this program:

1. Click **File ➔ Save as…** from the Menu Commands toolbar.

   The **Save As** dialog box displays.

2. Enter **COUNTERS** in File Name: field of the dialog box.

3. Click the **OK** button or press the [ENTER] key to confirm.
Step 11: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
7. Set/Reset Instructions

SET and RST (Reset)
The SET and RST instructions are used to set (on) or reset (off) outputs, internal relays, and shift register bits. The SET and RST instructions require one address which must be selected from the appropriate operand numbers. The same output can be set and reset many times within a program. SET and RST instructions operate in every scan while the input is on.

Ladder Diagram

Timing Chart

The SET command is a Latch command.
The RST command is an Unlatch command.
Exercise 6: Creating a Set/Reset Program

In Exercise 1, you learned how to turn on/off an output using the LOD and OUT instructions. In this exercise, you will use the SET and RST instructions to create a latch/unlatch program.

The figure below illustrates a latch/unlatch scenario in which an NO pushbutton is used to latch (SET) a pump. A NO pushbutton is used to unlatch (RST) the pump. You will use the elements from this figure to create your program.

**Latch:** N.O. alternate action pushbutton wired to I1

**Unlatch:** N.O. alternate action pushbutton wired to I2

**Pump A:** Wired to Q0
Follow the steps below to create a set/reset program. When completed, your program screen should look like this:

**Step 1: Click the New Icon to Open a New File**

A blank ladder logic program screen displays.
Step 2: Enter the Input Contact I1

To enter the input contact I1 contact:

1. Click the Normally Open icon.

   Notice that as you move the pointer away from the icon, the icon symbol remains next to the pointer.

2. Click on Column 1 of Rung 1.

   The Normally Open (LOD) dialog box displays.

   You can move from field to field within a dialog box by pressing the [TAB] key or by clicking in the desired field.

3. Enter I1 in the Tag Name: field of the dialog box.

4. Enter I1 in the Allocation Number: field.

5. Enter N.O. I1 “Latch” in the Tag Name Comment: field.
Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The \textit{II} contact displays at the beginning of Rung 1.
Step 3: Enter the Pump A Set Output Coil

To enter the set output coil:

1. Click the Set icon.

2. Click on Column 2 of Rung 1.

The Set (SET) dialog box displays.

3. Enter Q0 in the Tag Name: field of the dialog box.

4. Enter Q0 in the Allocation Number: field.

5. Enter Pump A is wired to Q0 in the Tag Name Comment: field.

Your completed dialog box should look like this:

6. Click the OK button or press the [ENTER] key to confirm.

The Pump A set output coil displays at the end of Rung 1.
Step 4: Enter a Comment for Rung 1

To enter a rung comment:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Rung comment… from the menu commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter:

   This program shows a normally open contact I1 latching an output Q0 and a normally open contact I2 unlatching it.

   When I1 comes on once or stays on, Q0 is set or latched.

4. Click OK to save the comments.

The rung comment displays above Rung 1.
Step 5: Enter the Stop 1 Contact on Rung 2

To enter the Stop 1 contact:

1. Click the **Normally Open** icon.
2. Click on Column 1 of Rung 2.

The **Normally Open (LOD)** dialog box displays.

3. Enter \textit{I2} in the **Tag Name**: field of the dialog box.
4. Enter \textit{I2} in the **Allocation Number**: field.
5. Enter \textit{N.O. I2 “Unlatch”} in the **Tag Name Comment**: field.

Your completed dialog box should look like this:

6. Click the **OK** button or press the [ENTER] key to confirm.

The \textit{I2} contact displays at the beginning of Rung 2.
Step 6: Enter the Pump A Reset Output Coil on Rung 2

To enter the reset output coil:

1. Click the Reset icon.

2. Click on Column 2 of Rung 2.

The Reset (RST) dialog box displays.

3. Enter Q0 in the Tag Name: field of the dialog box.

   Notice that the properties for the Pump A output coil automatically display in the Allocation Number: and Tag Name Comment: fields.

Your completed dialog box should look like this:

![Reset dialog box](image)

4. Click the OK button or press the [ENTER] key to confirm.

The Q0 reset output coil displays at the end of Rung 2.
Step 7: Enter a Comment for Rung 2

To enter a rung comment:

1. Click anywhere on Rung 2.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen, and enter: When I2 comes on Q0 will reset or unlatch.

4. Click OK to save the comments.

The rung comment displays above Rung 2.
Step 8: Save the Program

To name and save this program:

1. Click File → Save as… from the Menu Commands toolbar.

   The Save As dialog box displays.

2. Enter SETRSET in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 9: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
8. Single Output Instructions

Micro³, Micro³C, ONC, and MicroSmart Single Output Up (SOTU) and Down (SOTD) Instructions

For the Micro³, Micro³C, ONC, and MicroSmart, there are two SOT instructions: single output up (SOTU) and single output down (SOTD). The SOTU instruction is comparable to the SOT instruction for the remaining IDEC PLCs.

The SOTU instruction looks for the transition of a given input from off to on. The SOTD instruction looks for the transition of a given input from on to off. When this transition occurs, the desired output will turn on for the length of one scan. The SOTU or SOTD instruction converts an input signal to a one-shot pulse signal. The SOTU or SOTD instruction is followed by one address.

SOTU and SOTD instructions can be used repeatedly.

If an operation is started while the given input is already on, then the SOTU output will not turn on. The transition from off to on is what triggers the SOTU instruction.

NOTE: The scan time on some PLCs can be much faster than the time interval needed to operate a relay output.
The single output (SOT) instruction looks for the transition of a given input from off to on. When this transition occurs, the desired output will turn on for the length of one scan. The SOT instruction converts an input signal to a one-shot pulse signal, and is followed by one address.

The same SOT instruction cannot be used repeatedly.

If an operation is started while the given input is already on, then the SOT output will not turn on. The transition from off to on is what triggers the SOT instruction.

The SOT is comparable to the SOTU instruction for the Micro³, Micro³C, and ONC.
Exercise 7: Creating an SOTU/SOTD Program (Micro\textsuperscript{3}, Micro\textsuperscript{3}C, ONC, and MicroSmart)

Because most PLC instructions are executed with every scan, it is sometimes necessary to use single-shot commands.

As shown in Exercise 6, it is possible for both I\textsubscript{1} (latch) and I\textsubscript{2} (unlatch) to be on at the same time. In this exercise, you will add SOTU instructions (Micro\textsuperscript{3}, Micro\textsuperscript{3}C, and ONC) to Exercise 6 to ensure that the latch/unlatch commands are executed for one scan only. This minimizes conflicts in the execution of the instructions.

Ladder Diagram

- Single Shot when I\textsubscript{1} Switches from Off to On
- Single Shot when I\textsubscript{1} Switches from On to Off
The figure below illustrates a latch/unlatch scenario in which an NO pushbutton is used to issue a single-shot command (SOTU) to latch (SET) a pump. An NO pushbutton is used to issue an SOTU command to unlatch (RST) the pump. You will use the elements from this figure to create your program.

**Latch:** NO pushbutton wired to I1

**Unlatch:** Wired to I2

**Pump A:** Wired to Q0
Follow the steps below to create an SOTU program. When completed, your program screen should look like this:

**Step 1: Open the SETRSET Program from Exercise 6**
8-6 Basic Instructions

WindLDR™ Tutorial

Step 2: Enter an SOTU Instruction on Rung 1

To enter an SOTU instruction:

1. Click the SOTU icon.

2. Click on Column 2 of Rung 1.

The SOTU instruction displays next to the I1 contact on Rung 1.

Step 3: Enter an SOTD Instruction on Rung 2

Repeat the instructions in Step 2 to add an SOTD instruction on Rung 2.

Step 4: Enter a Comment for Rung 1

To enter a rung comment:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Edit Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.
3. Click in the white area below the existing text in the dialog screen and enter:
   
   **SOTU executes for one scan only when contact I1 goes from off to on.**

4. Click **OK** to save the comments.

   The rung comment displays above Rung 1.

---

**Step 5: Enter a Comment for Rung 2**

To enter a rung comment:

1. Click anywhere on Rung 2.

2. Click **Edit ➔ Edit Rung comment…** from the Menu Commands toolbar or click the **Rung comment** icon.

   The **Rung Comment Dialog** screen displays.

3. Click in the white area below the existing text in the dialog screen and enter:
   
   **SOTD executes for one scan only when contact I2 goes from on to off.**

4. Click **OK** to save the comments.

   The rung comment displays above Rung 2.
Step 6: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.

The Save As dialog box displays.

2. Enter SOTU in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 7: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.

Exercise 8: Creating an SOT Program (Micro-1 and FA Series PLCs)

Because most PLC instructions are executed with every scan, it is sometimes necessary to use single-shot commands.

As shown in Exercise 6, it is possible for both I1 (latch) and I2 (unlatch) to be on at the same time. In this exercise, you will add SOT instructions (Micro-1 and FA series) to Exercise 6 to ensure that the latch/unlatch commands are executed for one scan only. This minimizes conflicts in the execution of the instructions.
The Micro-1 and FA series PLCs have only one SOT type which executes for one scan when an normally open (NO) contact switches from off to on. If an SOT instruction is preceded by a normally closed (NC) contact, then SOT will execute for one scan when the NC contact switches from on to off.

The figure below illustrates a latch/unlatch scenario in which an NO pushbutton is used to issue a single-shot command (SOT) to latch (SET) a pump. An NO pushbutton is used to issue an SOT command to unlatch (RST) the pump. You will use the elements from this figure to create your program.

**Latch:** NO pushbutton wired to I1  
**Unlatch:** Wired to I2  
**Pump A:** Wired to Q0
Follow the steps below to create an SOT program. When completed, your program screen should look like this:

**Step 1: Open the SETRSET Program from Exercise 6**
Step 2: Enter an SOT Instruction on Rung 1

To enter an SOT instruction:

1. Click the SOT icon.

2. Click on Column 2 of Rung 1.

The SOT instruction displays next to the I1 contact on Rung 1.

Step 3: Enter an SOT Instruction on Rung 2

Change I2 on the second Rung from normally open to normally closed. To do so:

1. Double-click on contact I2.

2. Click on normally closed (NC) instead of normally open (NO)

Step 4: Enter an SOT Instruction on Rung 2

Repeat the instructions in Step 2 to add an SOT instruction on Rung 2.
Step 5: Enter a Comment for Rung 1

To enter a rung comment:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area below the existing text in the dialog screen and enter:
   SOT executes for one scan only when contact I1 goes from off to on.

4. Click the OK to save the comments.

The rung comment displays above Rung 1.
Step 6: Enter a Comment for Rung 2

To enter a rung comment:

1. Click anywhere on Rung 2.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area below the existing text in the dialog screen and enter:
   
   SOT executes for one scan only when contact I2 goes from on to off.

4. Click the OK to save the comments.

The rung comment displays above Rung 2.
Step 7: Save the Program

To name and save this program:

1. Click File → Save as… from the Menu Commands toolbar.

The Save As dialog box displays.

2. Enter SOT in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 8: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
9. Advanced Instructions: Introduction

**NOTE:** There is no move instruction for IDEC’s Micro-1 PLC.

**Data Move (Micro³, Micro³C, ONC and MicroSmart)**

Data of different forms can be moved in a PLC with the use of an instruction such as MOV (move), MOVN (move not), IMOV (indirect move), IMOVN (indirect move not), BMOV (batch move), or WMOV (word move). This data can be digital or bit operand, represented in 16- or 32-bit registers.

**Structure of an Advanced Instruction**

**Repeat Designation:** Specifies whether repeat is used for the operand or not.

**Repeat Cycles:** Specifies the quantity of repeat cycles: 1 – 31 (Micro³ and Micro³C) and 1 – 99 (ONC and Microsmart).

**Opcode:** The opcode is a symbol to identify the advanced instruction.
Source Operand: The source operand specifies the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data to be processed by the advanced instruction. Some advanced instructions require two source operands.

Destination Operand: The destination operand specifies the 16-bit word data to store the result of the advanced instruction. Some advanced instructions require two destination operands.

Input Condition for Advanced Instructions
Almost all advanced instructions must be preceded by a contact, except no operation (NOP) and high speed (HSC0 – HSC3) instructions. (The high speed mode is applicable for Micro³ and Micro³C PLCs only.) The input condition can be programmed using a bit operand such as an input, output, internal relay, or shift register. A timer or counter can also be used as an input condition to turn on the contact when the timer times out or the counter counts out.

While the input condition is on, the advanced instruction is executed in each scan. To execute the advanced instruction only at the rising or falling edge of the input, use the single output up (SOTU) or single output down (SOTD) instruction.

[Diagram of SOTU and advanced instruction]
**Source and Destination Operands**

The source and destination operands specify 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data. When a bit operand such as input, output, internal relay, or shift register is designated as a source or destination operand, 16 or 32 points starting with the designated number are processed as source or destination data. When a word operand such as a timer or counter is designated as a source operand, the current value is read as source data. When a timer or counter is designated as the destination operand, the result of the advanced instruction is set to the preset value for the timer or counter. When a data register is designated as a source or destination operand, the data is read from or written to the designated data register.

**Using a Timer or Counter as the Source Operand**

Since all timer instructions subtract from the preset value, the current value is decremented from the preset value and indicates the remaining time. As described above, when a timer is designated as the source operand of an advanced instruction, the current value (or the remaining time) of the timer is read as the source data.

Reversible counters start counting at the preset value, and the current value is incremented or decremented from the preset value. Adding counters start counting at 0, and the current value is incremented up to the preset value. When any counter is designated as the source operand of an advanced instruction, the current value is read as the source data.
Using a Timer or Counter as the Destination Operand

As described above, when a timer or counter is designated as the destination operand of an advanced instruction, the result of the advanced instruction is set to the preset value of the timer or counter. Since timer and counter preset values can be 0 – 9999 (Micro³ and Micro³C) or 0 – 65535 (ONC and MicroSmart) word data, make sure that the result of the advanced instruction does not exceed the corresponding value.

When a timer or counter preset value is designated using a data register, the timer or counter cannot be designated as the destination of an advanced instruction. When executing such an advanced instruction, a user program execution error will result. If a timer or counter is designated as the destination of an advanced instruction and if the timer or counter is not programmed, then a user program execution error will also result.

**NOTE:** When a user program execution error occurs, the result is not set to the destination.

Using an Input or Output as the Source or Destination Operand

When an input or output is designated as the source or destination operand of an advanced instruction, 16 or 32 points (starting with the designated number) are used. Depending on the micro PLC base unit used (alone or in the expansion link system) and depending upon the MicroSmart and ONC I/O modules used, the available input terminals are limited and special care is needed.
### Input Allocation for the Micro³, Micro³C, ONC and MicroSmart

<table>
<thead>
<tr>
<th>Model</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro³/³C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>With expansion</td>
<td>I0 - I35</td>
<td>Q0 – Q31</td>
</tr>
<tr>
<td>Alone</td>
<td>I0 – I15</td>
<td>Q0 – Q11</td>
</tr>
<tr>
<td>ONC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>With expansion</td>
<td>I0 – I597</td>
<td>Q0 – Q597</td>
</tr>
<tr>
<td>Alone</td>
<td>I0 – I277</td>
<td>Q0 – Q277</td>
</tr>
<tr>
<td>MicroSmart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All-In-One Type CPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alone</td>
<td>I0 – I15</td>
<td>Q0 – Q3</td>
</tr>
<tr>
<td>10-I/O Type CPU</td>
<td>I0 – I10</td>
<td>Q0 – Q6</td>
</tr>
<tr>
<td>16-I/O Type CPU</td>
<td>I0 – I115</td>
<td>Q0 – Q11</td>
</tr>
<tr>
<td>24-I/O Type CPU</td>
<td>I30 – I1107</td>
<td>Q30 – Q107</td>
</tr>
<tr>
<td>Slim Type CPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 I/O Type CPU</td>
<td>I0 - I13</td>
<td>Q0 - Q7</td>
</tr>
<tr>
<td>With Expansion</td>
<td>I30 - I1187</td>
<td>Q30 - Q187</td>
</tr>
<tr>
<td>20 I/O Type CPU</td>
<td>I0 - I13</td>
<td>Q0 - Q7</td>
</tr>
<tr>
<td>With Expansion</td>
<td>I30 - I1307</td>
<td>Q30 - Q307</td>
</tr>
<tr>
<td>40 I/O Type CPU</td>
<td>I0 - I27</td>
<td>Q0 - Q17</td>
</tr>
<tr>
<td>With Expansion</td>
<td>I30 - I1307</td>
<td>Q0 - Q307</td>
</tr>
</tbody>
</table>

Exercise 9: Creating a Data Move Program (Micro³, Micro³C, ONC and MicroSmart series)

In this exercise, you will use the data move instruction (MOV) to move different values into the preset of the counter. The preset values of the counter 6 and 12 represent the number of packages in a soda packaging plant. One pushbutton represents each package.

Complete the following steps to create a MOV program. When completed, your program screen should look like this:

NOTE: The MOV instruction can be used to move constants, data registers, inputs, outputs, timers, counters, and internal relays.
Step 1: Create the Basic Program Logic: Open Contacts, Counters, SOTUs, Tag Names, and Comments as Illustrated in Exercises 1–8.

1. Click the **New** icon to open a new file.

Step 2: Enter a MOV Instruction on Rung 2

To enter a MOV instruction:

1. Click the **MOV** icon.

2. Click on Column 3 of Rung 2.

The **Move** dialog box displays. The completed dialog box should look like this.

3. The **Move** option in the **Type**: field is selected by default. Select **Word** in the **Data Type**: field.
4. In both the Tag Name: and Allocation Number: fields, enter 6 for S1 and C003 for D1.

5. In the Tag Name Comments: field, enter 6 Bottles moved to C3 for S1 and Soda Packaging machine in D1.

6. Click OK or press the [ENTER] key to confirm that the MOV instruction displays at the end of Rung 2.

**Step 3: Enter a MOV Instruction on Rung 3**

To enter a MOV instruction:

1. Click the MOV icon.

2. Click on Column 3 of Rung 3.

The Move dialog box displays. The completed dialog box should look like this.

3. The Move option in the Type: field is selected by default. Select Word in the Data Type: field.
4. In both the Tag Name: and Allocation Number: fields, enter **12** for S1 and **C003** for D1.

5. In the Tag Name Comments: field, enter **12 Bottles moved to C3** for S1 and **Soda Packaging machine** in D1.

4. Click OK.

**Step 4: Save the Program**

To name and save this program:

1. Click **File ➔ Save as...** from the Menu Commands toolbar.

   The Save As dialog box displays.

2. Enter **MOVE** in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

**Step 5: Print, Transfer, and Monitor the Program**

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
Data Move (FA series)

**NOTE:** Different FA series PLCs may have different move instructions such as WMOV (word move) and WBMOV (word block move). Refer to your individual PLC’s instruction manual for details.

The WMOV (word move) and WBMOV (word block move) instructions are used for transferring word data to data registers. The WNSET (word “N” set) instruction is used for setting the data to be displayed or printed to data registers. The WNRS (word “N” repeat set) instruction is used for clearing a series of data registers. The data move instructions are executed in each scan while the input is on. Use a level input or pulse input as requires.

**Operand Designation**

isp S       Source including special internal relays  
ece S       Source excluding constants and external settings  
NW, c NW    Number of data blocks, such as data registers, in word (2 bytes)

When the input is on, 16-bit data from the operand designated by S is set to equal that of the operand designated by D.
Exercise 10: Creating a Data Move Program (FA series)

In this exercise, you will use the data move instruction (WMOV) to move the input register into D0. This instruction moves all 16-bit I0 – I7 and I10 – I17 into D0.

Follow the steps below to create a MOV program. When completed, your program screen should look like this:

NOTES: The MOV instruction can be used to move constants, data registers, inputs, outputs, timers, counters, and internal relays.

**Step 1: Create the Basic Program Logic: Open Contacts, Tag Names, and Comments as Illustrated in Exercises 1-8.**

1. Click the New icon to open a new file.
9-12 Advanced Instructions WindLDR™ Tutorial

Step 2: Enter a WMOV Instruction on Rung 1

To enter a WMOV instruction:

1. Click the **WMOV** icon.
2. Click on Column 2 of Rung 1.

The **Word Move** dialog box displays. The completed dialog box should look like this.

3. The **Word Move** option in the **Type:** field is selected by default.
4. In both the **Tag Name:** and **Allocation Number:** fields, enter **I0** for **ispS** and **D0** for **D**.
5. In the **Tag Name Comments:** field, enter **16-bit Input Register** for **ispS** and **Moved to D0** in **D**.
6. Click **OK**.
Step 3: Enter a WMOV Instruction on Rung 2

To enter a WMOV instruction:

1. Click the **WMOV** icon.
2. Click on Column 2 of Rung 2.

The **Word Move** dialog box displays. The completed dialog box should look like this.

3. The **Word Move** option in the **Type**: field is selected by default.
4. In both the **Tag Name**: and **Allocation Number**: fields, enter **D0** for **ispS** and **Q0** for **D**.
5. In the **Tag Name Comments**: field, enter **Moved to d0 for ispS** and **16-Bit Output Register** in **D**.
6. Click **OK**.
Step 4: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.

The Save As dialog box displays.

2. Enter WMOV in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 5: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
10. Arithmetic/Comparison Instructions

For Micro³, Micro³C, ONC, and MicroSmart Only

Binary Arithmetic Instructions

The binary arithmetic instructions make it possible for you to program computations using addition, subtraction, multiplication, and division. For addition and subtraction operands, internal relay M303 (Micro³ and Micro³C) or M8003 (ONC and MicroSmart) are used to carry or to borrow.

Addition (ADD)

```
ADD S1(R) S2(R) D1(R) REP
```

S1 + S2 → D1, CY

When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are added. The result is set to destination operand D1 and carry M303 (Micro³ or Micro³C) or M8003 (ONC or MicroSmart).
**Subtraction (SUB)**

```
| SUB | S1(R) | S2(R) | D1(R) | REP |
```

S1 – S2 → D1, BW

When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operand S2 is subtracted from the 16-bit word data designated by source operand S1. The result is set to destination operand D1 and borrow M303 (Micro³ or Micro³C) or M8003 (ONC or MicroSmart).

**Multiplication (MUL)**

```
| MUL | S1(R) | S2(R) | D1(R) | REP |
```

S1 x S2 → D1

When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operand S1 is multiplied by the 16-bit word data designated by source operand S2. The result is set to 32 Bit Data designated by destination operand D1.

When the result exceeds 65535, error indicator ERR1 and special internal relay M304 (Micro³ or Micro³C) or M8004 (ONC or MicroSmart) (user program execution error) are turned on.
Division (DIV)

\[
\text{DIV } S1(\text{R}) \quad S2(\text{R}) \quad D1(\text{R}) \quad \text{REP}
\]

\[ ***** \quad ***** \quad ***** \quad ** \]

S1 ÷ S2 → D1 (quotient), D1+1 (remainder)

When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or the 16- or 32-bit (ONC) word data designated by source operand S1 is divided by the 16-bit word data designated by source operand S2. The quotient is set to destination operand D1 and the remainder is set to D1+1.

When S2 is 0 (dividing by 0), error indicator ERR1 and special internal relay M304 (Micro³ or Micro³C) or M8004 (ONC) (user program execution error) are turned on.

Root (ROOT) — ONC and MicroSmart Only

\[
\text{ROOT(W)} \quad S1 \quad D1
\]

\[ ***** \quad ***** \]

Square Root of S1 → D1

When an input is on, the square root of the operand designated by S1 is extracted and is stored to the destination designated by D1.

Valid values are 0 to 65535. The square root is rounded to two decimals.

Since the ROOT instruction is executed in each scan while the input is on, a pulse from the SOTU or SOTD instruction should be used as required.
**Sum (SUM) — ONC Only**

The SUM instruction can be selected for ADD or XOR operations.

ADD: S1 to S2 added → D1•D1+1
XOR: S1 to S2 XORed → D1

When the input is on with ADD selected, all data of the operands designated by S1 to S2 are added. The result is stored to the destination operand designated by D1 and the next operand D1+1.

When the input is on with XOR selected, all data of the operands designated by S1 to S2 are XORed. The result is stored to the destination operand designated by D1.

When a timer (T) or counter (C) is used as S1 or S2, the current value of the timer/counter is read out.

Since the SUM instruction is executed in each scan while the input is on, a pulse input from the SOTU or SOTD instruction should be used as required.

**NOTE:** For further details about the ROOT and SUM instructions, refer to the ONC user’s manual.
## Valid Operands (Standard Processing)

<table>
<thead>
<tr>
<th>Operand</th>
<th>PLC</th>
<th>Function</th>
<th>CPU Type</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (Source 1)</td>
<td>Micro3 and Micro3C</td>
<td>Data for calculation</td>
<td>—</td>
<td>0–35</td>
<td>0–31</td>
<td>0–317</td>
<td>0–31</td>
<td>0–63</td>
<td>0–99/498</td>
<td>0–65535</td>
<td>1–31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ONC</td>
<td></td>
<td></td>
<td>0–597</td>
<td>0–597</td>
<td>0–2557</td>
<td>0–255</td>
<td>0–255</td>
<td>0–7999</td>
<td>0–4E9</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MicroSmart</td>
<td>10-I/O</td>
<td>0–5</td>
<td>0–3</td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>16-I/O</td>
<td>0–10</td>
<td>0–6</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>24-I/O</td>
<td>0–107</td>
<td>0–107</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20-I/O (FC4A)</td>
<td>D20K3/D2053</td>
<td>0–187</td>
<td>0–187</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D20RK1/D20RS1</td>
<td>0–307</td>
<td>0–307</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40 I/O</td>
<td>0–307</td>
<td>0–307</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td>S2 (Source 2)</td>
<td>Micro3 and Micro3C</td>
<td>Data for calculation</td>
<td>—</td>
<td>0–35</td>
<td>0–31</td>
<td>0–317</td>
<td>0–31</td>
<td>0–63</td>
<td>0–99/498</td>
<td>0–65535</td>
<td>1–31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ONC</td>
<td></td>
<td></td>
<td>0–597</td>
<td>0–597</td>
<td>0–2557</td>
<td>0–255</td>
<td>0–255</td>
<td>0–7999</td>
<td>0–4E9</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MicroSmart</td>
<td>10-I/O</td>
<td>0–5</td>
<td>0–3</td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>16-I/O</td>
<td>0–10</td>
<td>0–6</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>24-I/O</td>
<td>0–107</td>
<td>0–107</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20-I/O (FC4A)</td>
<td>D20K3/D2053</td>
<td>0–187</td>
<td>0–187</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D20RK1/D20RS1</td>
<td>0–307</td>
<td>0–307</td>
<td>0–127</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
</tbody>
</table>
### 10-6 Advanced Instructions

<table>
<thead>
<tr>
<th>Operand</th>
<th>PLC</th>
<th>Function</th>
<th>CPU Type</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 (Destination 1)</td>
<td>Micro(^3) and Micro(^3)C</td>
<td>—</td>
<td>—</td>
<td>0–31</td>
<td>0–287</td>
<td>0–31</td>
<td>0–63</td>
<td>0–99/498</td>
<td>—</td>
<td>1–31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>O NC</td>
<td>—</td>
<td>—</td>
<td>0–597</td>
<td>0–2557</td>
<td>0–255</td>
<td>0–255</td>
<td>0–7999</td>
<td>—</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>10-I/O</td>
<td>—</td>
<td>0–3</td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>—</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>16-I/O</td>
<td>—</td>
<td>0–6</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>—</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>24-I/O</td>
<td>—</td>
<td>0–107</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>—</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>20-I/O</td>
<td>D20KI/D20K3</td>
<td>0–187</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>—</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>40-I/O</td>
<td>—</td>
<td>0–307</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>—</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Micro\(^3\) = 0–99. Micro\(^3\)C = 0–498.*
Comparison Instructions (Micro³, Micro³C, ONC and MicroSmart only)

Data can be compared using comparison instructions such as equal to, not equal to, less than, greater than, less than or equal to, and greater than or equal to. When the comparison result is true, an output or internal relay is turned on. The repeat operation can also be used to compare more than one set of data.

Since the comparison instructions are executed with each scan while an input is on, a pulse input from an SOTU or SOTD instruction should be used as required.

**Compare Equal To (CMP=)**

S1 = S2 → D1 on
When an input is on, the 16-bit (Micro³, Micro³C, and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are compared. When S1 data is equal to S2 data, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.
**Compare Not Equal To (CMP≠)**

\[
\begin{array}{c}
\text{CMP≠} \quad \text{S1}(R) \quad \text{S2}(R) \quad \text{D1}(R) \quad \text{REP} \\
\text{*****} \quad \text{*****} \quad \text{*****} \quad \text{**}
\end{array}
\]

S1 ≠ S2 → D1 on
When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are compared. When S1 data is not equal to S2 data, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.

**Compare Less Than (CMP<)**

\[
\begin{array}{c}
\text{CMP<} \quad \text{S1}(R) \quad \text{S2}(R) \quad \text{D1}(R) \quad \text{REP} \\
\text{*****} \quad \text{*****} \quad \text{*****} \quad \text{**}
\end{array}
\]

S1 < S2 → D1 on
When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are compared. When S1 data is less than S2 data, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.
**WindLDR™ Tutorial**

**Advanced Instructions 10-9**

---

**Compare Greater Than (CMP>)**

```
CMP> S1(R) S2(R) D1(R) REP
***** ***** ***** **
```

S1 > S2 → D1 on
When an input is on, the 16-bit (Micro3, Micro3C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are compared. When S1 data is greater than S2 data, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.

---

**Compare Less Than or Equal To (CMP≤)**

```
CMP≤ S1(R) S2(R) D1(R) REP
***** ***** ***** **
```

S1 ≤ S2 → D1 on
When an input is on, the 16-bit (Micro3, Micro3C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are compared. When S1 data is less than or equal to S2 data, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.

---

© 2001 IDEC Corporation
**Compare Greater Than or Equal To (CMP≥)**

\[
\text{CMP≥ } S1(R) \quad S2(R) \quad D1(R) \quad \text{REP}
\]

S1 ≥ S2 → D1 on
When an input is on, the 16-bit (Micro³, Micro³C and MicroSmart) or 16- or 32-bit (ONC) word data designated by source operands S1 and S2 are compared. When S1 data is greater than or equal to S2 data, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.

**Interval Compare Greater Than or Equal to (ICMP≥)**

— ONC and MicroSmart Only

\[
\text{ICMP≥ } S1 \quad S2 \quad S3 \quad D1 \quad \text{REP}
\]

S1 ≥ S2 ≥ S3 → D1
When an input is on, the 16- or 32-bit (ONC) word data designated by source operands S1, S2, and S3 are compared. When the condition is met, the destination operand D1 is turned on. When the condition is not met, D1 is turned off.
## Valid Operands (Standard Processing)

<table>
<thead>
<tr>
<th>Operand</th>
<th>PLC Function</th>
<th>CPU Type</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (Source 1)</td>
<td>Micro3 and Micro3C</td>
<td>—</td>
<td>0–35</td>
<td>0–31</td>
<td>0–317</td>
<td>0–31</td>
<td>0–63</td>
<td>0–99</td>
<td>0–65535</td>
<td>1–31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ONC</td>
<td></td>
<td>0–597</td>
<td>0–597</td>
<td>0–2557</td>
<td>0–255</td>
<td>0–255</td>
<td>0–7999</td>
<td>0–4E9</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data for calculation</td>
<td></td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10-I/O</td>
<td></td>
<td>0–5</td>
<td>0–3</td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td>16-I/O</td>
<td></td>
<td>0–10</td>
<td>0–6</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td>24-I/O</td>
<td></td>
<td>0–15</td>
<td>0–107</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td>20-I/O (FC4A)</td>
<td>D20K3/D2053</td>
<td>0–187</td>
<td>0–187</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td>D20RKI/D20RS1</td>
<td>0–307</td>
<td>0–307</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 I/O</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td></td>
</tr>
<tr>
<td>S2 (Source 2)</td>
<td>Micro3 and Micro3C</td>
<td>0–35</td>
<td>0–31</td>
<td>0–317</td>
<td>0–31</td>
<td>0–63</td>
<td>0–99</td>
<td>0–65535</td>
<td>1–31</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ONC</td>
<td>0–597</td>
<td>0–597</td>
<td>0–2557</td>
<td>0–255</td>
<td>0–255</td>
<td>0–7999</td>
<td>0–4E9</td>
<td>1–99</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data for calculation</td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10-I/O</td>
<td>0–5</td>
<td>0–3</td>
<td>0–317</td>
<td>0–31</td>
<td>0–31</td>
<td>0–63</td>
<td>0–399</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16-I/O</td>
<td>0–10</td>
<td>0–6</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24-I/O</td>
<td>0–15</td>
<td>0–107</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20-I/O (FC4A)</td>
<td>D20K3/D2053</td>
<td>0–187</td>
<td>0–187</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
</tr>
<tr>
<td></td>
<td>D20RKI/D20RS1</td>
<td>0–307</td>
<td>0–307</td>
<td>0–1277</td>
<td>0–99</td>
<td>0–99</td>
<td>0–127</td>
<td>0–1299</td>
<td>0–65535</td>
<td>1–99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 I/O</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td>0–307</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 11: Creating a Binary Arithmetic/Comparison Program

The binary arithmetic ADD, SUB, MUL, and DIV can be used to manipulate data in the PLC and use the comparison instruction to execute a specific task.
Complete the following steps to create an arithmetic/comparison program.

When completed, your program screen should look like this:
Step 1: Create the Basic Program Logic: Open and Closed Contacts, Tag Names, and Comments as Illustrated in Exercises 1–8.

1. Click the New icon to open a new file.

Step 2: Enter an Upward Counting Counter on Rung 1

To enter an upward counter:

1. Click the Counter icon.

2. Click on Column 2 of Rung 1.

The Up Counter dialog box displays. The completed dialog box should look like this.

3. Click the Up Counter option in the Type: field.

4. Enter C4 in both the Tag Name: and Allocation Number: fields.

5. Enter 1000 in the Preset Value: field.

6. Enter C4 Counts Parts in the Tag Name Comment: field.

7. Click OK.
Step 3: Enter an Addition Instruction on Rung 2

To enter an addition instruction:

1. Click the Binary Arithmetic icon.

2. Click on Column 2 of Rung 2.

The Addition dialog box displays. The completed dialog box should look like this.

3. Click the Addition option in the Type: field.

4. Select Word from the Data Type: field.

5. Enter C4, C4, and C4 Counts Parts in the S1 column.

6. Enter 3000, 3000, and Constant # added to C4 in the S2 column.

7. Enter D10, D10, and D10 Stores C4 + 3000 in the D1 column.

8. Click OK.
Step 4: Enter a Multiplication instruction on Rung 3

To enter a multiplication instruction:

1. Click the **Binary Arithmetic** icon.

2. Click on Column 2 of Rung 3.

The **Binary Arithmetic** dialog box displays. The completed dialog box should look like this.

3. Click the **Multiplication** option in the **Type:** field.

4. Select **Word** from the **Data Type:** field.

5. Enter **D10, D10, and D10 Stores C4 + 3000** in the **S1** column.

6. Enter **50, 50, and 50 * D10 each scan** in the **S2** column.

7. Enter **D20, D20, and Result of D10 * 50** in the **D1** column.

8. Click **OK**.
Step 5: Enter a Comparison Instruction on Rung 4

To enter a comparison instruction:

1. Click the Comparison icon.

2. Click on Column 2 of Rung 4.

The Compare Equal To dialog box displays. The completed dialog box should look like this.

3. Click the CMP= option in the Type: field.

4. Select Word from the Data Type: field.

5. Enter D20, D20, and Result of D10 * 50 in the S1 column.

6. Enter 17500, 17500, and A critical set value in the S2 column.

7. Enter Q2, Q2, and Alarm On Wired to Q2 in the D1 column.

8. Click OK.
Step 6: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the menu commands toolbar.

The Save As dialog box displays.

2. Enter ARITHM in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 7: Print, Transfer, and Monitor the Program

To print this program, refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
Binary Arithmetic Instructions: FA Series PLCs

The binary arithmetic instructions make it possible for you to program computations using addition, subtraction, multiplication, and division. For addition and subtraction operations, internal relay M307 (707) is used to carry or borrow.

**Add with Carry (JADDC)**

D0 + operand + CY $\rightarrow$ D0, CY

When an input is on, the data (S) and the carry (M307) are added. The result is set to D0 and the carry (M307). Prior to this instruction, the data to be added must be set to D0 and the carry must also be reset as required.

**NOTE:** When the sum, including the carry, is 20,000 or more, a numeric value error will occur.

### Computing Example

<table>
<thead>
<tr>
<th>D0 +</th>
<th>Operand +</th>
<th>Carry</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 +</td>
<td>900</td>
<td>0 $\rightarrow$ 0</td>
<td>901</td>
</tr>
<tr>
<td>0 +</td>
<td>2000 +</td>
<td>1 $\rightarrow$ 0</td>
<td>2001</td>
</tr>
<tr>
<td>1000 +</td>
<td>9000 +</td>
<td>0 $\rightarrow$ 1</td>
<td>0000</td>
</tr>
<tr>
<td>9999 +</td>
<td>9999 +</td>
<td>1 $\rightarrow$ 1</td>
<td>9999</td>
</tr>
</tbody>
</table>
Normal Range of Sum

<table>
<thead>
<tr>
<th></th>
<th>D0</th>
<th>Operand</th>
<th>Carry</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0</td>
<td>0</td>
<td>0 ➔ 0</td>
<td>0</td>
</tr>
<tr>
<td>Maximum</td>
<td>9999</td>
<td>9999</td>
<td>1 ➔ 1</td>
<td>9999</td>
</tr>
</tbody>
</table>

Subtraction with Borrow (JSUBB)

D0 – operand – BY ➔ D0, BW
When an input is on, the data (S) is subtracted from the data of D0. The borrow (M307) is also subtracted from D0 and the result is set to D0 and borrow (M307). Prior to this instruction, the data from which to be subtracted must be set to D0 and the borrow must also be reset as required.

A negative value is indicated as its positive complement with respect to 10,000. To determine the absolute value of a negative value, subtract the positive complement from zero.

**NOTE:** When the result is less than 10,000 or greater than 9999, a numeric value error will occur.
**Multiplication (JMUL)**

D0 x operand → D1, D0;
D1 = upper 4 digits,
D0 = lower 4 digits

When an input is on, the data of D0 is multiplied by the data designated by S. The result is set to D0 and D1. Prior to this instruction, multiplication must be set to D0.

**NOTE:** When a multiplier or multiplicand exceeds 9999, a numeric value error occurs.

<table>
<thead>
<tr>
<th>Symbol: JMUL</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Normal Range of Product</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D0</strong></td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
</tr>
</tbody>
</table>

When the result is less than 10,000 (or 2710 in hexadecimal), it is set to D0 and zero is set to D1. When the result is more than 9999 (or 270F in hexadecimal), it is set to D1 as upper digits.
**10-22 Advanced Instructions**

**WindLDR™ Tutorial**

**Division (JDIV)**

D0 ÷ operand → D1, D0;
D1 = remainder, D0 = quotient

When an input is on, the data of D0 is divided by the data designated by S. The quotient is set to D0 and the remainder is set to D1. Prior to this instruction, the data must be set to D0 as a dividend.

**NOTE:** When the divisor is 0 or when the divisor or dividend exceeds 9999, a numeric value error occurs.

**NOTE:** When D0 is smaller than the operand, the quotient equals zero and the original data of D0 becomes the remainder.

**Normal Range of Quotient**

Quotient D0 = zero to 9999

Remainder D1 = zero to (divisor 1)
10,000 Addition 3 Operand (TADD3)

\[ S_1 + S_2 \rightarrow D, CY \]

When an input is on, the data designated by \( S_1 \) and \( S_2 \) are added. The result is set to the operand designated by \( D \) and the carry (M307).

Symbol: TADD3 \( S_1 \) \( S_2 \) \( D \)

10,000 Subtraction 3 Operand (TSUB3)

\[ S_1 - S_2 \rightarrow D, BY \]

When an input is on, the data designated by \( S_2 \) is subtracted from the data designated by \( D_1 \). The result is set to the operand designated by \( D \) and the borrow (M307).

Symbol: TSUB3 \( S_1 \) \( S_2 \) \( D \)

10,000 Multiplication (TMUL)

\[ D \times S \rightarrow (D + 1) \text{ (upper digits), } D \text{ (lower digits)} \]

When an input is on, the data designated by \( D \) is multiplied by the data designated by \( S \). The result is set to \( D \) and \( D+1 \).

Symbol: TMUL \( S \) \( D \)
**Comparison Instructions: FA Series PLCs**

**Word Greater Than or Equal To (WGE)**

D = S \(\rightarrow\) D (quotient),
D + 1 (remainder)

When an input is on, the data designated by D is divided by the data designated by S. The quotient is set to D and the remainder is set to D+1. FUN30 and FUN31 are used together for setting timer/counter preset values or other operands using external digital switches.

\[ D \div S \rightarrow D \text{ (quotient)}, \]
\[ D + 1 \text{ (remainder)} \]

\[ D = S \rightarrow D \]

\[ D + 1 \]

When an input is on, the data designated by D is divided by the data designated by S. The quotient is set to D and the remainder is set to D+1. The WGE instruction identifies a greater than or equal to relationship. Both data to be compared can be designated, and the result can be an output or an internal relay.

When an input is on, the data designated by S1 and S2 are compared. If the condition is met, the output or internal relay designated by oir goes on. If not, it goes off.
**NOTE:** Comparison instructions are executed in each scan while the input is on. So, use a level or pulse input as required. To execute the instruction only at the leading edge of the input, use an SOT instruction.

For data comparison instructions, if the condition is met when the input is on, the output designated by oir goes on. If the conditions are not met, the output goes off. When the input is off, the existing output status is maintained. For example, if the input goes off while the output is on, the output remains on.

**Word Less Than or Equal To (WLE)**

\[ S_1 \leq S_2 \rightarrow \text{oir set on} \]

The WLE instruction identifies a less than or equal to relationship. Both data to be compared can be designated, and the result can be an output or an internal relay.

When an input is on, the data designated by S1 and S2 are compared. If the condition is met, the output or internal relay designated by oir goes on. If not, it goes off.
**Word Equal To (WEQ)**

The WEQ instruction identifies an equivalent relationship. Both data to be compared can be designated and the result can be an output or an internal relay.

When an input is on, the data designated by S1 and S2 are compared. If the condition is met, the output or internal relay designated by oir goes on. If not, it goes off.

\[ S_1 = S_2 \rightarrow \text{oir set on} \]

The WEQ instruction identifies an equivalent relationship. Both data to be compared can be designated and the result can be an output or an internal relay.

When an input is on, the data designated by S1 and S2 are compared. If the condition is met, the output or internal relay designated by oir goes on. If not, it goes off.
Exercise 12: Creating an FA Series Advanced Instruction Program

The binary arithmetic ADD, SUB, MUL, and DIV can be used to manipulate data in the FA series PLC and use the comparison instruction to execute a specific task.

Complete the following steps to create an advanced instruction program for the FA series PLC. When completed, your program screen should look like this:
Step 1: Create the Basic Program Logic: Open and Closed Contacts, Tag Names, and Comments as Illustrated in Exercises 1–8.

1. Click the New icon to open a new file.

Step 2: Enter a 100-msec Timer on Rung 1

To enter a 100-msec timer:

1. Click the Timer icon.

2. Click on Column 2 of Rung 1.

The 100-msec Timer dialog box displays. The completed dialog box should look like this.

3. In both the Tag Name: and Allocation Number: fields, enter T0.

4. Enter 1800 in the Preset Value: field.

5. Enter 3 Min. Timer Auto. Reset in the Tag Name Comment: field.

6. Click OK.
Step 3: Enter an Addition Operand Instruction on Rung 2

To enter an addition operand instruction:

1. Click on Column 2 of Rung 2.

2. Type TAdd on your keyboard.

The 10,000 Addition 3 Operand dialog box displays. The completed dialog box should look like this.

3. Click the 10,000 Addition 3 Operand option in the Type: field.

4. Enter T0, T0, and 3 Min. Timer Auto. Reset in the S1 column.

5. Enter 500, 500, and constant value = 500 in the S2 column.

6. Enter D20, D20, and T0 + 500 into D20 in the D column.

7. Click OK.
Step 4: Enter a Comparison Instruction on Rung 3

To enter a comparison instruction:

1. Click the Word Comparison icon.

2. Click on Column 2 of Rung 3.

The Word Greater Than or Equal To dialog box displays. The completed dialog box should look like this.

3. Click the Word Greater Than or Equal option in the Type field.

4. Enter D20, D20, and T0 + 500 into D20 in the S1 column.

5. Enter 1000, 1000, and Constant value 1000 in the S2 column.

6. Enter Q1, Q1, and Q1 comes On if D10 = 1000 in the oir column.
Step 5: Enter a Word Move Instruction on Rung 4

To enter a word move instruction:

1. Click the Word Move icon.
2. Click on Column 2 of Rung 4.

The Word Move dialog box displays. The completed dialog box should look like this.

3. Enter T0, T0, and 3 Min. Timer Auto. Reset in the ispS column.
4. Enter D10, D10, and Store Current T0 Value in the D column.
Step 6: Save the Program

To name and save this program:

1. Click File → Save as… from the menu commands toolbar.

   The Save As dialog box displays.

2. Enter FA-ARITH in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 7: Print, Transfer, and Monitor the Program

To print this program, refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
11. Analog Read Instructions

Introduction to Analog Read Instructions

The Micro³, Micro³C, and MicroSmart PLCs have analog potentiometers in the front panel of the unit. The Micro³C, 10-I/O Micro³, and MicroSmart 10-, 16-20- and 40 I/O type CPU have one potentiometer (ANR0 for Micro³ and Micro³C and D8057 for MicroSmart). The 16- and 24-I/O Micro³, and 24-I/O MicroSmart PLCs have two potentiometers (ANR0 and ANR1 for Micro3, and D8057 and D8058 for MicroSmart). The potentiometer(s) can be addressed to any memory location in the PLC for purposes such as changing timer or counter presets. The 0 – 249 range of the potentiometer can be scaled with mathematical instructions.

**Analog Read 0 (ANR0)**

When an input is on, the value (0 – 249) set with analog potentiometer 0 is read to the data register designated as the destination. This instruction is useful for adjusting preset values of the timer (TIM, TMH, and TMS) and pulse (PULS and PWM) instructions.
11-2 Advanced Instructions

Analog Read 1 (ANR1)

When an input is on, the value (0 – 249) set with analog potentiometer 1 is read to the data register designated as the destination. This instruction is useful for adjusting the preset values of the timer (TIM, TMH, and TMS) and pulse (PULS and PWM) instructions.

Valid Operands (Standard Processing)

In the high-speed processing mode, data registers for these instructions are limited to D0 – D31. Since the ANR0 and ANR1 instructions are executed in each scan while the input is on, a pulse input from an SOTU or SOTD instruction should be used as required.

<table>
<thead>
<tr>
<th>Operand Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination to store the analog potentiometer value</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0 – 99 (Micro°C)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 498 (Micro°C)</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Analog Potentiometer Setting for Micro³ and Micro³C

The analog potentiometer positions and set values are shown below:

- Turned Fully to the Left: Minimum Value = 0
- Turned Fully to the Right: Maximum Value = 249

- The 10-I/O type Micro³ has one potentiometer.
- The 16- and 24-I/O Micro³ types have two potentiometers: analog potentiometer 0 on the left and analog potentiometer 1 on the right.
- The 16- and 24-I/O Micro³C types have only one potentiometer: analog potentiometer 0 on the left.

Example: ANR0

When input I0 is on, the value from analog potentiometer 0 is read to data register D80.
Exercise 13: Creating an Analog Read Program

In this exercise, you will use the PLC’s built-in potentiometer 0 as an external set-point device to determine the maximum volume of liquid allowed in a tank. The ANR0 instruction lets you read the PLC potentiometer value and store it in a data register to be used for comparison/data manipulation.

The potentiometer value varies on a scale from 0 to 249. When D10 is greater or equal to 100, output Q2 will come on. Otherwise, Q2 will remain off.

Pump A: Wired to Q0
Follow the steps below to create an analog read program. When completed, your program screen should look like this:

**Step 1: Create the Basic Program Logic: Open Contacts, Tag Names, and Comments as Illustrated in Exercises 1-8.**

1. Click the **New** icon to open a new file.
Step 2: Enter the Analog Read Instruction on Rung 1

To enter the analog read instruction:

1. Click the Analog Read icon.

2. Click on Column 2 of Rung 1.

The Analog Read 0 (ANR0) dialog box displays.

3. Enter D10 in both the Tag Name: and Allocation Number: fields.

4. Enter D10 value = 0-249 in the Tag Name Comment: field.

5. Click OK.

The analog read instruction displays at the end of Rung 1.
Step 3: Enter the Comparison Instruction on Rung 2

To enter the comparison instruction:

1. Click the Comparison icon.

2. Click on Column 2 of Rung 2.

The Comparison dialog box displays. The completed dialog box should look like this.

3. Click CMP in the Type: field.

4. Select Word in the Data Type: field.

5. In the S1 column, enter D10, D10, and D10 value = 0-249.

6. In the S2 column, enter 100, 0100, and Set value of 100.

7. In the D1 column, enter Q2, Q2, and D10 = 100 Q2 Comes On.

8. Click OK.
Step 4: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the menu commands toolbar.

The Save As dialog box displays.

2. Enter POT in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 5: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
**Analog Potentiometer Setting for MicroSmart**

The analog potentiometer positions and set values are shown below:

**All-In-One Type CPU Module**

- Analog Potentiometer 1
- Analog Potentiometer 2
  (24V I/O Type Only)

**Slim Type CPU Module**

- Analog Potentiometer 1
• The 10, 16, 20 and 40-I/O type CPU modules have one analog potentiometer.
• The 24-I/O type CPU module has two analog potentiometers.
• The values (0 through 255) set with analog potentiometers 1 and 2 are stored to data registers D8057 and D8058, respectively, and updated in every scan.
• The analog potentiometer can be used to change the preset value for a timer or counter.

<table>
<thead>
<tr>
<th>Special Data Registers for Analog Potentiometers</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Module</td>
</tr>
<tr>
<td>10- and 16-I/O Module</td>
</tr>
<tr>
<td>24-I/O Type</td>
</tr>
<tr>
<td>20 and 40-I/O Type</td>
</tr>
</tbody>
</table>

**Example: Changing Counter Present Value Using Analog Potentiometer**

This example demonstrates a program to change the counter preset value using analog potentiometer 1.

Analog potentiometer 1 value is stored to data register D8057, which is used as a present value for counter C0.

The preset value is changed between 0 and 255 using the potentiometer.
12. A/D Conversion Instructions

For Micro³ and Micro³C Only

Introduction to Analog/Digital Instructions

The analog to digital (A/D) instruction is used in conjunction with an A/D converter. The A/D converter can be used with the Micro³ and Micro³C to perform an 12-bit A/D conversion.

The A/D converter unit reads analog input signals from an analog output device such as an analog distance sensor or temperature sensor.

**Analog/Digital (A/D) Conversion**

![A/D Conversion Diagram]

When an input is on, the analog data from the A/D converter unit (FC2A-AD1, -AD2, -AD3, or -AD4) connected to input I0 is converted to a digital value and is set to a data register designated by destination operand D1.

The resolution of A/D conversions can be either 8 bits or 12 bits.

**NOTE:** The ONC and FA series PLCs have multiple analog inputs on one card. These analog input points are mapped directly to the input registers (FA series) and to the link registers (ONC). For details, refer to the user manual of your specific PLC.
**Valid Operands (Standard Processing)**

When an input is on, the analog data from the A/D converter unit connected to the input I0 is converted to a digital value and set to a data register designated by destination operand D1.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Destination to store data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–99/498</td>
<td></td>
</tr>
</tbody>
</table>

* Micro$^3 = 0–99$. Micro$^3C = 0–498.
Exercise 14: Creating an A/D Conversion Program

In this exercise, you will learn how to use the A/D instruction to convert the analog output of a pressure sensor to digital format data by using an A/D input card. This data can then be manipulated in the PLC.

Follow the steps below to create an A/D conversion program. When completed, your program screen should look like this:

Step 1: Create the Basic Program Logic: Open Contacts, Tag Names, and Comments as Illustrated in Exercises 1–8.

1. Click the New icon to open a new file.
Step 2: Enter an Analog to Digital Conversion on Rung 1

To enter an analog to digital conversion:

1. Click the Analog to Digital icon.
2. Click on Column 2 of Rung 1.

The Analog to Digital Conversion dialog box displays. The completed dialog box should look like this.

3. Click 12 bits for the Digital Data Range option.
4. Enter D30 in both the Tag Name: and Allocation Number: fields.
5. Enter pressure signal in the Tag Name Comments: field.
Step 3: Enter a Comparison Instruction on Rung 2

To enter a comparison instruction:

1. Click the Comparison icon.
2. Click on Column 2 of Rung 2.

The Comparison dialog box displays. The completed dialog box should look like this.

3. Click CMP>= in the Type field.
4. Select Word from the Data Type option.
5. Enter D30, D30, and pressure signal in the S1 column.
7. Enter Q2, Q2, and Valve Opens wired to Q2 in the D1 column.
Step 4: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.

The Save As dialog box displays.

2. Enter ANALOG in File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 5: Print, Transfer, and Monitor the Program

To print this program, refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
13. Calendar/Clock Instructions

Calendar and Clock
IDEC Micro³, Micro³C, ONC series, the FA3S-CP11T, and the MicroSmart are the only PLCs with a real-time clock and calendar. These PLCs can be programmed with date and time functions in applications in which any of these PLCs are used.

Calendar/Clock Instructions: (Micro³ and Micro³C)
The 16- and 24-I/O type Micro³ and Micro³C base units feature five real-time calendar and clock instructions used for programming the calendar and clock: calendar read (CALR), calendar write (CALW), clock read (CLKR), clock write (CLKW), and adjust (ADJ). These instructions cannot be used on the 10 I/O type Micro³ base units.

After initially setting the calendar and clock using FUN (function) 28, the date and time are maintained. If control data registers D95 – D98 are enabled using FUN10, the day of the week, hour, minute, and second data can be read out to these data registers when the Micro³ or Micro³C PLC is running or stopped. (For a detailed description of FUN28 and FUN10, please refer to the user’s manual included with your PLC.)
Calendar Read (CALR)

When an input is on, the calendar data (year, month, day, and day of week) is read to four data registers starting with the designated operand.

D = Year (0 to 99)
D+1 = Month (1 to 12)
D+2 = Day (1 to 31)
D+3 = Day of week (0 to 6) assigned as follows:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>Monday</td>
<td>Tuesday</td>
<td>Wednesday</td>
<td>Thursday</td>
<td>Friday</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

Example: CALR

When input I0 is on, calendar data is read to data registers D30 – D33.

D30 = Year
D31 = Month
D32 = Day
D33 = Day of the week
Clock Read (CLKR)

When an input is on, the clock data (hour, minute, and second) is read to three data registers starting with the designated operand.

D = Hour (0 to 23)  
D+1 = Minute (0 to 59)  
D+2 = Second (0 to 59)

Example: CLKR

When input I2 is on, clock data is read to data registers D50–D52.

D50 = Hour (0 to 23)  
D51 = Minute (0 to 59)  
D52 = Second (0 to 59)
**Valid Operands (Standard Processing)**

In the high-speed processing mode, data registers for this instruction are limited to D0 – D31. Since the CALR and CLKR instructions are executed in each scan while the input is on, a pulse input from an SOTU or SOTD instruction should be used as required.

<table>
<thead>
<tr>
<th>Operand Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination to read the calendar/clock data</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0 – 99 (Micro³)</td>
<td>—</td>
</tr>
</tbody>
</table>

| | | | | | | | | 0 – 498 (Micro³C) | |

**Repeat Operation in the Move Instruction**

When the source (S1) is designated with a repeat, as many operands as repeat cycles, starting with the operand designated by S1, are moved to the destination. As a result, only the last of the source operands is moved to the destination.
When the destination (D1) is designated with a repeat, the source operand designated with S1 is moved to as many destination operands as repeat cycles, starting with the destination designated by D1.

```
13 MOV S1 D10 D20 REP 3
```

Source (Repeat = 0)  Destination (Repeat = 3)

<table>
<thead>
<tr>
<th>D10</th>
<th>D20</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>111</td>
<td>110</td>
</tr>
<tr>
<td>112</td>
<td>110</td>
</tr>
</tbody>
</table>

When both S1 and D1 are designated with a repeat, as many operands as repeat cycles starting, with the operand designated by S1, are moved to the same quantity of operands, starting with the operand designated by D1.

```
14 MOV S1 R D10 D20 REP 3
```

Source (Repeat = 3)  Destination (Repeat = 3)

<table>
<thead>
<tr>
<th>D10</th>
<th>D20</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>112</td>
<td>112</td>
</tr>
</tbody>
</table>
Exercise 15: Creating a Calendar/Clock Program

In this exercise, you will learn how to read the clock and calendar values of a PLC and use them to control certain events at a specific time.

The following example explains how you can program the Micro³ PLC to turn a factory freezer off at 7A.M. Sundays and back on at 7A.M. Mondays of every week of the year.
Follow the steps below to create a clock/calendar program. When completed, your program screen should look like this:

**Step 1: Open a new program.**

1. Click the New icon 🎨 to open a new file.
Step 2: Enter an NO Contact on Rung 1

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 1 of Rung 1.

The Normally Open (LOD) dialog box displays.

3. Enter M0317 in the Tag Name: and Allocation Number: fields.

4. Enter Always On Contact in the Name Comment: field.

5. Click OK.
Step 3: Enter a Calendar Read Instruction on Rung 1

To enter a calendar read instruction:

1. Click the Clock/Calendar icon.
2. Click on Column 2 of Rung 1.

The Clock Read (CLKR) dialog box displays.

3. Click Calendar Read in the Type: field of the dialog box.
4. Enter D90 in the Tag Name: and Allocation Number: fields.
5. In the Tag Name Comment: field, enter:
   
   D90 = Year  
   D91 = Month  
   D92 = Day

6. Click OK.

The calendar read instruction displays at the end of Rung 1.
Step 4: Create a Second Line for Rung 1

To create a second line for Rung 1:

1. Click the Draw Line icon.

2. Click the crossbar to the right of the AlwaysOn contact on Rung 1, and drag the pointer down to the corresponding crossbar on Rung 2.

A line connecting Line 1 with Line 2 displays.
Step 5: Enter a Clock Read Instruction on Line 2

To enter a clock read instruction:

1. Click the Clock/Calendar icon.

2. Click on the last column of Line 2.

   The Clock Read (CLKR) dialog box displays.

3. Enter D96 in the Tag Name: and Allocation Number: fields.

4. In the Tag Name Comment: field, enter:
   
   D96 = Hour
   D97 = Minute
   D98 = Second

5. Click OK.

Notice that Rung 1 now has two lines. The clock read instruction displays at the end of Rung 1, Line 2.
Step 6: Enter an Comparison instruction on Rung 2

To enter an comparison instruction:

1. Click the Compare icon.

2. Click on Column 2 of Rung 2.

The Compare Equal To dialog box displays.

3. In the S1 column, enter D93, D93. The Data Register = Day of Week.

4. In the S2 column, enter 0, 0, 0 is Sunday = the first Day of Week.

5. In the D1 column, enter M10, M10, M10 is On if D93 = 0 “Sundays”.

6. Click OK.

The CMP≥ icon displays on Rung 2.
Step 7: Enter an Comparison instruction on Rung 3

To enter an comparison instruction:

1. Click the Compare icon.

2. Click on Column 2 of Rung 3.

The Compare Equal To dialog box displays.

3. In the S1 column, enter \textbf{D93}, \textbf{D93}, \textbf{The Data Register = Day of Week}.

4. In the S2 column, enter \textbf{1}, \textbf{1}, \textbf{1 is Monday = the Second Day of Week}.

5. In the D1 column, enter \textbf{M20}, \textbf{M20}, \textbf{M20 is On if D93 = 1 “Mondays”}.

6. Click OK.

The \textbf{CMP\geq} icon displays on Rung 3.
Step 8: Enter an Comparison instruction on Rung 4

To enter an comparison instruction:

1. Click the Compare icon.

2. Click on Column 2 of Rung 4.

The Compare Equal To dialog box displays.

3. In the S1 column, enter D96 = Hour D97 = Minute D98 = Second.

4. In the S2 column, enter 7, 7 = 7 Am The Freezer Goes Off.

5. In the D1 column, enter M30 = 7 AM.

6. Click OK.

The CMP ≥ icon displays on Rung 4.
**Step 9: Enter an NO Contact on Rung 5**

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 1 of Rung 5.

The Normally Open (LOD) dialog box displays.

3. Enter M10 in the Tag Name: and Allocation Number: fields.

4. Enter M10 is On if D93 = 0 “Sundays” in the Tag Name Comment: field.

5. Click OK.

The M0010 contact displays at the beginning of Rung 5.
Step 10: Enter an NO Contact on Rung 5

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 2 of Rung 5.

The Normally Open (LOD) dialog box displays.

3. Enter M30 in the Tag Name: and Allocation Number: fields.

4. Enter M30 is On if D96 = 7 AM in the Tag Name Comment: field.

5. Click OK.

The M0030 contact displays on Rung 5.
Step 11: Enter a Reset Instruction on Rung 5

To enter a Reset instruction:

1. Click the Reset icon.

2. Click on Column 3 of Rung 5.

The Reset dialog box displays.

3. Click the Reset option in the Type: field.

3. Enter Q0 in the Tag Name: and Allocation Number: fields.

4. Enter Freeze Off Sunday 7 AM and On Monday 7 AM in the Tag Name Comment: field.

5. Click OK.

The Reset contact displays at the end of Rung 5.
Step 12: Enter an NO Contact on Rung 6

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 1 of Rung 6.

The Normally Open dialog box displays.

3. Click the Normally Open option in the Type: field.

4. Enter M20 in the Tag Name: and Allocation Number: fields.

5. Enter M20 is On if D93 = 1 “Mondays” in the Tag Name Comment: field.

5. Click OK.

The M0020 contact displays at the beginning of Rung 6.
Step 13: Enter a Set instruction on Rung 6

To enter a Set instruction:

1. Click the Set icon.

2. Click on Column 3 of Rung 6.

The Set dialog box displays.

3. Click the Set option in the Type field.

4. Enter Q0 in the Tag Name and Allocation Number fields.

5. Enter Freezer Off Sunday 7 AM and On Monday 7 AM in the Tag Name Comment field.

6. Click OK.

The Q0 “Set” contact displays at the end of Rung 6.
Step 14: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.
   
   The Save As dialog box displays.

2. Enter CLOCKCAL in File Name field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 15: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.

Real-Time Clock and Calendar (ONC)

The real-time clock and calendar of the ONC features functionality and instructions used for programming the dates and times of events.

The Week Comparison Instruction

This set of instructions is the week compare on/off (WKCMP ON/OFF), week table (WKTBL), and calendar and clock word set (NSET).
As many WK cmp instructions as needed can be used to turn on and off designated output and internal relays at predetermined times and days of the week.

Once the internal calendar/clock is set, the WK cmp ON and OFF instructions compare the predetermined time with the internal clock. When the preset time is reached, the internal relay or output designated as the destination operand is turned on or off as scheduled.

**Week Compare ON (WK cmp ON)**

<table>
<thead>
<tr>
<th>WK cmp</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>D1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the input is on, the WK cmp ON instruction compares the S1 and S2 preset data with the current day and time.

When the current day and time reaches the resets, an output or internal relay designated by operand D1 is turned on, depending on the week table output control designated by S3.

**Week Compare OFF (WK cmp OFF)**

<table>
<thead>
<tr>
<th>WK cmp</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>D1</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the input is on, the WK cmp OFF instruction compares the S1 and S2 preset data with the current day and time.

When the current day and time reach the resets, an output or internal relay designated by operand D1 is turned off, depending on the week table output control designated by S3.
13-22 Advanced Instructions

WindLDR™ Tutorial

**Week Table (WKTBL)**

S1, S2, S3, …, SN → Week Table

When the input is on, N blocks of special month/day data in the operands designated by S1, S2, S3, …, SN are set to the week table.

The quantity of special days can be up to 50. The special days stored in the week table are used to add or skip days to turn on or off the comparison outputs programmed in subsequent WKCMP ON or WKCMP OFF instructions.

---

**Operand Function**

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>R</th>
<th>T</th>
<th>C</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 1 (S1)</td>
<td>Day of week comparison data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0–127</td>
<td>—</td>
</tr>
<tr>
<td>Source 2 (S2)</td>
<td>Hour/Minute comparison data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0–2359</td>
<td>—</td>
</tr>
<tr>
<td>Source 3 (S3)</td>
<td>Week table output control</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0–2</td>
<td>—</td>
</tr>
<tr>
<td>Destination 1 (D1)</td>
<td>Comparison ON output (WKCMP ON)</td>
<td>—</td>
<td>X</td>
<td>▲</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comparison OFF output (WKCMP OFF)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>

---

**Operands**

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>R</th>
<th>T</th>
<th>C</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 1 (S1)</td>
<td>Special month/day data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>101–1231</td>
<td>—</td>
</tr>
</tbody>
</table>

---

© 2001 IDEC Corporation
Examples: WKCMP ON/OFF

Without Special Days (S3 = 0)

This example is the basic program for week programmer application without using the WKTBL (week table) instruction. While the CPU is running, the WKCMP ON and WKCMP OFF compare the S1 and S2 preset data with the current day and time.

When the current day and time reach the presets, an output designated by operand D1 is turned on and off.

<table>
<thead>
<tr>
<th>M8125</th>
<th>WKCMP</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>D1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ON</td>
<td>62</td>
<td>815</td>
<td>0</td>
<td>Q0</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>62</td>
<td>1715</td>
<td>0</td>
<td>Q0</td>
</tr>
</tbody>
</table>

M8125 is the in-operation output special internal relay. S1 (62) specifies Monday through Friday. The WKCMP ON turns on output Q0 at 8:15 on Monday through Friday. The WKCMP OFF turns off output Q0 at 17:15 on Monday through Friday.

With Additional Days in the Week Table (S3 = 1)

When the current time reaches the hour/minute preset time on the special days programmed in the WKTBL, the designated output is turned on (WKCMP ON) or turned off (WKCMP OFF). In addition, the designated output is turned on and off every week as designated by operand S1 of WKCMP.

In normal execution, when the current day and time coincide with the preset day (S1) and time (S2), the designated output is turned on or off. Execution on the special days has precedence over execution on normal days.
This example demonstrates operation on special days in addition to regular weekends. The output is turned on from 10:18 a.m. to 11:03 p.m. on every Saturday and Sunday. Without regard to the day of week, the output is also turned on December 31 through January 3, and May 3 through May 5.

WKTBL designates Dec. 31 to Jan. 3 and May 3 to May 5 as special days. S1 (65) specifies Saturday and Sunday. S3 (1) adds special days. WKCMP ON turns on output Q0 at 10:18 on every Saturday, Sunday, and special days. The WKCMP OFF turns off output Q0 at 23:03 on the same days.

With Skip Days in the Week Table (S3 = 2)

On the special days programmed in the WKTBL, the designated output is not turned on or off, while the designated output is turned on and off every week as designated by operand S1 of WKCMP.

In normal execution, when the current day and time coincide with the preset day (S1) and time (S2), the designated output is turned on or off. Execution on the special days has precedence over execution on normal days.
This example demonstrates operation aborted on special days. The output is turned on from 8:45 a.m. to 10:32 p.m. on every Monday through Friday, but is not turned on December 31 through January 3, and May 3 through May 5.

<table>
<thead>
<tr>
<th>WKTBL</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1231</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>503</td>
<td>504</td>
<td>505</td>
</tr>
</tbody>
</table>

WKCM ON S1 (62) specifies Monday to Friday. S3 (2) skips special days. WKCM ON turns on output Q0 at 8:45 on every Monday through Friday except on special days. The WKCM OFF turns off output Q0 at 22:32 on the same days.

**Setting Calendar/Clock Using WindLDR**

Before using the week programmer instructions for the first time, the internal calendar clock must be set using WindLDR or executing a user program to transfer correct calendar/clock data to special data registers allocated to the calendar/clock. Once the calendar/clock data is stored, the data is held by the backup battery while the CPU power is turned off.
To Set the Calendar/Clock:

1. Select Online from the WindLDR menu bar, then select Monitor. The screen display changes to the monitor window.

2. From the Online menu, select PLC Status. The OpenNet PLC Status dialog is displayed. The current calendar/clock data is read out from the OpenNet Controller CPU and displayed in the Calendar box.

3. Click the Change button in the Calendar box. The Set Calendar and Time dialog comes up with the date and time values read from the computer internal clock.
Another way of setting the calendar/clock data is to move the values to special data registers dedicated to the calendar and clock and to turn on special internal relay M8020 by executing a user program. Data registers D8015 through D8021 do not hold the current values of the calendar/clock data but hold unknown values before executing a user program.

### Setting Calendar/Clock Using a User Program

Another way of setting the calendar/clock data is to move the values to special data registers dedicated to the calendar and clock and to turn on special internal relay M8020 by executing a user program. Data registers D8015 through D8021 do not hold the current values of the calendar/clock data but hold unknown values before executing a user program.

### Calendar/Clock Special Data Registers

<table>
<thead>
<tr>
<th>Data Register No.</th>
<th>Data</th>
<th>Value</th>
<th>Read/Write</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>D8008</td>
<td>Year (current data)</td>
<td>0 to 99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8009</td>
<td>Month (current data)</td>
<td>1 to 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8010</td>
<td>Day (current data)</td>
<td>1 to 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8011</td>
<td>Day of week (current data)</td>
<td>0 to 6 (Note)</td>
<td>Read only</td>
<td>100 msec or one scan time whichever is larger</td>
</tr>
<tr>
<td>D8012</td>
<td>Hour (current data)</td>
<td>0 to 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8013</td>
<td>Minute (current data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8014</td>
<td>Second (current data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Calendar/Clock Instructions

**Note:** The day of week value is assigned for both current and new data as follows:

<table>
<thead>
<tr>
<th>Day of Week</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>0</td>
</tr>
<tr>
<td>Monday</td>
<td>1</td>
</tr>
<tr>
<td>Tuesday</td>
<td>2</td>
</tr>
<tr>
<td>Wednesday</td>
<td>3</td>
</tr>
<tr>
<td>Thursday</td>
<td>4</td>
</tr>
<tr>
<td>Friday</td>
<td>5</td>
</tr>
<tr>
<td>Saturday</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Register No.</th>
<th>Data</th>
<th>Value</th>
<th>Read/Write</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>D8015</td>
<td>Year (new data)</td>
<td>0 to 99</td>
<td>Write only</td>
<td>Not updated</td>
</tr>
<tr>
<td>D8016</td>
<td>Month (new data)</td>
<td>1 to 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8017</td>
<td>Day (new data)</td>
<td>1 to 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8018</td>
<td>Day of week (new data)</td>
<td>0 to 6 (Note)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8019</td>
<td>Hour (new data)</td>
<td>0 to 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8020</td>
<td>Minute (new data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8021</td>
<td>Second (new data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Setting Calendar/Clock Data

This example demonstrates how to set calendar/clock data using a ladder program. After storing new calendar/clock data into data registers D8015 through D8021, special internal relay M8020 (calendar/clock data write flag) must be turned on to set the new calendar/clock data to the CPU.

M8120 is the initialize pulse special internal relay.

M8120

SOTU

MOV(W) S1 R D0 D1 R D8015 REP

10

M8020

SOTU

MOV(W) S1 R D4 D1 R D8019 REP

11

M1

MOV(W) S1 R D8008 D1 R D10 REP

M8125

M0

M1

© 2001 IDEC CORPORATION

13. Calendar/Clock Instructions
When the CPU starts, the NSET moves calendar/clock data to data registers D0 through D6.

When input I0 turns on, new calendar data (year, month, day, and day of week) are moved to data registers D8015 through D8018, and internal relay M0 is turned on for 1 scan time.

When input I1 turns on, new clock data (hour, minute, and second) are moved to data registers D8019 through D8021, and internal relay M1 is turned on for 1 scan time.

When either M0 or M1 is turned on, calendar/clock data write flag special internal relay M8020 is turned on to set the new calendar/clock data to the CPU.

M8125 is the in-operation output special internal relay.

While the CPU is running, the MOV(W) moves current calendar/clock data to data registers D10 through D16.

**Real-Time Clock and Calendar (FA3S-CP11T)**

The FA3S series CPU with real-time clock and calendar, PF3S-CP11T, features five instructions used for programming the clock and calendar: clock set (CLKST), clock read (CLKRD), clock comparison format (CLKCF), clock comparison execute (CLKCP), and clock adjust (CLKA). After initially setting the clock, the date and time are maintained and can be read out when the CPU is running or stopped.
Clock Set (CLKST)

\[ \text{CLKST} \quad \text{drS} \]

S + 1, S + 2, S + 3, … \rightarrow \text{clock, according to assignment code S}

When the input is on, the CPU clock is set using the data stored in a block of data registers starting with \( \text{drS} - 1 \), according to the data format designated by the assignment code stored in \( \text{drS} \). The assignment code may be a zero, one, or two. A zero specifies that the date and time are both assigned using values preset to the seven data registers following \( \text{drS} \). A one designates that time is assigned using the data preset to the three data registers following \( \text{drS} \). A two specifies that the date is assigned using values preset to the four data registers following \( \text{drS} \).

The CLKST instruction is executed with each scan when the input is on. Internal relay M630 turns on in case of a clock function interruption.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>R</th>
<th>T</th>
<th>C</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>drS</td>
<td>First source data register</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Assignment Codes

```
<table>
<thead>
<tr>
<th>I2</th>
<th>SOT</th>
<th>CLKST</th>
<th>drS</th>
<th>D10</th>
<th>D10</th>
</tr>
</thead>
</table>

D10  0   drS    Assignment Code = 0
D10  99  drS + 1 Year (0 – 99)
D10  02  drS + 2 Month (1 – 12)
D10  27  drS + 3 Day (1 – 31)
D11  04  drS + 4 Day of Week (0 – 6)
D12  15  drS + 5 Hour (0 – 23)
D11  33  drS + 6 Minute (0 – 59)
D12  112 drS + 7 Second (0 – 59)
```

Days of Week

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>1</td>
<td>Monday</td>
<td>2</td>
<td>Tuesday</td>
<td>3</td>
<td>Wednesday</td>
</tr>
</tbody>
</table>
```

Using the programming shown below, the CPU clock is set according to the assignment code stored in data register D10. Each of the three possible assignment codes is demonstrated with date and time values stored in the appropriate quantity of data registers.

The SOT instruction is used to execute the CLKST instruction only when input I2 turns on.

**NOTE:** Additional programming is required to store the desired values in the appropriate data registers before executing the CLKST instruction.
If a zero is stored in the data register designated by drS, then the next seven data registers must be preset with the appropriate values to assign the date and time. The first block of data registers shown on the left is used to set Saturday, February 27, 1999, 3:33:45.

**NOTE:** If the date or time is not within the specified range, then this invalid data will result in a numeric value end and M306 will turn on.

**Clock Read (CLKRD)**

```
CLKRD drD
```

Clock ➔ D + 1, D + 2, D + 3, ..., according to read-out code D

When the input is on, the current clock data is read out from the CPU to a block of data registers beginning with drD + 1, according to the data format designated by the read-out code stored in drD. The read-out code may be a zero, one, or two. A zero specified that the data and time are both read out to 7 data registers following drD. A one designates that the time is read out to 3 data registers following drD. A two specifies that the date is read out to 4 data registers after drD.

The CLKRD instruction is executed with each scan when the input is on, so use a level or pulse input signal as necessary. M630 turns on to signal that the clock function has been interrupted.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Input</th>
<th>Output</th>
<th>Internal Relay</th>
<th>Shift Register</th>
<th>Data Register</th>
<th>Timer/Counter</th>
<th>10msec Timer</th>
<th>Constant</th>
<th>External Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>drD</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
**Clock Adjust (CLKAJ)**

Adjust CPU clock: seconds → zero

When the input is on, the CPU clock is adjusted with respect to seconds. If seconds are between 0 and 29 for the current time, the adjustment for seconds will be set to zero and the minutes will remain the same. If the seconds are between 30 and 59 for the current time, then the adjustment for seconds will be set to zero and minutes are incremented by one. The CLKAJ instruction is useful for precise timing which starts at zero seconds. The CLKAJ instruction is executed with each scan when the input is on, so use a level or pulse input as necessary.
Comparison table \( \rightarrow T \); N data sets \( \rightarrow T + 1 \); Comparison pattern P \( \rightarrow T + 2 \);
format for clock comparison

When the input is on, a comparison table is created beginning with the data register designated by drT. A block of data registers beyond drT (depending on cdrN and cdrP) is required for storing format information. The CLKCF instruction is executed with each scan when the input is on, so use a level or pulse input as necessary.

### Clock Function (MicroSmart)

With the optional clock cartridge (FC4A-PT1) installed on any type of MicroSmart CPU modules, the MicroSmart can be used for time-scheduled control.
Setting Calendar/Clock

The calendar/clock data in the clock cartridge must be set to transfer the correct data from special data registers allocated to the calendar/clock. Once the calendar/clock data is stored, the data is held by the backup battery in the clock cartridge.

To Set Calendar/Clock

1. Select Online from the menu bar, select Monitor. The screen display changes to the monitor window.

2. From the Online menu, select PLC Status. The MicroSmart PLC Status dialog box is displayed. The current calendar/clock data is read out from the clock cartridge and displayed in the Calendar box.

3. Click the Change button in the Calendar box. The Set Calendar and Time dialog box comes up with the date and time values read from the computer internal clock.

4. Click the Down Arrow button on the right of the Calendar, then the calendar is displayed where you can change the year, month, and date. Enter or select new values.

5. To change hours and minutes, click in the Time box, and type a new value or use the up/down keys. When new values are entered, click the OK button to transfer the new values to the clock cartridge.
Another way of setting the calendar/clock data is to store the values in special data registers dedicated to the calendar and clock and to turn on special internal relay M8016, M8017, or M8020. Data registers D8015 through D8021 do not hold the current values of the calendar/clock data but hold unknown values executing a user program.

<table>
<thead>
<tr>
<th>Data Register No.</th>
<th>Data</th>
<th>Value</th>
<th>Read/Write</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>D8008</td>
<td>Year (current data)</td>
<td>0 to 99</td>
<td>Read only</td>
<td>500 msec or one scan time whichever is larger</td>
</tr>
<tr>
<td>D8009</td>
<td>Month (current data)</td>
<td>1 to 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8010</td>
<td>Day (current data)</td>
<td>1 to 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8011</td>
<td>Day of week (current data)</td>
<td>0 to 6 (Note)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8012</td>
<td>Hour (current data)</td>
<td>0 to 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8013</td>
<td>Minute (current data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8014</td>
<td>Second (current data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8015</td>
<td>Year (new data)</td>
<td>0 to 99</td>
<td>Write only</td>
<td>Not updated</td>
</tr>
<tr>
<td>D8016</td>
<td>Month (new data)</td>
<td>1 to 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8017</td>
<td>Day (new data)</td>
<td>1 to 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8018</td>
<td>Day of week (new data)</td>
<td>0 to 6 (Note)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8019</td>
<td>Hour (new data)</td>
<td>0 to 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8020</td>
<td>Minute (new data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8021</td>
<td>Second (new data)</td>
<td>0 to 59</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The day of week value is assigned for both current and new data as follows:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sunday</td>
<td>Monday</td>
<td>Tuesday</td>
<td>Wednesday</td>
<td>Thursday</td>
<td>Friday</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

© 2001 IDEC CORPORATION
Example: Setting Calendar/Clock Data

This example demonstrates how to set calendar/clock data using a ladder program. After storing new calendar/clock data into data registers D8015 through D8021, special internal relay M8020 (calendar/clock data write flag) must be turned on to set the new calendar/clock data to the clock cartridge.

M8120 is the initialize pulse special internal relay.

When the CPU starts, seven MOV(W) instructions store calendar/clock data to data registers D0 through D6.

When input I0 is turned on, new calendar data (year, month, day, and day of week) are moved to data registers D8015 through D8018, and internal relay M0 is turned on for 1 scan time.

When input I1 is turned on, new clock data (hour, minute, and second) are moved to data registers D8019 through D8021, and internal relay M1 is turned on for 1 scan time.
When either M0 or M1 is turned on, calendar/clock data write flag special internal relay M8020 is turned on to set the new calendar/clock data to the clock cartridge.

M8125 is the in-operation output special internal relay.

While the CPU is running, the MOV(W) moves current calendar/clock data to data registers D10 through D16.

**Adjusting Clock Using a User Program**

Special internal relay M8021 (clock data adjust flag) is provided for adjusting the clock data. When M8021 is turned on, the clock is adjusted with respect to seconds. If seconds are between 0 and 29 for current time, adjustment for seconds will be set to 0 and minutes remain the same. If seconds are between 30 and 59 for current time, adjustment for seconds will be set to 0 and minutes are incremented one. M8021 is useful for precise timing which starts at zero seconds.

**Example: Adjusting Calendar/Clock Data to 0 Seconds**

When input I2 is turned on, clock data adjust flag special internal relay M8021 is turned on and the clock is adjusted with respect to seconds.
14. Transmit/Receive Instructions

For Micro³C, ONC, and MicroSmart Only

Transmit and Receive

The ONC has two RS232 loader ports, the Micro³C has one RS232C loader port, and the MicroSmart 10-I/O type CPU module has one RS232 loader port and the 16-, 20-, 24 and 40 I/O and type CPU modules have one RS232 loader port (with Port 2 installing an optional RS232C communication adapter FC4A-PC1) that allow these controllers to communicate (that is, transmit and receive data) with any RS232C intelligent device or remote terminal. The instructions for transmitting and receiving data are as follows:

**Transmit (TXD)**

\[\text{TXD} \quad \text{S1} \quad \text{D1} \quad \text{D2}\]

When an input is on, the data designated by S1 is converted into a special format and transmitted through the loader port to a remote terminal with an RS232C port.
## Valid Operands (Standard Processing)

<table>
<thead>
<tr>
<th>PLC</th>
<th>Operand</th>
<th>Function</th>
<th>CPU Type</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro³C</td>
<td>S1</td>
<td>Source 1</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 498</td>
<td>00h – FFh</td>
</tr>
<tr>
<td>ONC</td>
<td>10-I/O</td>
<td>Transmit data</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 7999</td>
<td>00h – FFh</td>
</tr>
<tr>
<td>MicroSmart</td>
<td>16- or 24-I/O</td>
<td>—</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 1299</td>
<td>00h – FFh</td>
</tr>
<tr>
<td></td>
<td>20-, 40-I/O</td>
<td>—</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 1299</td>
<td>00h – FFh</td>
</tr>
<tr>
<td>Micro³C</td>
<td>D1</td>
<td>Destination 1</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 31</td>
<td>0 – 287</td>
</tr>
<tr>
<td>ONC</td>
<td>10-I/O</td>
<td>Transmit completion output</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 597</td>
<td>0 – 2557</td>
</tr>
<tr>
<td>MicroSmart</td>
<td>16-I/O</td>
<td>—</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 6</td>
<td>0 – 127</td>
</tr>
<tr>
<td></td>
<td>24-I/O</td>
<td>—</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 107</td>
<td>0 – 187</td>
</tr>
<tr>
<td></td>
<td>20-I/O</td>
<td>D20K3/D2053</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 187</td>
<td>0 – 1277</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D20RK1/D20RS1</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 307</td>
<td>0 – 307</td>
</tr>
<tr>
<td></td>
<td>40I/O</td>
<td>—</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 307</td>
<td></td>
</tr>
</tbody>
</table>

1. Transmit data
2. Transmit completion output

© 2001 IDEC Corporation
When an input is on, the data received through the loader port from a remote terminal with an RS232C port is converted and stored in data registers according to the receive format designated by S1.
### Valid Operands (Standard Processing)

<table>
<thead>
<tr>
<th>PLC</th>
<th>Operand</th>
<th>Function</th>
<th>CPU Type</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro3C</td>
<td></td>
<td>Transmit data</td>
<td>CPU Type</td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-498 FFh</td>
<td></td>
</tr>
<tr>
<td>ONC</td>
<td></td>
<td>S1 (Source 1)</td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-7999 FFh</td>
<td></td>
</tr>
<tr>
<td>MicroSmart</td>
<td></td>
<td>Transmit data</td>
<td>CPU Type</td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-399 FFh</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10-I/O</td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-1299 FFh</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D1 (Destination 1)</td>
<td>Transmit completion output</td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-7999 FFh</td>
<td></td>
</tr>
<tr>
<td>Micro3C</td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-287</td>
<td></td>
</tr>
<tr>
<td>ONC</td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-2557 FFh</td>
<td></td>
</tr>
<tr>
<td>MicroSmart</td>
<td></td>
<td>D1 (Destination 1)</td>
<td>CPU Type</td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-317 FFh</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10-I/O</td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-1277 FFh</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>16- or 24-I/O</td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-1277 FFh</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20-, 40-I/O</td>
<td></td>
<td>I</td>
<td>Q</td>
<td>M</td>
<td>T</td>
<td>C</td>
<td>R</td>
<td>D</td>
<td>0-1277 FFh</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The table includes valid operands for various PLC types with specific ranges for each operand type.
Since data communication requires more data registers, the ONC has 8000 data registers (D0 – D7999), the Micro³C has 500 data registers (D0–D499), and the MicroSmart 10-I/O type has 400 data registers (0–399) and the 16-,24-I/O types have 1300 data registers (0–1299). The ONC and Micro³C offer superior communication. They are capable of communicating with any RS232 intelligent device using either ASCII or binary code.

<table>
<thead>
<tr>
<th>PLC</th>
<th>Operand</th>
<th>Function</th>
<th>CPU Type</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro³C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–497</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONC</td>
<td>D2 (Desti-nation 2)</td>
<td>Transmit status</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–7999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MicroSmart</td>
<td>10-I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–399</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16- or 24-I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–1299</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20-, 40- I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0–1299</td>
<td>2000-7999</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 16: Creating a Transmit Program

In this exercise, you will learn how to use the Micro³C, ONC, or MicroSmart transmit instruction to automatically dial and send a message to a pager.

This is a flow control program which shows the Micro³C controlling a tank and reporting any “overflow” or “low level” to a maintenance engineer’s pager. If the pager is alphanumeric, then the PLC can transmit text messages as well as numeric ones.
Follow the steps below to create a transmit program. When completed, your program screen should look like this:
Step 1: Click the New Icon to Open a New File

A blank ladder logic program screen displays.

Step 2: Enter an NO Contact on Rung 1

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 1 of Rung 1.

The Normally Open (LOD) dialog box displays.

3. Enter M0317 in the Tag Name: and Allocation Number: fields.

4. Enter Always On in the Name Comment: field.

5. Click OK.

The Always On contact displays at the beginning of Rung 9.
Step 3: Enter an Analog to Digital Conversion Instruction

To enter an analog to digital conversion instruction:

1. Click the Analog to Digital icon.

2. Click on Column 2 of Rung 1.

The Analog to Digital Conversion dialog box displays.

3. Click 12 bits in the Digital Data Range field.

4. Enter D0 in the Tag Name: and Allocation Number: fields.

5. Enter Water Tank Level Sensor in the Name Comment: field.

6. Click OK.

The analog to digital conversion instruction displays on Rung 1.
Step 4: Enter a Rung Comment for Rung 1

To enter a comment for Rung 1:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter: This is a flow control program which shows the Micro3C controlling a tank and reporting any “overflow” or “low level” to maintenance engineer Pager. If the pager is alphanumeric, then the PLC can transmit text messages as well as numeric ones.

4. Click OK to save the comments.

The rung comment displays above Rung 1.
Step 5: Enter a Comparison Instruction on Rung 2.

To enter a comparison instruction:

1. Click the Comparison icon.

2. Click on Column 2 of Rung 2.

The Comparison dialog box displays.

3. Click CMP in the Type: field.

4. Enter D0, D0, and Water Tank Level Sensor in the S1 column.


4. Enter M0, M0, and I.R. Trigger TXD Instruction in the D1 column.

5. Click OK.
Step 6: Enter a Comparison Instruction on Rung 2.

To enter a comparison instruction:

1. Click the Comparison icon.

2. Click on below the previous comparison instruction on Column 2 of Rung 2.

The Comparison dialog box displays.

3. Click CMP<= in the Type: field.

4. Enter D0, D0, and Water Tank Level Sensor in the S1 column.

4. Enter 100, 100, and 100 = Level very low in the S2 column.

4. Enter M1, M1, and I.R.Trigger TXD Instruction in the D1 column.

5. Click OK.
Step 7: Enter a Single Output Up (SOTU) Instruction on Rung 2

To enter an SOTU instruction:

1. Click the SOTU icon.
2. Click on Column 2 of Rung 2.

The SOTU instruction displays next to the TankFull contact on Rung 2.

Step 8: Enter a Transmit Instruction on Rung 2

To enter a transmit instruction:

1. Click the Transmit/Receive icon.
2. Click on Column 3 of Rung 2.
The Transmit (TXD) dialog box displays.

3. Click the Insert button.

The Data Type Selection dialog box displays.

4. Click the OK key or press the [ENTER] key to select the Constant (Character) default in the Type field.

The Constant (Character) dialog box displays.

5. Enter ATDT #######,, in the Character field.

**NOTE:** “####” is the pager number. “,,,,” creates an eight second transmission delay.
6. Click the **OK** button or press the [ENTER] key to confirm.

7. Click the **Insert** button in the **Transmit (TXD)** dialog box again.

8. Click **Variable (Data Register)** in the **Type** field of the **Data Type Selection** dialog box.

9. Click the **OK** button or press the [ENTER] key to confirm.

The **Variable (Data Register)** dialog box displays.

10. Enter **D0, D0, Water Tank Level Sensor** in the **DR No.** column.

11. Leave the **BCD to ASCII** default in the **Conversion Type** field.

12. Enter **4** in the **Digits** field.

13. Leave the **1** default in the **REP** field.

14. Click the **OK** button or press the [ENTER] key to confirm.
15. Click the Insert button in the Transmit (TXD) dialog box again.

The Data Type Selection dialog box displays.

16. Click the OK key or press the [ENTER] key to select the Constant (Hexadecimal) default in the Type field.

The Constant (Hexadecimal) dialog box displays.

17. Enter 0d in the Hexadecimal field.

18. Click OK to confirm. Notice that you return to the Transmit (TXD) dialog box.

19. In the D1 column, enter M10, M10, and Emergency “Overflow.”

20. In the D2 column, enter D12, D12, and Transmission Status.
Step 9: Enter an NO Contact on Rung 4

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 1 of Rung 4.

The Normally Open (LOD) dialog box displays.

3. Enter M1 in the Tag Name field of the dialog box.

Notice that the properties for the M1 contact automatically display in the Allocation Number and Tag Name Comment fields.

4. Click the OK button or press the [ENTER] key to confirm.

The CmpltnD contact displays at the beginning of Rung 4.
Step 10: Enter a Single Output Up (SOTU) Instruction on Rung 4

To enter an SOTU instruction:

1. Click the SOTU icon.
2. Click on Column 2 of Rung 4.

The SOTU instruction displays next to the M1 contact on Rung 4.

Step 11: Enter a Transmit Instruction on Rung 4

To enter a transmit instruction:

1. Click the Transmit/Receive icon.
2. Click on Column 3 of Rung 3.

The Transmit (TXD) dialog box displays.

3. Click the Insert button in the dialog box.

The Data Type Selection dialog box displays.

4. Click the OK button or press the [ENTER] key to select the Constant (Character) default in the Type field.
The Constant (Character) dialog box displays.

5. Enter `ATDT1800262433211111 EMPTY#` in the Character field.

6. Click the OK button or press the [ENTER] key to confirm.

7. Click the Insert button again.

8. Click Constant (Hex decimal) in the Type field of the Data Type Selection dialog box.

9. Click the OK button or press the [ENTER] key to confirm.
The Constant (Hexadecimal) dialog box displays.

10. Enter **0D** in the **Hex decimal** field of the dialog box.

11. Click the **OK** button or press the **[ENTER]** key to confirm.

Notice that you return to the **Transmit (TXD)** dialog box.

12. Enter **M12** in the **D1 Tag Name:** field of this dialog box.

13. Enter **M12** in the **D1 Allocation Number:** field.

14. Enter **Emergency “low Level”** in the **D1 Tag Name Comment:** field.

15. Enter **D12** in the **D2 Tag Name:** field.

16. Enter **D12** in **D2 Allocation Number:** field.

17. Enter **Transmission Status D12** in the **D2 Tag Name Comment:** field.

18. Click the **OK** button or press the **[ENTER]** key to confirm.

The transmit instruction displays at the end of Rung 4.
Step 12: Save the Program

To name and save this program:

1. Click File ➔ Save as… from the Menu Commands toolbar.

   The Save As dialog box displays.

2. Enter DIALIT in the File Name: field of the dialog box.

3. Click the OK button or press the [ENTER] key to confirm.

Step 13: Print, Transfer, and Monitor the Program

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
Transmit and Receive Instructions for FA3S High Performance Series

Serial Transmitting (STXFT and STXEX)

The FA3S can transmit data to any device which is capable of receiving RS232 or RS422 serial communication. In addition, the FA3S can dial up a remote device equipped with a modem and transmit data. The instructions for transmitting are similar to those for receiving, including the STXFT instruction for formatting and the STXEX instruction for execution.

Serial TX Format (STXFT)

Designate execution conditions for user-defined communication, serial transmission.

- cPORT: Communication port
- STDR: Communication status
- oir: Completion output signal
When an input is on, a buffer in the system program is formatted to transmit communication specified by the STXEX instruction with the same value for “L” (0 to 9). Signal is received through cPORT and set to 0 for the program loader port.

**Communication Status (STDR)**

The operand designated by STDR is specified as the data register for the system to store the status of the current communication.

The data register, STDR, is intended to use primarily for monitoring to determine the current stage of communication or why communication is not successful.

If communication is proceeding successfully, the following value will be stored in STDR:

$$STDR = 10$$  Character array has been created; waiting for transmission

The result of the following status message is that the output or internal relay designated by OIR will be turned on to signal that communication is complete:

$$STDR = 20$$  Transmission complete; echo, if specified by controller D3032, is received

The following status message signals unsuccessful communication:

$$STDR = 256$$  Transmission buffer is full when another STXEX command is sent
Serial RX Format (STXFT) User-Defined Communication

<table>
<thead>
<tr>
<th>SRXFT</th>
<th>cPORT</th>
<th>STDR</th>
<th>oir</th>
<th>MAXSD</th>
<th>MINSD</th>
<th>ENDDL</th>
<th>dr D</th>
<th>DNLMT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Designate execution conditions for user-defined communication, serial receiving.

- cPORT: Communication port
- STDR: Communication status
- oir: Completion output signal
- MAXSD: Maximum start delimiter
- MINSD: Minimum start delimiter
- ENDDL: End delimiter
- dr D: First destination data register
- DNLMT: Maximum total bytes

The FA3S has a serial port to receive and transmit RS232 and RS422 communication, as well as communication with the program loader using FA instructions.

Communication is generally classified as standard or user-defined. Standard communication does not require programming to transmit and receive; only the appropriate CPU configurations must be set.

User-defined communication includes all other 1:1 and 1:N communication. User-defined 1:1 communication includes receiving and dialing/transmitting through a modem or two-way dialog with an operator interface/annunciator. It is also possible to accept alpha or numeric data from a bar code scanner/decoder and to send alpha and numeric characters to an LED display or as Centronics-compatible printer.
Serial RX Execute (STXEX) User-Defined Communication

Execute user-defined communication, serial receiving.

- \( \text{dr S} \): First source data register
- \( \text{oir} \): Comparison output signal
- \( c_{\text{NBLK}} \): Quantity of data blocks (1 to 255)
- \( K \): Code for comparison/conversion
- \( \text{dr S/D} \): Source data register for comparison or destination data register for conversion

When an input turns on, data is taken from the source data registers starting with \( \text{dr S} \) (which are the same as the destination data registers, \( \text{dr D} \), in the SRXFT instruction).

Incoming data is divided into “n” blocks, designated by \( c_{\text{NBLK}} \) (1 to 255). Each block of data is represented by a value for \( K \) and a corresponding operand, \( \text{dr S/D} \). Used with FUN147, \( K \) can be a constant value specified by the operand or supplied by an external setting.

When \( K \) equals zero, the block of data is not processed, no matter how \( \text{dr S/D} \) is specified. When \( K \) equals 100, a certain quantity of characters is not processed, as specified by \( \text{dr S/D} \).

When \( K \) equals 1 to 99 or 1000 to 1255, the block is a comparison. If \( K \) equals 1 to 99, the quantity of source data registers is specified, starting with \( \text{dr S/D} \), to be compared with incoming data. If \( K \) equals 1000 to 1255, the incoming data
is compared with values of existing records in the corresponding data file (zero to 255), starting with the record dr S/D. The quantity of data file records to be compared is specified by the next data register (dr S/D) + 1.

When K equals 201 to 204, 301 to 304, or 401 to 404, the block is a conversion. K specifies the kind of conversion, 20x, 30x, or 40x, with “x” as the quantity of digits or source data registers. Destination data register for the converted value is designated by dr S/D.

**NOTE:** With comparison or conversion, incoming data is taken sequentially from data registers starting with (dr S) + 1, corresponding to blocks K₁ to Kₙ. The first data register, dr S in the SRXEX instruction, stores the quantity of bytes for the incoming message.

The output or internal relay designated by oir turns on when the comparison is true or when no comparison block exists in the SRXEX instruction. When the comparison is false, oir turns off.

**Designating Data Comparison or Conversion**

In the SRXEX instruction, the operand designated by c NBLK designates the quantity of data blocks to be compared or converted (1 to 255). Each block of data is represented by a value for K and a corresponding operand, dr S/D. The value for K determines the operation and the quantity of data registers or data file records to be operated on, as:

\[
\begin{align*}
K = 0 & \quad \text{Block of data is not processed, no matter how dr S/d is specified} \\
K = 1 \text{ to } 99 & \quad \text{Data is compared to “K” data registers, starting with dr S/D} \\
K = 100 & \quad \text{Certain characters of data are not processed, as specified by dr S/D}
\end{align*}
\]
K = 201 to 204  Data is converted to BCD (1 to 4 digits), stored in dr S/D

K = 301 to 304  Data is converted to hexadecimal ASCII (1 to 4 digits), stored in dr S/D

K = 401 to 404  Data is converted to hexadecimal pseudo-ASCII (1 to 4 digits), stored in dr S/D

K=1000 to 1255  Data is compared to data file #(K – 1000), starting with record #(dr S/D); the quantity of data file records to be compared is specified by (dr S/D) + 1

**User-Defined Communication Mode (UDCMS)**

Designate whether or not user-defined communication is allowed.

The FA3S will operate in the standard or the user-defined communication mode. There are two additional switches which designate user-defined communication. The first is the value stored in the control register D3032. This 16-bit value has three significant bits:
When the command bit is set to one, user-defined communication is possible. However, user-defined communication is not enabled without using the input or internal relay designated by an UDCMS instruction.

When the input or internal relay designated by iir is on and when the command bit of control register D3032 is set to 1, user-defined communication is enabled. When the operand designated by iir is off, the standard communication mode is used.

**Example: Serial Receiving**

The first example for user-defined communication will demonstrate incoming data received in the format shown for dr S and processed to dr S/D.

**NOTE:** Data for comparison are given values and data for conversion are represented by shaded blocks.
Before comparison, desired values must be pre-loaded into data registers D15 through D18 as shown. After conversion, incoming data is stored in data registers D19 and D20 as shown.

\[
\text{dr S:} \quad \text{RD} \quad \text{#} \quad \text{C_R}
\]

\[
\text{dr S/D:} \\
\begin{array}{c|c}
D15 & 82 \quad R \\
D16 & 68 \quad D \\
D17 & 35 \quad # \\
D18 & 13 \quad C_R \\
D19 & \text{ASCII 2 Digits} \\
D20 & \text{ASCII 4 Digits}
\end{array}
\]

<table>
<thead>
<tr>
<th>Desired Character (Alpha or Numeric)</th>
<th>Hexadecimal Character Code</th>
<th>Decimal Character Code</th>
<th>Values Set to Data Registers in Advance</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>52</td>
<td>82</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>44</td>
<td>68</td>
<td>#</td>
</tr>
<tr>
<td>#</td>
<td>23</td>
<td>35</td>
<td>C_R</td>
</tr>
<tr>
<td>C_R</td>
<td>OD</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>
Example: Serial Receiving

In the ladder diagram above, five data blocks are specified in the SRXEX instruction. The incoming data is compared or converted according to the values assigned to K1 through K5.

Data Comparison

First Block  
K₁ = 2, S₁ = D₁₅

K₂ = 30₂, D₂ = D₁₉

K₃ = 1, S₃ = D₁₇

K₄ = 2₀₄, D₄ = D₂₀

K₅ = 1, S₅ = D₁₈

Data Conversion

D₁₅ = 8₂

D₁₆ = ₆₈

D₁₇ = ₃₅

D₁₈ = ₁₃

D₁₉ = D₁₉

D₂₀ = D₂₀

The data registers corresponding to dr S/D are shown. For comparison, dr S/D is shown as S for source. For conversion, dr S/D is shown as D for destination.

Example: Bar Code Reading with FIFO File

In this example, serial receiving is used to accept bar code data from a scanner/decoder. This data is then combined with first-in, first-out (FIFO) data files and data search instructions to devise a security system.

The bar code data might be used as an access code for using specialized machinery, editing confidential files, or for entering high-security areas. In the primary mode, the bar code data is compared to an existing FIFO data file. When there is
a match, access is granted. If there is no match, access is denied. In the secondary mode, a new set of approved codes may be entered into the FIFO data file by authorized personnel.

**FA3S PLC System**

The wiring for a bar code scanner/decoder is shown above.

The system stores the code to indicate the communication status in data register D11 (STDR). Start delimiters are 255 and 0, accepting any beginning byte. When CR is received, data is complete and internal relay M1 turns on to start conversion and comparison.
Incoming data for the ladder diagram below is initially stored in the data registers starting with D20 (dr D for the SRXFT instruction and dr S for the SRXEX instruction), as shown.

The first-in, first-out (FIFO) data file is created using data registers starting with D30. The first record, as specified by the first-in pointer, is stored in data register D32.

Programming can be used to initialize the FIFO data file. The constant value, zero, is set to D30 as the first-in pointer to store the first record to D32 (first-in pointer is not used). Eight records are cleared to zero.
XY Conversion

The ONC and the FA series CPUs have coordinate conversion instructions. These ONC instructions convert one data point of one plane to a corresponding point of another plane, using a linear relationship between values of X and Y. The ONC instructions are Convert X to Y (CVXTY), Convert Y to X (CVYTX), and Average (AVRG).

XY Format Set (XYFS)

When an input is on, the format for XY conversion is set. The number of XY coordinates, defining the linear relationship between X and Y, can be 2 to 32 points ($0 \leq n \leq 31$).

Valid Operands (Standard Processing)

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (Source 1)</td>
<td>Format number</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0 – 29</td>
<td>—</td>
</tr>
<tr>
<td>X0 – Xn</td>
<td>X value</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0 to 32767</td>
<td>—</td>
</tr>
<tr>
<td>Y0 – Yn</td>
<td>Transmit status</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-32768 to 32767</td>
<td>—</td>
</tr>
</tbody>
</table>

When the timer (T) or counter (C) is used as X0 through Xn or Y0 through Yn, the current value of the timer/counter is read out.
S1 = Format Number
Select a format number 0 through 29. A maximum of 30 formats for XY conversions can be set.

Xn = X Value
Enter a value for the X coordinate. The integer value can be 0 through +32767. If the X value becomes negative, a user program execution error will result, turning on special internal relay M8004 and the ERROR LED.

Yn = Y Value
Enter a value for the Y coordinate. The integer value can be −32768 through +32767.

Valid Data Types

<table>
<thead>
<tr>
<th>Word (W)</th>
<th>Integer (I)</th>
<th>Double Word (D)</th>
<th>Long (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>X</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

When a bit operand such as input (I), output (Q), internal relay (M), or shift register (R) is designated as Xn or Yn, 16 points (integer data type) are used.

When a word operand such as timer (T), counter (C), data register (D), or link register (L) is designated as Xn or Yn, 1 point (integer data type) is used.
**Convert X to Y (CVXTY)**

When an input is on, the X value designated by operand S2 is converted into corresponding Y value according to the linear relationship defined in the XYFS instruction. Operand S1 selects a format from a maximum of 30 XY conversion formats. The conversion result is set to the operand and designated by D1.

**Valid Operands (Standard Processing)**

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (Source 1)</td>
<td>Format number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 – 29</td>
<td></td>
</tr>
<tr>
<td>S2 (Source 2)</td>
<td>X value</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0 to 32767</td>
<td></td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>Destination to store results</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>▲</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

▲ Internal relays M0 through M2557 can be designated as D1. Special internal relays cannot be designated as D1. When the timer (T) or counter (C) is used as S2, the timer/counter current value is read out. When the timer (T) or counter (C) is used as D1, the data is written in as a preset value which can be 0 though 65535.

The data conversion error is ±0.5.

**NOTE:** The only type of valid data for an XY conversion is an integer.
**15-4 Advanced Instructions**

**WindLDR™ Tutorial**

---

**Convert Y to X (CVYTX)**

When an input is on, the Y value designated by operand S2 is converted into corresponding X value according to the linear relationship defined in the XYFS instruction. Operand S1 selects a format from a maximum of 30 XY conversion formats. The conversion result is set to the operand and designated by D1.

**Valid Operands (Standard Processing)**

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>I</th>
<th>Q</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (Source 1)</td>
<td>Format number</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0 – 29</td>
<td>—</td>
</tr>
<tr>
<td>S2 (Source 2)</td>
<td>Y value</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>–32768 to 32767</td>
<td>—</td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>Destination to store results</td>
<td>—</td>
<td>X</td>
<td>▲</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

▲ Internal relays M0 through M2557 can be designated as D1. Special internal relays cannot be designated as D1. When the timer (T) or counter (C) is used as S2, the timer/counter current value is read out. When the timer (T) or counter (C) is used as D1, the data is written in as a preset value which can be 0 though 65535.

The data conversion error is ±0.5.

**NOTE:** The only type of valid data for an XY conversion is an integer.
Average (AVRG)

When an input is on, the sampling data designated by operand S1 is processed according to the sampling conditions designated by operands S2 and S3.

When sampling is complete, the average, maximum, and minimum values are stored to the three consecutive operands starting with the operand designated by D1. Then sampling completion output designated by operand D2 is turned on.

This instruction is effective for the data processing of analog input values. A maximum of 10 AVRQ instructions can be programmed in a user program.

Valid Operands (Standard Processing)

<table>
<thead>
<tr>
<th>Operand</th>
<th>Function</th>
<th>F</th>
<th>O</th>
<th>M</th>
<th>T</th>
<th>C</th>
<th>R</th>
<th>D</th>
<th>L</th>
<th>Constant</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (Source 1)</td>
<td>Sampling data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>S2 (Source 2)</td>
<td>Sampling end input</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>S3 (Source 3)</td>
<td>Sampling cycles</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>0 to 65535</td>
<td>—</td>
</tr>
<tr>
<td>D1 (Destination 1)</td>
<td>First operand number to store results</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>D2 (Destination 2)</td>
<td>Sampling completion output</td>
<td>—</td>
<td>X</td>
<td>▲</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

NOTE: The only type of valid data for an XY conversion is an integer.
Exercise 17: Creating an X/Y Linear Relationship

This exercise demonstrates how to set up two coordinate points to define the linear relationship between X and Y. The two points are \((X_0, Y_0) = (0,0)\) and \((X_1, Y_1) = (4000, 2000)\). Once these are set, there is an X to Y conversion, as well as a Y to X conversion.

Follow the steps below to create a linear conversion relationship. When completed, your program screen should look like this:
Step 1: Click the New Icon to Open a New File

A blank ladder logic program screen displays.

Step 2: Enter an NO Contact on Rung 1

To enter an NO contact:

1. Click the Normally Open icon.
2. Click on Column 1 of Rung 1.

The Normally Open (LOD) dialog box displays.

3. Enter M8120 in the Tag Name: and Allocation Number: fields.
4. Enter Initialize Pulse in the Name Comment: field.
5. Click OK.

The Initialize Pulse contact displays at the beginning of Rung 1.
Step 3: Enter an XY Format Set Instruction on Rung 1

To enter an XY format set instruction:

1. Click the XY Format Set icon.

2. Click on Column 2 of Rung 1.

The XY Convert Format (XYFS) dialog box displays.

3. Enter 1, 1, First Point in the S1 field.

4. Enter 0, 0, x value initial Pt. in the X0 field.

5. Enter 0, 0, y value initial Pt. in the Y0 field.

6. Enter 4000, 4000, x value second Pt in the X1 field.

7. Enter 2000, 2000, y value second point in the Y1 field.

8. Click OK.

The XYFS instruction displays on Rung 1.
Step 4: Enter a Rung Comment for Rung 1

To enter a comment for Rung 1:

1. Click anywhere on Rung 1.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

   The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter: This is an X to Y conversion program. If any two points on a line are determined, this XY instructions determine all remaining points.

   This program allows the X values to be converted to Y values and Visa Versa. The points are (0,0) and (4000,2000).

4. Click OK to save the comments.

   The rung comment displays above Rung 1.
Step 5: Enter an NO Contact on Rung 2

To enter an NO contact:

1. Click the Normally Open icon.

2. Click on Column 1 of Rung 2.

The Normally Open (LOD) dialog box displays.

3. Enter 10 in the Tag Name: and Allocation Number: fields.

4. Enter Convert X to Y in the Name Comment: field.

5. Click OK.

The Convert X to Y contact displays at the beginning of Rung 2.
Step 6: Enter an XY Conversion Instruction on Rung 2

To enter an XY conversion instruction:

1. Click the XY Convert icon.

2. Click on Column 2 of Rung 2.

The XY Convert Format (XYFS) dialog box displays.

3. Click Convert X to Y (CVXTY) in the Type field.

4. Enter 1, 1, First Point in the S1 field.

5. Enter D10, D10, X value in D10 in the S2 field.

6. Enter D15, D15, Converted X value in D15 in the D1 field.

7. Click OK.

The CVXTY (I) instruction displays on Rung 2.
Step 7: Enter a Rung Comment for Rung 2

To enter a comment for Rung 2:

1. Click anywhere on Rung 2.

2. Click Edit ➔ Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

   The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter: **When I0 is On X value in D10 is converted to Y value and stored in D15.**

4. Click OK to save the comments.

The rung comment displays above Rung 2.
Step 8: Enter an NO Contact on Rung 3

To enter an NO contact:

1. Click the **Normally Open** icon.

2. Click on Column 1 of Rung 3.

The **Normally Open** (LOD) dialog box displays.

3. Enter **I1** in the **Tag Name:** and **Allocation Number:** fields.

4. Enter **Convert Y to X** in the **Name Comment:** field.

5. Click **OK**.

The **Convert Y to X** contact displays at the beginning of Rung 3.
Step 9: Enter an XY Conversion Instruction on Rung 3

To enter an XY conversion instruction:

1. Click the XY Convert icon.
2. Click on Column 2 of Rung 3.

The XY Convert Format (XYFS) dialog box displays.

3. Click Convert Y to X (CVYTX) in the Type field.
4. Enter 1, 1, First Point in the S1 field.
5. Enter D11, D11, Y value in D11 in the S2 field.
6. Enter D20, D20, Converted Y value in D20 in the D1 field.
7. Click OK.

The CVYTX (I) instruction displays on Rung 3.
Step 10: Enter a Rung Comment for Rung 3

To enter a comment for Rung 3:

1. Click anywhere on Rung 3.

2. Click Edit → Rung comment… from the Menu Commands toolbar or click the Rung comment icon.

   The Rung Comment Dialog screen displays.

3. Click in the white area of the dialog screen and enter: **When I1 is on the Y value in D11 is converted to x value and stored in d20.**

4. Click OK to save the comments.

The rung comment displays above Rung 3.
**Step 11: Save the Program**

To name and save this program:

1. Click **File → Save as…** from the Menu Commands toolbar.

   The **Save As** dialog box displays.

2. Enter **XYCONVS** in the **File Name:** field of the dialog box.

3. Click the **OK** button or press the **[ENTER]** key to confirm.

**Step 12: Print, Transfer, and Monitor the Program**

To print this program, please refer to page 4-9. To transfer this program, refer to page 4-12. To monitor this program, refer to page 4-13.
**FA3S Series XY Conversion**

**XY Format Set (XYFS) Linear Conversion**

Format using XY coordinates.

When an input is on, the format for XY conversions are set. The number of (Xn, Yn) coordinates, defining the linear relationship between X and Y, is designated by the operand NBLK (2 to 32 points).

Since the XYFS instruction is executed in each scan while an input is on, use a pulse input as required. The initialize pulse from special internal relay M304 is recommended to set the format when the CPU operation starts. A level input such as M317 will increase the scan time.

**Convert X to Y (CVXTY)**

Assign data S as X, find corresponding Y, and place in D.

Similar to the ONC PLC, the FA series has convert X to Y (CVXTY) and convert Y to X (CVYTX) instructions.
When an input is on, the value in the operand designated by \( S \) is assigned as \( X \). The linear relationship defined in the XYFS instruction is used to identify the corresponding \( Y \). This value is set to the operand designated by etc \( D \). The variable “\( dd \)” can be 30 to 39.

The XYFS instruction, which defines the linear relationship between \( X \) and \( Y \), must precede this instruction. The variable “\( dd \)” in the CVXTY instruction number equals 30 plus the value assigned to “\( LL \)” (from 0 to 9) for the corresponding XYFS instruction. For example, if the XYFS instruction number is 803, the CVXTY instruction number must be 833.

The XYFS instruction is only required once, to set up the (\( X, Y \)) coordinates for many conversions, both CVXTY and CVYTX. However, up to ten XYFS instructions can also be used to set up different linear relationships between \( X \) and \( Y \) for different conversions. When different XYFS instructions are used, make sure that the “\( dd \)” variable in the CVXTY instruction number corresponds to the appropriate “\( LL \)” variable for the desired XYFS instruction.

Values for the source (\( S = X \)) can be 0 to 9999, and values for the destination (\( D = Y \)) can be 9999 to 9999. The destination operand cannot be a timer or a counter. Conversion is only executed within the range that is defined by the XYFS instruction. If the \( X \) value exceeds the range designated to corresponding \( Y \) values, a numeric value error will occur.
INDEX

A

action items, identifying 1-2
adding (up) counters
   Micro-1 6-4
adjust (ADJ) instructions. See also
calendar/clock instructions 13-1
adjusting calendar/clock using a user
program 13-39
advanced instructions
   applicable PLCs 1-2
   destination operands 9-2, 9-3
   input condition 9-2
   source operands 9-2, 9-3
   structure of 9-1
   toolbar 3-3
analog potentiometers 11-1
   settings for 11-3, 11-9
analog read 0 (ANR0)
   dialog box 11-6
   instructions
      entering in a
      program 11-6, 11-7
      uses for 11-1
   ladder logic example 11-3
   programs, creating 11-4
   valid operands 11-2
analog read 1 (ANR1)
   instructions, uses for 11-2
   valid operands 11-2
analog to digital (A/D)

B

basic instructions
   applicable PLCs 1-2
   toolbar 3-3
batch monitor 4-15
binary arithmetic
   instructions
      addition
         (ADD) 10-1, 10-19, 10-20, 10-
         -21, 10-22, 10-23, 10-24
      division (DIV) 10-3
      multiplication (MUL) 10-2
      subtraction (SUB) 10-2, 10-3
      sum (SUM) 10-4
      uses for 10-1
   programs, creating 10-12, 10-27
   valid operands 10-5, 10-11

C

calendar read (CALR) 13-2
index ii

index ii

WindLDR™ Tutorial

calendar read (CALR)
instructions 13-1–13-2
entering in a program 13-9
ladder logic example 13-2
uses for 13-1
calendar write (CALW)
instructions 13-1
calendar/clock
adjusting using a user program 13-39
dialog box 13-9
FUN28 13-1
instructions
adjust (ADJ) 13-1
calendar read (CALR) 13-2
calendar write (CALW) 13-1
clock read (CLKR) 13-1, 13-3
clock write (CLKW) 13-1
programs, creating 13-6
setting 13-36
setting using a user program 13-27, 13-37
setting using WindLDR 13-25
valid operands 13-4
central processing unit (CPU). See also PLC components 2-3
clock function
MicroSmart 13-35
clock read (CLKR)
instructions 13-1, 13-3
entering in a program 13-11
ladder logic example 13-3
uses for 13-1
clock write (CLKW) instructions 13-1
closing a program 4-10
CNT 6-1
Coil Selection Dialog box 3-3
communication error 4-16
communication port, configuring 4-11
communication settings 3-8
Communication Settings dialog box 4-11
comparison
instructions
compare equal to (CMP=) 10-7
compare greater than (CMP>) 10-9
compare greater than or equal to (CMP≥) 10-10
compare less than (CMP 10-8
compare less than or equal to (CMP≤) 10-9
compare not equal to (CMP≠) 10-8
interval compare greater than or equal to (ICMP≥) 10-10
uses for 10-7
programs, creating 10-12, 10-27
Comparison dialog box 11-7
Configuration dialog box 3-5
configure PLC function/communication settings 3-8
Constant (Character) dialog box 14-14, 14-16
counter comparisons 6-6
CC=, uses for 6-6
CC≥, uses for 6-6
dialog box 6-15
errors (ERR1) 6-6
instructions, entering in a
program 6-15
programs, creating 6-8–6-17
counter tables 6-3
counters 6-1
designating as the destination operand 9-4
designating as the source operand 9-3
dialog box 6-11
instructions, entering in a program 6-11–6-12
Micro-1
adding (up) counters 6-4
dual-pulse reversible counter 6-4–6-5
up/down selection reversible counter 6-5
preset value designated using a data register 9-4
preset values 9-4
programs, creating 6-8–6-17
reversible vs. adding 6-1
uses for 6-1
CPU (central processing unit). See also PLC components 2-3
create
line 3-4
Cross Reference 4-9

D

data entry instructions 1-2
data move 9-1
data registers
designating as source or destination operand 9-3

transmit/receive 14-5
defining ladder logic input conditions 2-8
delete
line 3-4
deleting
columns 3-4
instructions 3-4
ladder lines 3-4
rungs 3-4
destination operand. See also advanced instructions, structure of 9-2, 9-3
dialog boxes
Analogue Read 0 (ANR0) 11-6
Clock/Calendar Read 13-9
Coil Selection 3-3
Communication Settings 4-11
Configuration 3-5
Constant (Character) 14-14, 14-16
Counter (CNT) 6-11
Counter Comparison 6-15
Download 4-12
Function Area Settings 3-8
Ladder Preferences 3-12
NC 5-18
Normally Open (LOD) 4-5
Output (OUT) 4-6
PLC Selection 3-5
Reset (RST) 7-9
Rung Comment 4-7
Set (SET) 7-6
Timer (TIM) 5-11
Timer (TML) 5-19
Transmit/Receive (TXD/RXD) 14-14
INDEX IV

Variable (Data Register) 14-15
Download Dialog box 4-12
downloading programs to a PLC. See also transferring programs to a PLC 4-12
draw
   line 3-4
drawing ladder lines 4-23
dual-pulse reversible counters
   Micro-1 6-4–6-5

E

Electrically Erasable Read Only Memory (EEPROM) 2-3
ENTER key 1-2
entering instructions in the PLC. See also programming device 2-3
errors (ERR1)
   counter comparisons 6-6
E-Stop contact, entering 4-22
examine-on/off contacts, defined 2-8
executing programmed logic 2-7

F

FA Series
   binary arithmetic instructions 10-19
   comparison instructions 10-24
   Function Area and Communications Settings dialog boxes 3-10
FA1J/2J
timers 5-1
FA3S Series
   transmit (TXD) instructions 14-22
   FA3S-CP11/-CP11T
timers 5-1
   FA3S-CP12/13
timers 5-1
   Function Area Settings dialog box 3-8
   function settings 3-8
   configuring 3-8

H

hot keys 3-3

I

"if" statement, defined 2-8
input
   condition for advanced instructions 9-2
   designating as a source or destination operand 9-4
   devices 2-4
   instructions. See also load (LOD) and load not (LODN) 4-1
   interface. See also PLC components 2-4
   reading 2-7
input/output program, creating. See also load/out program, creating 4-3–4-7
installing WindLDR 3-1
instructions
   accessing toolbars 3-3
instructions, accessing
  advanced instructions toolbar 3-3
  basic instructions toolbar 3-3
  hot keys 3-3
  right mouse button 3-4

memory. See also PLC
  components 2-3
  menu commands toolbar 3-2
  menu instructions 1-2

Micro-1
  counters
    adding (up) counters 6-4
    dual-pulse reversible counter 6-4–6-5
    up/down selection reversible counter 6-5
  Function Area and Communications
  Settings dialog boxes 3-9
  single output (SOT) instructions 8-2
  timers 5-1

Micro3
  analog potentiometers 11-1
  binary arithmetic instructions 10-1
  calendar/clock instructions 13-1
  comparison instructions 10-7
  counter table 6-3
  counters 6-2
  data move 9-1
  Function Area and Communications
  Settings dialog boxes 3-9
  input allocation table 9-5
  single output down (SOTD) instructions 8-1–8-2
  single output up (SOTU) instructions 8-1–8-2
  timers 5-3–5-6

Micro3C
  analog potentiometer 11-1
  binary arithmetic instructions 10-1
  calendar/clock instructions 13-1
  comparison instructions 10-7

L

ladder lines, drawing 4-23
ladder logic
  defined 2-5
  external field devices 4-19
  programming 2-6
  solving 2-7
ladder preferences
  dialog box 3-12
  selecting font 3-12
  selecting rung display and coil parameters 3-12
Ladder Preview 4-8
latch commands. See also SET instructions 7-1
light emitting diode (LED) 2-4
  line
    create 3-4
load (LOD) instruction 4-1
load not (LODN) instruction 4-1
load/out program, creating. See also input/output program, creating 4-7
logic, defined. See also ladder logic 2-1

M

maximizing screens 3-3
INDEX VI

counter table 6-3
counters 6-2
data move 9-1
Function Area and Communications
Settings dialog boxes 3-9
input allocation table 9-5
receive (RXD) instructions 14-3, 15-3, 15-4, 15-5
single output down (SOTD) instructions 8-1–8-2
single output up (SOTU) instructions 8-1–8-2
timers 5-3–5-6
transmit (TXD) instructions 14-1

MicroSmart
analog potentiometer 11-1
binary arithmetic instructions 10-1
clock function 13-35
counter table 6-3
counters 6-2
data move 9-1
input allocation table 9-5
single output down (SOTD) 8-1–8-2
single output up (SOTU) 8-1–8-2
timers 5-5
transmit (TXD) instructions 14-1
monitor PLC status 4-14
monitoring a program 4-13
move (MOV) instructions
repeat operation 13-4–13-5
move instruction 9-1
multiple load/out program, creating 4-26
multiple start/stop program, creating 4-20

N
naming a program 4-8
NC dialog box 5-18
Normally Closed
dialog box 5-18
normally closed (NC) contact 4-1
conventions 4-18
inputs 4-19
normally open (NO) contact 4-1
conventions 4-18
dialog box 4-5
entering in a program 4-4–4-5

O
ONC
binary arithmetic instructions 10-1
clock and calendar 13-20
comparison instructions 10-7
counter table 6-3
counters 6-2
data move 9-1
Function Area and Communications
Settings dialog boxes 3-10
input allocation table 9-5
real-time clock and calendar 13-20
single output down (SOTD) instructions 8-1
single output up (SOTU) instructions 8-1
timers 5-4
transmit (TXD) instructions 14-1
1 msec timedown timer (TMS) 5-3
100 msec timedown timer (TIM) 5-3
On-Line
Help 4-7, 4-8, 4-9, 4-24, 4-25, 4-26
on-line help, accessing 1-3
on-line monitoring 4-13
Online option 4-14
opcode. See also advanced
instructions, structure of 9-1
operands
destination 9-2, 9-3
source 9-2, 9-3
word 9-3
optical isolation, defined 2-4
output
designating as a source or destination
operand 9-4
dialog box 4-6
entering in a program 4-6
interface. See also PLC
components 2-4–2-5
updating 2-8
output (OUT) instruction 4-2
output not (OUTN) instruction 4-2

P

PLC
components 2-2–2-5
central processing unit
(CPU) 2-3
input interface 2-4
memory 2-3
output interface 2-4–2-5
power supply 2-3
programming device 2-3
defined 2-1

function/communication
settings 3-8
scanning 2-6–2-8
communications and
housekeeping 2-8
defined 2-6
executing programmed logic 2-7
reading inputs 2-7
updating outputs 2-8
selecting 3-5–3-7

PLC Selection dialog box 3-5
PLC status 4-14
point write 4-17
power supply. See also PLC
components 2-3
printing a program 4-9
programming device. See also PLC
components 2-3

R

Random Access Memory (RAM) 2-3
receive (RXD)
data registers 14-5
instructions 14-3, 15-3, 15-4, 15-5
repeat
cycles. See also advanced instructions,
structure of 9-1
designation. See also advanced
instructions, structure of 9-1

Reset (RST)
dialog box 7-9
instructions
to enter in a program 7-9
uses for 7-1
INDEX VIII

INDEX © 2001 IDEC Corporation

WindLDR™ Tutorial

programs, creating 7-2–7-10
right mouse button
accessing instructions 3-4
deleting rungs, columns, ladder lines 3-4
Root 10-3
RST instructions 7-1
Rung Comment Dialog screen 4-7
rung comment, entering in a program 4-7

S

saving a program 4-8
screens, maximizing 3-3
selecting PLC type 3-5–3-7
serial port settings, selecting 4-11
SET
    dialog box 7-6
    instructions
    entering in a program 7-6
    uses for 7-1
    programs, creating 7-2–7-10
setting calendar/clock 13-36
setting calendar/clock using a user program 13-27, 13-37
setting calendar/clock using WindLDR 13-25
single output (SOT)
    instructions
    entering in a program 8-6, 8-11, 9-7, 9-8, 9-12, 9-13
    uses for 8-2

T

technical support, accessing 1-3
10 msec timedown timer (TMH) 5-3
timer comparison program 5-15
timers
    designating as the destination operand 9-4
    designating as the source operand 9-3
dialog box 5-11, 5-19
FA3S-CP12/13 5-1
equations (TIM), entering in a
program 5-11, 5-19, 5-20
Micro-1, FA1J/2J, FA3S-CP11/
-CP11T 5-1
Micro3 5-3–5-6
Micro3C 5-3–5-6
MicroSmart 5-5
ONC 5-4
preset value designated using a data
register 9-4
preset values 9-4
programs,
creating 5-7, 5-14, 5-21, 5-23
uses for 5-1
toolbars
advanced instructions 3-2
basic instructions 3-2
menu commands 3-2
transferring programs to a PLC. See
also downloading programs to a
PLC 4-12
transmit (TXD)
data registers 14-5
dialog box 14-14
instructions 14-1, 14-22, 14-24, 14
-25, 14-27, 15-1, 15-17
entering in a program 14-13
programs, creating 14-6, 15-6
valid
operands 14-2, 14-3, 15-1, 15-2, 15
-3, 15-4, 15-5
tutorial conventions 1-1–1-3

U
un latch command. See also RST
command 7-1
un latch commands. See also Reset
(RST) instructions 7-1
up/down selection reversible counters
Micro-1 6-5
user program
adjusting calendar/clock 13-39
setting calendar/clock 13-27, 13-37

V
Variable (Data Register) dialog
box 14-15

W
WindLDR
setting calendar/clock 13-25
WindLDR installation
CD 3-1
instructions 3-1–3-2
word operands 9-3