

Mechatronics

**Programmable Logic Controller
Basic Programming**

Courseware Sample

52281-F0

Order no.: 52281-10
First Edition
Revision level: 08/2015

By the staff of Festo Didactic

© Festo Didactic Ltée/Ltd, Quebec, Canada 2015
Internet: www.festo-didactic.com
e-mail: did@de.festo.com

Printed in Canada

All rights reserved

ISBN 978-2-89747-462-1 (Printed version)

ISBN 978-2-89747-464-5 (CD-ROM)

Legal Deposit – Bibliothèque et Archives nationales du Québec, 2015

Legal Deposit – Library and Archives Canada, 2015

The purchaser shall receive a single right of use which is non-exclusive, non-time-limited and limited geographically to use at the purchaser's site/location as follows.

The purchaser shall be entitled to use the work to train his/her staff at the purchaser's site/location and shall also be entitled to use parts of the copyright material as the basis for the production of his/her own training documentation for the training of his/her staff at the purchaser's site/location with acknowledgement of source and to make copies for this purpose. In the case of schools/technical colleges, training centers, and universities, the right of use shall also include use by school and college students and trainees at the purchaser's site/location for teaching purposes.

The right of use shall in all cases exclude the right to publish the copyright material or to make this available for use on intranet, Internet and LMS platforms and databases such as Moodle, which allow access by a wide variety of users, including those outside of the purchaser's site/location.

Entitlement to other rights relating to reproductions, copies, adaptations, translations, microfilming and transfer to and storage and processing in electronic systems, no matter whether in whole or in part, shall require the prior consent of Festo Didactic GmbH & Co. KG.

Information in this document is subject to change without notice and does not represent a commitment on the part of Festo Didactic. The Festo materials described in this document are furnished under a license agreement or a nondisclosure agreement.

Festo Didactic recognizes product names as trademarks or registered trademarks of their respective holders.

All other trademarks are the property of their respective owners. Other trademarks and trade names may be used in this document to refer to either the entity claiming the marks and names or their products. Festo Didactic disclaims any proprietary interest in trademarks and trade names other than its own.

Safety and Common Symbols

The following safety and common symbols may be used in this manual and on the equipment:

Symbol	Description
	DANGER indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.
	WARNING indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.
	CAUTION indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.
	CAUTION used without the <i>Caution, risk of danger</i> sign  , indicates a hazard with a potentially hazardous situation which, if not avoided, may result in property damage.
	Caution, risk of electric shock
	Caution, hot surface
	Caution, risk of danger
	Caution, lifting hazard
	Caution, hand entanglement hazard
	Notice, non-ionizing radiation
	Direct current
	Alternating current
	Both direct and alternating current
	Three-phase alternating current
	Earth (ground) terminal

Safety and Common Symbols

Symbol	Description
	Protective conductor terminal
	Frame or chassis terminal
	Equipotentiality
	On (supply)
	Off (supply)
	Equipment protected throughout by double insulation or reinforced insulation
	In position of a bi-stable push control
	Out position of a bi-stable push control

Table of Contents

Preface.....	XI
About This Manual	XIII
To the Instructor	XV
Exercise 1 Familiarization with the PLC Trainer and RSLogix Micro.....	1
DISCUSSION.....	1
Introduction.....	1
PLC Hardware.....	3
Power Supply section	3
Input section	3
Output section.....	4
Central Processing Unit	4
Memory section (Input and Output)	4
Status Indicators	6
The PLC Trainer	7
The RSLogix Micro Software	10
Running RSLogix Micro	10
Projects.....	12
Configuring System Communications	14
Editing System Preferences.....	15
PROCEDURE	16
Running RSLogix Micro.....	16
Editing the Project Files Search Path.....	16
Configuring System Communications	16
Creating a New Project	17
The Project Tree.....	17
The Instruction Toolbar	18
The Online Section.....	18
Saving a Project to a File	18
Exercise 2 Online Operations and Monitoring I/O Data Files.....	21
DISCUSSION.....	21
Data file organization.....	21
Input and output data files	22
PLC input and output interfaces.....	24
PLC input interface	24
PLC output interface	25
PLC modes of operation.....	26
PROCEDURE	28
Set up and connections.....	28
Downloading a ladder program to the PLC processor	31
Input data file	32
Output data file	35
Word level.....	37
Clearing the PLC Memory	41

Table of Contents

Exercise 3	Relay Instructions Part 1	43
	DISCUSSION.....	43
	Introduction.....	43
	Input instructions	43
	Examine if Closed (XIC).....	43
	Examine if Open (XIO).....	44
	Output instruction	45
	Output Energize (OTE)	46
	PROCEDURE	47
	Set up and connections.....	47
	Ladder program using XIC and OTE instructions controlled by an NO push button.....	49
	Ladder program using XIC, XIO, and OTE instructions controlled by NO push buttons	51
	Ladder program using XIC, XIO, and OTE instructions controlled by NC push buttons	54
	Developing a program to control an output using two push buttons.....	56
	Developing a program to control an output using four push buttons.....	57
	Series (AND) and parallel (OR) logics	58
	Developing a program to control three outputs using one push button.....	60
	Developing a program to control three outputs using two push buttons.....	61
Exercise 4	Relay Instructions Part II	65
	DISCUSSION.....	65
	Introduction.....	65
	Latching instructions.....	65
	Output Latch (OTL)	65
	Output Unlatch (OTU)	66
	Latching instruction operation	66
	PROCEDURE	67
	Set up and connections.....	67
	Ladder program using latching instructions	67
	Developing a program acting as a single Start/Stop station using latching instructions.....	70
	Developing a program acting as a multiple Start/Stop station using latching instructions.....	71
	Developing a program acting as a Forward/Reverse control circuit using holding contacts	72
	Developing a program acting as a Forward/Reverse control circuit using latching instructions.....	74

Table of Contents

Exercise 5	Timer Instructions.....	77
	DISCUSSION.....	77
	Introduction.....	77
	Timer instruction parameters.....	78
	Timer.....	78
	Time base.....	78
	Preset.....	78
	Accumulator (Accum).....	78
	Timer status bits.....	79
	Timer Enable (EN).....	80
	Timer Timing (TT).....	80
	Done (DN).....	80
	Timer instructions.....	80
	Timer-on-Delay (TON).....	80
	Timer-off-Delay (TOF).....	81
	Retentive Timer-on-Delay (RTO).....	81
	Reset (RES) instruction.....	82
	PROCEDURE.....	83
	Set up and connections.....	83
	The Timer-on-Delay (TON) instruction.....	83
	The Timer-off-Delay (TOF) instruction.....	87
	The Retentive Timer-on-Delay (RTO) instruction.....	89
	Developing a flasher circuit using two timer instructions.....	91
	Developing a flasher circuit using three timer instructions.....	92
Exercise 6	Counter Instructions.....	97
	DISCUSSION.....	97
	Introduction.....	97
	Counter instructions parameters.....	97
	Address.....	97
	Preset.....	98
	Accumulator (Accum).....	98
	Counter status bits.....	98
	Count Up Enable (CU).....	99
	Count Down Enable (CD).....	99
	Done (DN).....	99
	Overflow (OV).....	100
	Underflow (UN).....	100
	Counter instructions.....	100
	Count Up (CTU).....	100
	Count Down (CTD).....	100
	Reset (RES) instruction.....	101
	PROCEDURE.....	101
	Set up and connections.....	101
	The Count Up (CTU) instruction.....	102
	The Count Down (CTD) instruction.....	107

Table of Contents

Exercise 7	Comparison Instructions	113
	DISCUSSION	113
	Introduction	113
	Comparison instruction parameters	114
	Source A	114
	Source B	114
	Low Limit	114
	High Limit	114
	Test	114
	Comparison instructions	114
	Equal (EQU)	114
	Not Equal (NEQ)	115
	Less Than (LES)	115
	Less Than or Equal (LEQ)	115
	Greater Than (GRT)	115
	Greater Than or Equal (GEQ)	116
	Limit (LIM)	116
	Using comparison instructions to perform sequential control	116
	PROCEDURE	117
	Set up and connections	117
	The Equal (EQU) instruction	117
	The Not Equal (NEQ) instruction	121
	The Greater Than (GRT) instruction	124
	The Greater Than or Equal (GEQ) instruction	127
	The Less Than (LES) instruction	129
	The Less Than or Equal (LEQ) instruction	131
	The Limit (LIM) instruction	133
	Counter-driven comparison instructions	135
	Timer-driven comparison instructions	137
	Developing a traffic light simulation program using comparison instructions	139
Exercise 8	Move Instructions	143
	DISCUSSION	143
	Introduction	143
	Move instruction parameters	143
	Source	143
	Destination	144
	Mask	144
	Move instructions	144
	Move (MOV)	144
	Move with Mask (MVM)	144

Table of Contents

PROCEDURE	145
Set up and connections	145
The Move (MOV) instruction using a constant as source value	146
The Move (MOV) instruction using a value from an addressed word as source value	149
Developing a program using move instructions to control outputs	152
Developing a program using comparison and move instructions to control six outputs	157
Developing a traffic light simulation program using move instructions	159
The Move with Mask (MVM) instruction	161
Exercise 9 Sequencer Instructions	167
DISCUSSION	167
Introduction	167
Sequencer instruction parameters	168
File	168
Mask	168
Source	168
Destination	168
Control	168
Length of the sequencer file	168
Position	169
Sequencer status bits	169
Enable (EN)	169
Done (DN)	169
Error (ER)	169
Found (FD)	170
Sequencer instructions	170
Sequencer Output (SQO)	170
Sequencer Compare (SQC)	172
Reset instruction	174
PROCEDURE	174
Set up and connections	174
The Sequencer Output (SQO) instruction	174
The Sequencer Compare (SQC) instruction	179
Developing a program using SQO and TON instructions	183
Appendix A Equipment Utilization Chart	187
Appendix B Glossary of New Terms	189
Appendix C Conversion Table	191

Table of Contents

Appendix D Boolean Algebra and Digital Logic	193
Appendix E PLC Trainer, Model 3240-A	197
Appendix F PLC Trainer, Model 3240-D	205
Appendix G PLC Trainer, Model 3240-3.....	213
Index.....	217
Acronyms	219
Bibliography	221

Preface

Programmable logic controllers (PLCs) are at the center of a multitude of automated systems. Some industries that benefit from these systems include manufacturing plants, breweries, wastewater treatment plants, mining facilities, and automotive assembly plants.

This manual, Programmable Logic Controller - Basic Programming, introduces students to PLC ladder program editing using the most common PLC instructions: relay instructions, timer instructions, counter instructions, comparison instructions, move instructions, and sequencer instructions. The software used to edit the ladder programs is the RSLogix Micro software from Rockwell. The edited programs are downloaded to an Allen-Bradley PLC and tested by activating PLC inputs and monitoring the PLC outputs status indicators.

We invite readers of this manual to send us their tips, feedback, and suggestions for improving the book.

Please send these to did@de.festo.com.

The authors and Festo Didactic look forward to your comments.

About This Manual

Manual objectives

When you have completed this manual, you will be able to navigate through the RSLogix Micro software from Rockwell and use the online resources. You will know how to develop ladder programs and use the most common PLC instructions.

Utilization of this manual with various PLC trainers

This manual was written specifically for the PLC Trainer, Model 3240-4, which uses an Allen-Bradley MicroLogix™ 1200 controller.

However, the manual can also be used with the PLC Trainer, Model 3240-A, that uses an Allen-Bradley MicroLogix™ 1100 controller, the PLC Trainer, Model 3240-D, that uses an Allen-Bradley MicroLogix™ 1100 controller, or the PLC Trainer, Model 3240-3, that uses an Allen-Bradley MicroLogix 1500 controller.

- If you are using the PLC Trainer, Model 3240-A, skip the DISCUSSION section entitled *The PLC Trainer* of Exercise 1, and refer to Appendix E instead for a description of Model 3240-A.
- If you are using the PLC Trainer, Model 3240-D, skip the DISCUSSION section entitled *The PLC Trainer* of Exercise 1, and refer to Appendix F instead for a description of Model 3240-D.
- If you are using the PLC Trainer, Model 3240-3, skip the DISCUSSION section entitled *The PLC Trainer* of Exercise 1, and refer to Appendix G instead for a description of Model 3240-3.

Safety considerations

Safety symbols that may be used in this manual and on the equipment are listed in the Safety Symbols table at the beginning of the manual.

Safety procedures related to the tasks that you will be asked to perform are indicated in each exercise.

Make sure that you are wearing appropriate protective equipment when performing the tasks. You should never perform a task if you have any reason to think that a manipulation could be dangerous for you or your teammates.

Reference material

The Programmable Logic Controllers from Frank D. Petruzella (McGraw-Hill Ed., 2010) is a reference to complement and expand on PLCs.

The User's manuals provided with the Allen-Bradley PLC is also another reference.

About This Manual

Prerequisite

As a prerequisite to this course, you should have a basic understanding of electricity. It is assumed that you have a general understanding of the following concepts:

- DC and ac voltage, current, power
- Series and parallel circuits
- NO and NC contacts
- Push buttons, toggle switches, relays

Systems of units

Units are expressed using the International System of Units (SI) followed by the units expressed in the U.S. customary system of units (between parentheses).

To the Instructor

You will find in this Instructor Guide all the elements included in the Student Manual together with the answers to all questions, results of measurements, graphs, explanations, suggestions, and, in some cases, instructions to help you guide the students through their learning process. All the information that applies to you is placed between markers and appears in red.

Accuracy of measurements

The numerical results of the hands-on exercises may differ from one student to another. For this reason, the results and answers given in this manual should be considered as a guide. Students who correctly performed the exercises should expect to demonstrate the principles involved and make observations and measurements similar to those given as answers.

Software requirements

In order for students to be able to perform the exercises in this manual, the RSLogix Micro, and the RSLinx Classic software must have been previously installed on the student's computer, according to the manufacturer's instructions.

Configuration of RSLinx Classic

In RSLinx Classic, configure the driver for an AB_DF1 device (communication device for a Micrologix PLC) at station 0. Select the serial or com port of the computer to which the Micrologix PLC will be connected through the serial 1761-CBL cable. Set the baud rate to 19 200. The recommended protocol for the Micrologix PLCs is full-duplex.



If the PLC used on your trainer is the Allen-Bradley Micrologix 1100 controller™, and you are using an Ethernet communication link, configure the driver for an Ethernet/IP device. The system must be configured via the BOOTP (bootstrap protocol). (Refer to the section "Connecting to network via Ethernet Interface" of the Allen-Bradley User Manual.)

You are now ready to use RSLogix Micro to program and control the PLC. To start RSLogix Micro, select the corresponding command in the Rockwell Software program group.

Sample Exercise
Extracted from
the Student Manual
and the Instructor Guide

Timer Instructions

EXERCISE OBJECTIVE When you have completed this exercise, you will be familiar with the timer instructions and their use in PLC programs. Throughout this exercise, you will develop programs using these instructions to control various operations.

DISCUSSION OUTLINE The Discussion of this exercise covers the following points:

- Introduction
- Timer instruction parameters
Timer. Time base. Preset. Accumulator (Accum).
- Timer status bits
Timer Enable (EN). Timer Timing (TT). Done (DN).
- Timer instructions
Timer-on-Delay (TON). Timer-off-Delay (TOF). Retentive Timer-on-Delay (RTO).
- Reset (RES) instruction

DISCUSSION

Introduction

The **Timer instructions** are frequently used in programs to control processes based on time constraints to achieve a specific operation. Such instances may be one of the following:

- To allow a specific amount of time to occur before an operation
- To allow a specific amount of time to occur after an operation
- To allow an operation to occur for a specific amount of time

In order to correctly use these instructions, we must investigate their operation and use in ladder logic programs.

Timer instruction parameters

The timer instructions have the following parameters: Timer, Time base, Preset, and Accumulator. These parameters must be programmed when entering a timer instruction.

Timer

The Timer parameter is the address of the timer file; it is located in the timer data file. The default timer file is T4 and it is created automatically at the time the program is developed. Figure 54 shows the content of the T4 – TIMER data file. Note that the content of this data file is similar to the input and output data files.

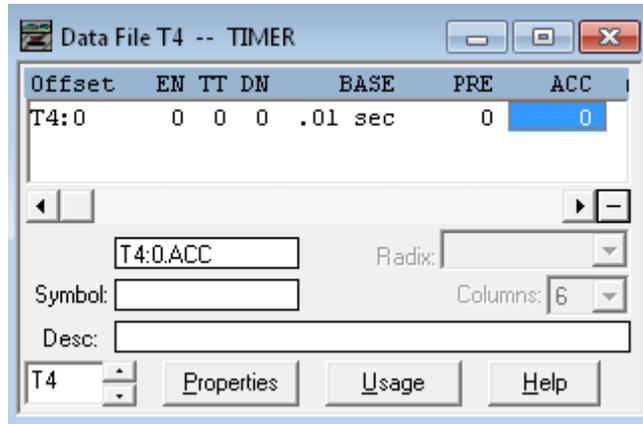


Figure 54. T4 – TIMER data file.

Time base

The time base is the duration of each time interval counted by the timer instruction (how often it updates its accumulator). The time base of your PLC trainer is selectable as 1.0, 0.01, and 0.001 second.

Preset

The preset value specifies the desired time value that the timer must reach. The preset value of your PLC trainer can be set between 0 and 32 767. This implies the following:

- For a time base of 0.01 s, time delays between 0 and 327.67 s can be obtained.
- For a time base of 1.0 s, time delays between 0 and 32 767 s can be obtained.

Accumulator (Accum)

The Accumulator value (Accum) parameter is the value accumulated since the timer instruction was last reset to zero. When true, the timer instruction updates the **accumulated value** continually until it reaches the preset value.

- For a Time Base of 1.0 s, the accumulator will increment every second until it reaches the preset.
- For a Time Base of 0.01 s, the accumulator will increment every 100th of a second until it reaches the preset.
- For a Time Base of 0.001 s, the accumulator will increment every 1000th of a second until it reaches the preset.

Timer status bits

Each timer requires three binary words (16 bits) to store the information required. The structure of the timer data file is shown in Table 8. Note that the three words in the timer data file are identified as:

- Status word (Word 0)
- Preset word (Word 1)
- Accumulator word (Word 2)

Table 8. Structure of the T4 – TIMER data file.

Word	B ₁₅	B ₁₄	B ₁₃	B ₁₂	B ₁₁	B ₁₀	B ₉	B ₈	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
Status word (0)	EN	TT	DN							Internal Use						
Preset word (1)	Preset Value (PRE)															
Accumulator word (2)	Accumulated Value (ACC)															

Word 0 is the status word and contains three status bits that provide important information on the timing process. The status bits are the Timer Enable (EN) bit, the Timer Timing (TT) bit, and the Done (DN) bit.

The status bits (EN, TT, and DN) of the timer instructions can be used in a ladder program to control relay instructions such as Examine if Closed (XIC) and Examine if Open (XIO) instructions. To do so, the relay-type instruction must be addressed to the status bit as shown below.

Addressing examples:

- T4:0/13 or T4:0/DN: Done bit
- T4:0/14 or T4:0/TT: Timer Timing bit
- T4:0/15 or T4:0/EN: Timer Enable bit

With the default data file defined as T4 - Timer, most timers will be addressed as T4:0 - T4:255. Remember that each timer requires three words of data. For this reason, sequential numbering is the best practice. If you skip a timer address those 3 words are used even if the timer is not.

Timer Enable (EN)

The Timer Enable (EN) status bit (bit number 15 of word 0) is set to logic state 1 when the rung containing the timer instruction is true. It is set to logic state 0 when the rung is false.

Timer Timing (TT)

- With a Timer-on-Delay (TON) instruction, the Timer Timing (TT) status bit is set to logic state 1 when the rung of the TON instruction is true and the accumulated value is less than the preset value. It is reset to logic state 0

when the rung of the TON instruction becomes false or when the accumulated value becomes equal to the preset value (Timer DN status bit set to logic state 1).

- With a Timer-off-Delay (TOF) instruction, the TT status bit is set to logic state 1 when the rung of the TOF instruction is false and the accumulated value is less than the preset value. It is reset to logic state 0 when the rung of the TOF instruction becomes true or when the accumulated value becomes equal to the preset value (Timer DN status bit set to logic state 0).
- With a Retentive Timer-on-Delay (RTO) instruction, the TT status bit is set to logic state 1 when the rung of the RTO instruction is true and the accumulated value is less than the preset value. It is reset to logic state 0 when the rung of the RTO instruction becomes false or when the accumulated value becomes equal to the preset value (Timer DN status bit set to logic state 1).

Done (DN)

- With a Timer-on-Delay (TON) instruction, the Done (DN) status bit is set to logic state 1 when the accumulated value becomes equal to the preset value, and reset to logic state 0 when the rung of the TON instruction becomes false.
- With a Timer-off-Delay (TOF) instruction, the DN status bit is set to logic state 1 when the rung of the TOF instruction is true, and reset to logic state 0 when the accumulated value becomes equal to the preset value.
- With an Retentive Timer-on-Delay (RTO) instruction, the DN status bit is set to logic state 1 when the accumulated value becomes equal to the preset value, and reset to logic state 0 when the associated Reset (RES) instruction becomes true.

Timer instructions

Your PLC Trainer has the following timer instructions: Timer-on-Delay (TON), Timer-off-Delay (TOF), and Retentive Timer-on-Delay (RTO). The PLC also has a Reset (RES) instruction to reset the accumulated value and the status bits.

Timer-on-Delay (TON)

The Timer-on-Delay (TON) instruction provides a time delay when the timer is energized. This instruction starts to count time base intervals when the rung in which it is contained goes from false to true. As long as the rung remains true, the TON instruction increases its accumulated value until it reaches the preset value. If the rung becomes false, the accumulated value is reset to zero regardless of whether the preset value has been reached.

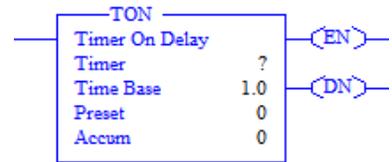
Mnemonic: TON

Parameters:

- Timer
- Time base
- Preset
- Accumulator (Accum)

Status Bits

- Enable (EN)
- Done (DN)



Timer-off-Delay (TOF)

The Timer-off-Delay (TOF) instruction starts to count time base intervals when the rung in which it is contained goes from true to false. As long as the rung remains false, the TOF instruction increases its accumulated value until it reaches the preset value. If the rung becomes true, the accumulated value is reset to zero regardless of whether the preset value has been reached. The operation of the TOF instruction is similar to the TON instruction; the difference is when the delay occurs. A TOF instruction provides a time delay when the timer is de-energized.

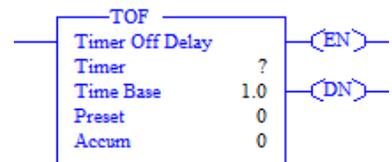
Mnemonic: TOF

Parameters:

- Timer
- Time base
- Preset
- Accumulator (Accum)

Status Bits

- Enable (EN)
- Done (DN)



Retentive Timer-on-Delay (RTO)

The Retentive Timer-on-Delay (RTO) instruction works in a way similar to that of the TON instruction: it starts to count time base intervals when the rung in which the RTO instruction is contained goes from false to true. As long as the rung remains true, the RTO instruction increases its accumulated value until it reaches the preset value.

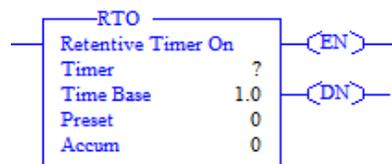
Mnemonic: RTO

Parameters:

- Timer
- Time base
- Preset
- Accumulator (Accum)

Status Bits

- Enable (EN)
- Done (DN)



Contrary to the TON instruction, however, the RTO instruction retains its accumulated value if the rung becomes false. When the rung becomes true

again, the RTO instruction resumes timing starting from the accumulated value. The accumulated value of the RTO instruction is also retained if the PLC is switched from the Run mode to another mode or turned off. In fact, the accumulated value of an RTO instruction is cleared by using a Reset (RES) instruction having the same address as the RTO instruction, as will be explained later in this discussion.

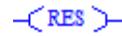
Reset (RES) instruction

The Reset (RES) instruction is used to reset the accumulated value of an RTO instruction. The RES instruction must be programmed with the same address as the RTO instruction to reset. When true, the RES instruction resets the accumulated value to zero.

Mnemonic: RES

Parameters:

- Address



PROCEDURE OUTLINE

The Procedure is divided into the following sections:

- Set up and connections
- The Timer-on-Delay (TON) instruction
- The Timer-off-Delay (TOF) instruction
- The Retentive Timer-on-Delay (RTO) instruction
- Developing a flasher circuit using two timer instructions
- Developing a flasher circuit using three timer instructions

PROCEDURE

Set up and connections

In this part of the exercise, you will make the required electrical connections and setup the PLC trainer.

1. Make sure that the PLC trainer is turned off.

On the PLC trainer, make the following connections:

- Push button 1 connects to PLC input 0.
- Push button 2 connects to PLC input 1.
- Toggle switch 5 connects to PLC input 4.

Make sure that the toggle switch 5 is set to the open position (downward).

2. Turn on the PLC trainer.

Make sure that the communication cable is connected between the PLC and the computer, and start the RSLinx Classic application.

The Timer-on-Delay (TON) instruction

In this part of the exercise, you will study the operation of the TON instruction through observation of the ladder program, the data files of the processor, as well as the status of the PLC outputs.

3. Start RSLogix Micro, and create a project having the following processor name: EXERC_5A.
4. Insert a rung that contains the following:
 - XIC instruction addressed to I:0/0
 - TON instruction with the following parameter settings:
 - Timer: T4:0
 - Time Base: 1.0
 - Preset: 5
 - Accumulator: 0

5. The created rung should look like the one shown in Figure 55.

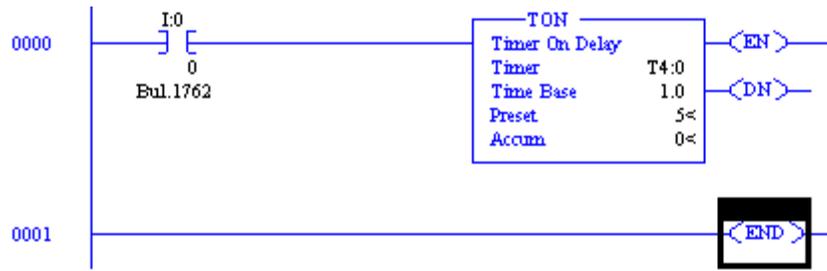


Figure 55. Rung with a Timer-on-Delay instruction.

6. Verify the rung using the Verify Rung command. Correct any errors.

Save your project in a file named EXERC_5A.RSS.

Download the program to the PLC.

Set the PLC to the Run mode and go online.

7. Verify the following operations:

- Press and maintain push button 1 to activate I:0/0. This should set the rung true and the timer accumulator should begin incrementing every second.
- Maintain push button 1 pressed until the accumulator stops incrementing.
- Release the push button.

8. Describe what happened to the accumulated value when push button 1 was released.

The accumulated value reset to 0 when push button 1 was released.

9. Open the T4 – TIMER data file from the Project tree.

Operate the timer by pressing push button 1 while observing how the logic state of the T4:0 status bits (EN, TT, and DN) varies. You will see in the next program that these status bits can be used to control other operations in a program.

10. Set the PLC to the Program mode and go offline.

11. Now you will modify the current program as shown in Figure 56.

Add three rungs to your program. In the first rung, enter the following instructions:

- XIC instruction addressed to T4:0/EN
- OTE instruction addressed to O:0/0

In the second rung, enter the following instructions:

- XIC instruction addressed to T4:0/TT
- OTE instruction addressed to O:0/1

In the third rung, enter the following instructions:

- XIC instruction addressed to T4:0/DN
- OTE instruction addressed to O:0/2

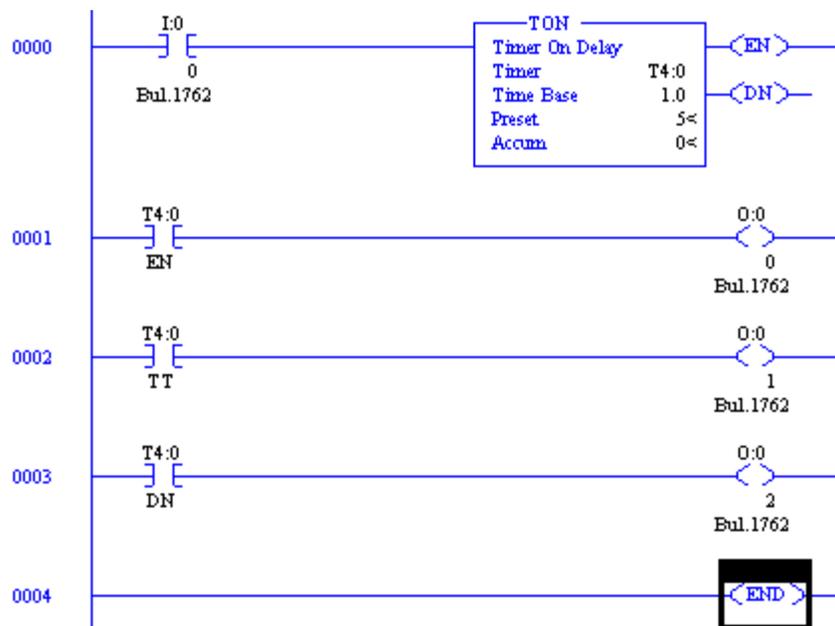


Figure 56. Ladder program using status bits EN, TT, and DN to control outputs.

12. Verify the rungs using the Verify Rung command. Correct any errors.

Change the processor name of your project to EXERC_5B.

Save your project in a file named EXERC_5B.RSS.

Download the program to the PLC.

Set the PLC to the Run mode and go online.

13. Verify the following operations:

- Pressing push button 1 sets the rung containing the Timer instruction to true, and the timer begins timing. After 5 seconds, the timer reaches its preset value and stops timing.
- While the timer is enabled, output 0 is activated (on). Note that when an output is activated (on), the corresponding PLC output lamp is turned on.
- When the timer is timing, output 1 is activated (on).
- When the accumulated value attains the preset value, output 2 activates.
- When push button 1 is released, the accumulated value of the timer and the status bits reset.

14. Open the T4 – TIMER data file from the Project tree. Operate push button 1 while monitoring the status bits of the timer T4:0.

What is the logic state of the EN status bit when the rung containing the TON instruction is true?

The logic state of the EN status bit is 1 when the rung containing the TON instruction is true.

15. What bits are at logic state 1 when the timer is timing?

The EN and TT status bits are at logic state 1 when the timer is timing.

16. Does the timing process occur when the push button is pressed or when it is released?

The timing process occurs when the push button is pressed.

17. Does the timer reset to 0 when the rung is false?

Yes. The timer resets to 0 when the rung is false.

18. When does the DN status bit change to logic state 1?

The DN status bit changes to logic state 1 when the timer preset and accumulated values are equal.

19. Set the PLC to the Program mode and go offline.

The Timer-off-Delay (TOF) instruction

In this part of the exercise, you will study the operation of the TOF instruction. The data structure for this timer is identical to the TON instruction, but the difference is when the delay occurs. A TOF instruction provides a time delay when the timer is de-energized.

20. Replace the TON instruction with a TOF instruction in your program. Set the parameters of the TOF instruction as shown in Figure 57.

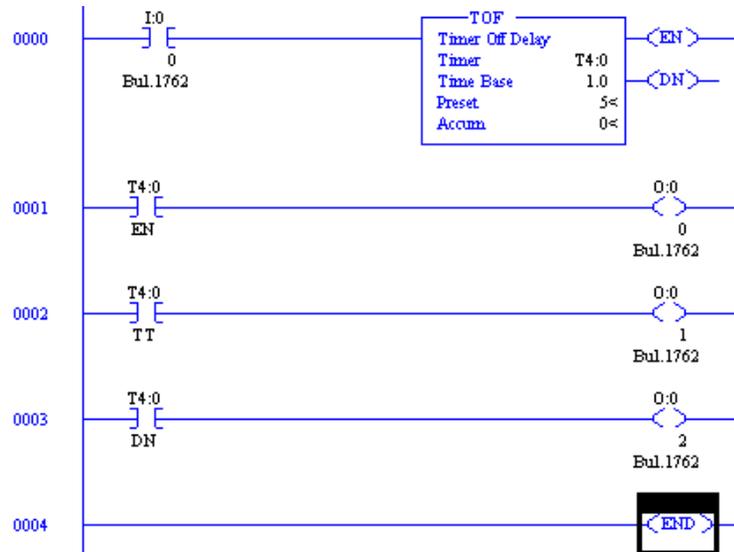


Figure 57. Ladder program using a TOF instruction.

21. Verify the rung using the Verify Rung command. Correct any errors.

Change the processor name of your project to EXERC_5C.

Save your project in a file named EXERC_5C.RSS.

Download the program to the PLC.

Set the PLC to the Run mode and go online.

22. Verify the following operations:

- Pressing push button 1 sets the rung containing the timer instruction to true.
- The timer instruction remains in this state until the push button is released, setting the rung false.
- When the rung goes false, the timer starts timing and when the accumulated value reaches the preset value, the DN status bit returns to logic state 0.
- While the timer is enabled, output 0 should be activated (on).

- When the timer is timing, output 1 should be activated (on).
- When the accumulated value attains the preset value, the timer and the status bits reset.

23. Open the T4 – TIMER data file from the Project tree.

Operate push button 1 and monitor the condition of the T4:0 status bits during operation.

What is the logic state of the EN status bit when the rung containing the TOF instruction is true?

The logic state of the EN status bit is 1 when the rung containing the TOF instruction is true.

24. What bits are at logic state 1 when the timer is timing?

The TT and DN status bits are at logic state 1 when the timer is timing.

25. Does the timing process occur when the push button is pressed or when it is released?

The timing process occurs when the push button is released.

26. Does the timer accumulator reset to 0 when the rung goes false?

No. The timer accumulator is reset to 0 when the rung goes true.

27. When does the DN status bit change to logic state 1?

The DN status bit changes to logic state 1 when the rung transitions from false to true.

28. Set the PLC to the Program mode and go offline.

The Retentive Timer-on-Delay (RTO) instruction

In this part of the exercise, you will study the operation of the Retentive Timer-on-Delay (RTO) instruction. The RTO instruction is a retentive timer and as the term suggests, it retains its accumulated value regardless of the rung condition.

29. Modify the current ladder program as follows:

- In rung 0000, replace the TOF instruction with a Retentive Timer-on-Delay (RTO) instruction having the same address. To do so, select instruction TOF T4:0 (Using the left button of the mouse) and choose the Change Instruction Type command in the context-sensitive menu (accessible using the right button of the mouse). Type RTO using the keyboard and press the Enter key. Do not modify any of the timer parameters.
- Select rung 0004 by clicking on "0004" at the left of the rung, then choose the Insert Rung command in the context-sensitive menu in order to insert a new rung before the END rung.
- In the newly created rung, insert instruction XIC I:0/1. Then, insert instruction RES T4:0. To do so, select the Timer/Counter category of instructions by clicking the corresponding selection tab. Click the RES button on the instruction toolbar to insert the instruction in the rung, type T4:0 using the keyboard, and press the Enter key.

The resulting program should be similar to the one shown in Figure 58.

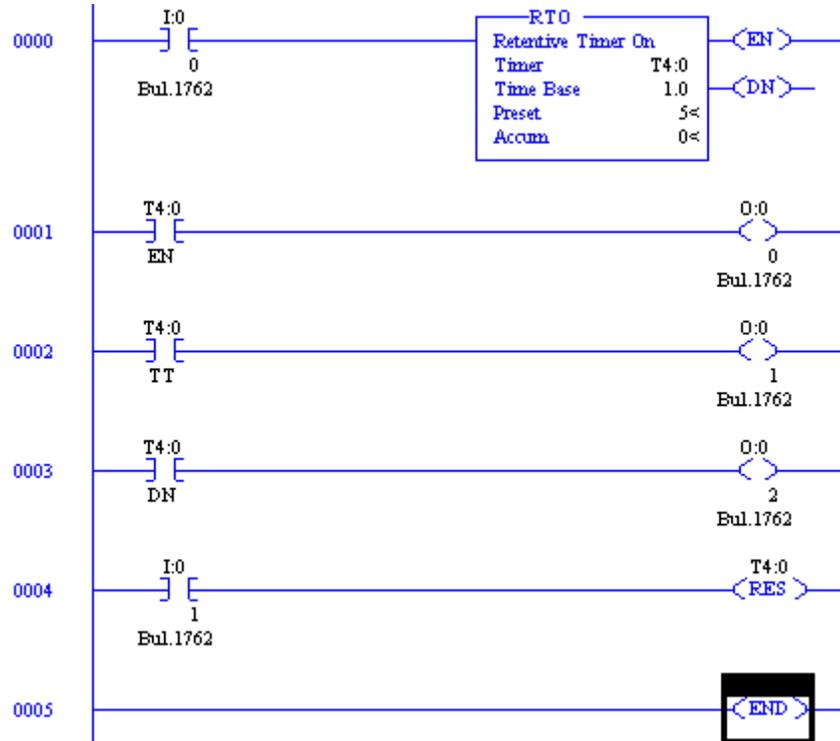


Figure 58. Ladder program using an RTO instruction.

30. Verify the rungs using the Verify Rung command. Correct any errors.

Change the processor name of your project to EXERC_5D.

Save your project in a file named EXERC_5D.RSS.

Download the program to the PLC.

Set the PLC to the Run mode and go online.

31. Verify the following operations:

- Pressing push button 1 sets the rung true and the timer starts timing.
- While the timer is enabled, output 0 should be activated (on).
- When the timer is timing, output 1 should be on.
- When the accumulated value attains the preset value, output 2 should activate.
- Releasing the push button will not reset the timer. The RTO instruction holds its accumulated value and resumes from that point when the rung goes true again (until the accumulated value equals the preset value).
- Pressing push button 2 energizes the RES instruction and resets the accumulated value and the status bits of the timer.

32. Open the T4 – TIMER data file from the Project tree.

Operate push button 1 and monitor the condition of the timer T4:0 status bits during operation.

What is the logic state of the EN status bit when the rung containing the timer instruction is true?

The logic state of the EN status bit is 1 when the rung containing the timer instruction is true.

33. What bits are at logic state 1 when the timer is timing?

The EN and TT status bits are at logic state 1 when the timer is timing.

34. Does the timing process occur when the push button is pressed or when it is released?

The timing process occurs when the push button is pressed.

35. Does the timer reset when the rung is false?

No. The timer does not reset when the rung is false.

36. When does the DN status bit change to logic state 1?

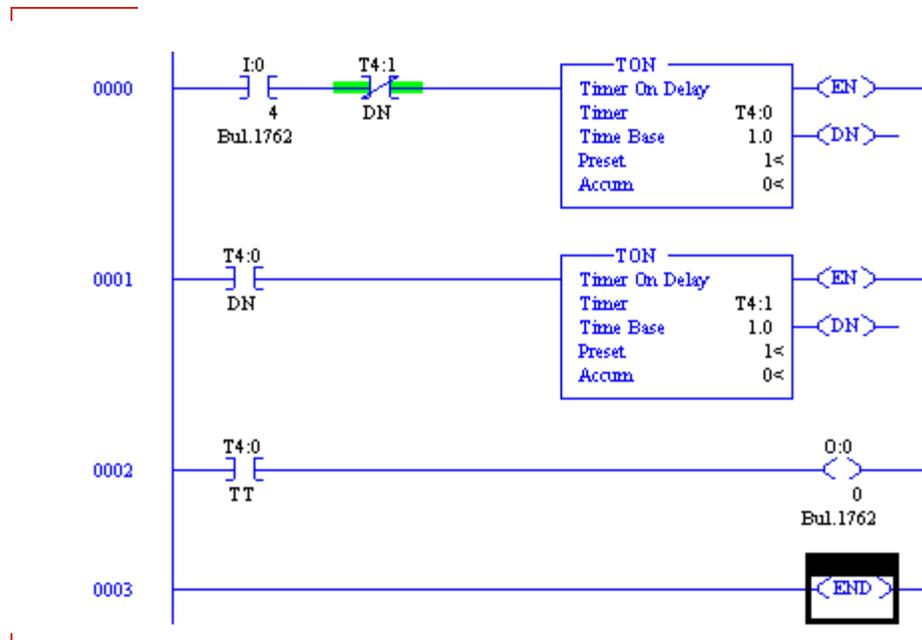
The DN status bit changes to logic state 1 when the preset and accumulated values are equal.

37. Set the PLC to the Program mode and go offline.

Developing a flasher circuit using two timer instructions

38. Using two timer instructions, develop a ladder program to cause a light to flash in a sequence as follows:

- The operation of the circuit is controlled by toggle switch 5 (connected to input 4).
- When input 4 is not activated, the timers are off and output 0 is deactivated.
- When input 4 is activated, timer T4:0 starts timing for 1 second.
- Output 0 is activated when timer T4:0 is timing (TT status bit at 1).
- When timer T4:0 is done, timer T4:1 starts timing for 1 second.
- When timer T4:1 is done, timer T4:0 resets and the cycle repeats until input 4 is deactivated.



- 39.** Verify the rungs using the Verify Rung command. Correct any errors.

Change the processor name of your project to EXERC_5E.

Save your project in a file named EXERC_5E.RSS.

Download the program to the PLC.

Set the PLC to the Run mode and go online.

- 40.** Test the program operation. Make sure that it operates as described in step 38.

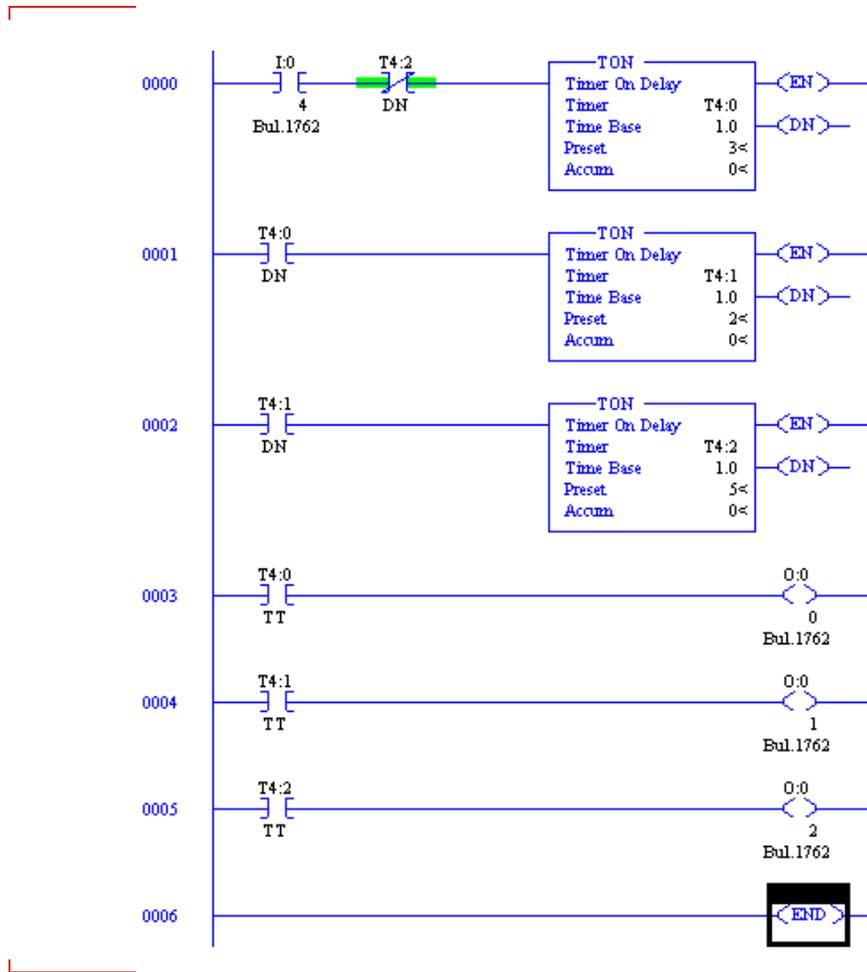
Once your program has been found operational, have your instructor verify it.

Set the PLC to the Program mode and go offline.

Developing a flasher circuit using three timer instructions

- 41.** Using three timer instructions, develop a ladder program to cause three lights to operate in a sequence as follows:

- Activating input 4 (using toggle switch 5) starts the process.
- Output 0 is activated for three seconds then deactivates.
- Then output 1 is activated for two seconds then deactivates.
- Then output 2 is activated for five seconds then deactivates.
- The cycle repeats until input 4 is deactivated.



- 42.** Enter the instructions in your program, and verify the rungs using the Verify Rung command. Correct any errors.

Change the processor name of your project to EXERC_5F.

Save your project in a file named EXERC_5F.RSS.

Download the program to the PLC.

Set the PLC to the Run mode and go online.

- 43.** Test the program operation. Make sure that it operates as described in step 41.

Once your program has been found operational, have your instructor verify it.

- 44.** In RSLogix Micro, set the PLC processor mode to Program, and clear the processor memory.

Close RSLogix Micro.

45. Make sure that all PLC inputs are deactivated.

Turn off the PLC trainer.

Remove all connection leads.

Return the equipment to the storage location.

CONCLUSION

In this exercise, you familiarized yourself with the timer instructions of the PLC trainer: the timer-on-delay (TON) instruction, the timer-off-delay (TOF) instruction, and the retentive timer-on-delay (RTO) instruction. You saw that these instructions have three parameters: a time base, an accumulated value, and a preset value.

The TON instruction counts time base intervals when the rung in which it is contained goes from false to true. The DN status bit is set to logic state 1 when the accumulated value becomes equal to the preset value, and to logic state 0 when the rung containing the TON instruction becomes false.

The TOF instruction counts time base intervals when the rung in which it is contained goes from false to true. The DN status bit is set to logic state 1 when the rung containing the TOF instruction becomes true, and to logic state 0 when the accumulated value becomes equal to the preset value.

The RTO instruction counts time base intervals when the rung in which it is contained is true. The DN status bit is set to logic state 1 when the accumulated value becomes equal to the preset value, and to logic state 0 when the associated Reset (RES) instruction becomes true.

REVIEW QUESTIONS

1. What is the time base of a timer instruction?

The time base of a timer instruction is the time interval which the accumulator will increment while timing.

2. What is the preset value of a timer instruction?

The preset value of a timer instruction is the desired total time interval of the timing operation.

3. Briefly describe the operation of the timer-on-delay (TON) instruction.

The TON instruction provides a time delay when the timer is energized. This instruction starts to count time base intervals when the rung in which it is contained goes from false to true. As long as the rung remains true, the TON instruction increases its accumulated value until it reaches the preset value. If the rung becomes false, the accumulated value is reset to zero regardless of whether the preset value has been reached or not.

4. Are the following statements about the timer-off-delay (TOF) instruction true or false? Explain.

"The TOF instruction counts time base intervals when the rung in which it is contained is false."

False. The TOF instruction counts time base intervals when the rung in which it is contained goes from false to true.

5. Explain how the TON and RTO instructions resemble each other and how they differ from each other in the way they work.

The TON and RTO instructions are similar except for the operation. The TON instruction resets its accumulator when the rung is false. The RTO instruction retains its accumulated value when the rung goes false. The RES instruction is used to clear the RTO accumulator.

Bibliography

Petruzella, Frank D., *Programmable Logic Controllers*, United States, 4th ed, McGraw-Hill Ed., 2010, ISBN-10: 0073510882, ISBN-13: 978-0073510880.