

Simatic S5 to RSLogix 5000 Application Conversion Guide



Application Solution

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication [SGI-1.1](#) available from your local Rockwell Automation sales office or online at <http://www.rockwellautomation.com/literature>) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.





In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING 	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.
ATTENTION 	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
SHOCK HAZARD 	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
BURN HAZARD 	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

RSLogix, RSLogix 5000, Logix5000, RSLinx, FactoryTalk View, CompactLogix, ControlLogix, GuardLogix, FlexLogix, DriveLogix, SoftLogix, Compact I/O, POINT I/O, ArmorPoint I/O, ArmorBlock MaXum, FLEX I/O, FLEX Ex, NetLinx, Integrated Architecture, Encompass, SLC 500, MicroLogix, PLC-2, PLC-3, PLC-5, SynchLink, and TechConnect are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Preface	Purpose	7
	Conversion versus Translation	7
	Terminology	8
	Additional Resources	8
	PLC Logic Conversion Services Provided by Rockwell Automation . .	9
	Service Features	9
	One-stop PLC Program Conversion Services	9
	Service Benefits	10
	Services Offered.	10
	Basic Conversion Package.	10
	Conversion Plus Initial Clean-up Package	10
	Additional Options	11
	Additional Program Conversions Available	11
Controllers and I/O Modules	Introduction	13
	S5 Controllers	14
	Logix Controllers	15
	Logix Controller Memory Use	16
	S5 Controller Memory Use	17
	Digital Input Modules	17
	Replacing 6ES5 420-4UA13/14 and 6ES5 430-4UA13/14 DI Modules	18
	Replacing 6ES5 432-4UA12 DI Modules.	19
	Replacing 6ES5 435-4UA12 DI Modules.	20
	Replacing 6ES5 436-4UB12 DI Modules.	20
	Digital Output Modules.	21
	Analog Input/Output Modules.	23
	Logix Controller with S5 I/O	27
	CompactLogix controllers.	27
	ControlLogix controllers	28
	S5 Controller with Logix I/O	29
Software Conversions - Program Structure	Introduction	31
	Organization Blocks/Logix Tasks.	31
	Continuous Task	32
	Periodic Task	33
	Event Task	34
	Task Properties	35
	RSLogix 5000 Programs	35
	Blocks/Routine	36
	Function Blocks/Add-On Instructions	38
	Watchdog Timer.	39
	I/O Addressing	39
	Logix Aliasing.	41
	Internal Memory Addressing.	43
	Flag Words/Bits.	43
	Data Blocks	44

	Programming Languages	48
	RSLogix 5000 Ladder Diagram	49
	RSLogix 5000 Structured Text	49
	RSLogix 5000 Function Block Diagram	49
	RSLogix 5000 Sequential Function Chart	49
	Conversion of STEP 5 Code to RSLogix 5000 Software	50
Software Conversions	Introduction	53
Instructions	Jump Operations	73
	Arithmetic Operations	77
	Block Call Operations	86
	Shift Operations	89
	Conversion Operations	91
Additional RSLogix 5000 Features	Introduction	97
	Task Monitor	97
	Asynchronous I/O Updating	98
	DINT Data Type	98
	String Data Type	98
	Coordinated System Time (CST)	99
	Coordinated System Time (CST) Object	99
	WallClockTime	99
	Timestamped Inputs	100
	Scheduled Outputs	100
Conversion of System Functions	Introduction	101
	Logix System Values	101
	Read/Write System Time	102
	Save/Load Scratchpad Area	104
Communication and Networking	S5 Serial Communication	105
	S5 Serial Communication Processors	
	(CP 521, CP 523, CP 524, and CP 525)	105
	Serial Communication Memory Submodule	105
	Serial Interface	105
	Logix Serial Communication	106
	S5 Distributed Control Systems	107
	S5 Network Communication	108
	CP 530 Network Communication Module	108
	CP 143 Network Communication Module	109
	NetLinx Networks	109
	Produced/Consumed Tags in RSLogix	111
	Fiber Optic Communication	
	IM 307/IM 317 Fiber Optic Interface Modules	112
	S5 CP 580 & CP 581 Communication Processor Modules	113
	IP 265 High Speed Sub Control	114

Drive and Loop Control	Encoder/Positioning Modules	115
	Logix Servo Drive Controllers.	115
	Replacing S5 Positioning Module with Logix.	116
	Controller with Motion	116
	S5 Analog Control Modules	117
	IP 243 Analog Module.	117
	IP 244 Temperature Controller Module.	117
	IP 252 Closed Loop Control Module.	117
	IP 260 Loop Controller Module	117
	Logix Analog Control	118
Logix Replacements for S5 Components	Introduction	119
	Controller Memory Use.	119
	This section covers the various Logix module types available for replacing S5 components and their advantages.	119
	CPU Modules.	119
	Digital Input Modules	121
	Digital Output Modules.	122
	Digital Relay Output Modules.	123
	Analog Input Modules.	124
	Siemens S5 Analog Input Range Cards	125
	Analog Output Modules	126
Other Communication Modules	Introduction	127
	In-rack Modules	127
	Profibus Network	127
	Siemens Industrial Ethernet/Modbus TCP/IP	129
	3964R Protocol	131
	Standalone Modules	133
	HMS	133
	ProSoft	134
	In-rack Interface Cards for S5 rack	136
	Efsys/A2i	136
	Rockwell Automation Support	140
	Installation Assistance	140
	New Product Satisfaction Return	140

Purpose

This user manual provides guidance for users and engineers who have used control systems based on one of these two platforms:

- Siemens S5 Controller
- Rockwell Automation Logix Programmable Automation Controller (PAC)

And in addition:

- have a desire or must take advantage of the PAC features, or are in the early stages of migrating a S5 to Logix.
- have STEP 5 program code that they wish to convert to effective and efficient RSLogix 5000 code.

Use this manual to help you adopt good practices and to avoid common mistakes when converting the project to Logix.

Conversion versus Translation

The theme of conversion versus translation is one that is repeated in this application conversion guide. Simple translation is focusing only on the line of code and finding an equivalent in the Logix languages. To convert an application optimally, you have to do more than just translate. For instance, you may benefit from choosing another programming language, utilizing other programming techniques, and designing another scheduling scheme to solve the same task. So, conversion is performed in a context of a higher level design and knowledge of the strengths of the Logix system.

If you have application code to convert, you must understand your STEP 5 program before you start conversion – by having been involved yourself in its development, or by reading documentation of the program and of the process that it controls. If the program or the process is unfamiliar or poorly documented, proper conversion will be difficult – it will be mere translation and is less likely to succeed. For example, in Logix, there is a global name space, whereas in the Siemens environment there are data blocks that can be loaded/unloaded by application code. Appreciation of this helps you design a strategy for conversion.

In cases where process and program documentation is poor, it may be more cost effective to draw up a new specification and begin your Logix program with minimal time spent on translating the old program.

Terminology

STEP 5 is the programming software environment for Siemens SIMATIC S5 controllers. RSLogix 5000 software is used with Rockwell Automation Logix programmable automation controllers. We refer to Logix as a programmable automation controller because it does so much more than a traditional general-purpose PLC. It provides an excellent control platform for multi-discipline control, a common namespace, Coordinated System Time for truly scalable multi-CPU architectures, user-defined data types, and full NetLinx connectivity.

The term “Logix” refers to any of the ControlLogix, CompactLogix, GuardLogix, FlexLogix, DriveLogix, or SoftLogix controllers, or the RSLogix 5000 programming environment where it is clear from the context to which it is being referred.

Additional Resources

Every section of this application conversion guide references other Rockwell Automation user manuals, selection guides, and documents in which more information can be found.

Publication Number	Publication Title
1756-SG001	ControlLogix Controllers Selection Guide
1769-SG001	1769 CompactLogix Controllers Selection Guide
1768-UM001	1768 CompactLogix Controllers User Manual
1769-SG002	Compact I/O Selection Guide
1756-RM094	Logix5000 Controllers Design Considerations Programming Manual
1756-PM001	Logix5000 Controllers Common Procedures Programming Manual
1756-RM003	Logix5000 Controllers General Instructions Reference Manual
1734-SG001	POINT I/O Selection Guide
1738-SG001	ArmorPoint I/O Selection Guide
1792-SG001	ArmorBlock MaXum I/O and ArmorBlock I/O Selection Guide
1794-SG002	FLEX I/O and FLEX Ex Selection Guide
NETS-SG001	NetLinx Selection Guide
VIEW-SG001	Visualization Platforms Selection Guide
IA-RM001	Integrated Architecture: Foundations of Modular Programming
6873-SG004	Encompass Program Product Directory
1756-PM010	Logix5000 Controllers Add-On Instructions Programming Manual
1756-RM087	Logix5000 Controllers Execution Time and Memory Use Reference Manual
IASIMP-RM001	IA Recommended Literature Reference Manual

PLC Logic Conversion Services Provided by Rockwell Automation

Rockwell Automation provides additional services for PLC logic conversion.

- Service Features
- One-stop PLC Program Conversion Services
- Service Benefits
- Services Offered
- Basic Conversion Package
- Conversion Plus Initial Clean-up Package
- Additional Program Conversions Available

Service Features

Program Conversion Services will convert your legacy Allen-Bradley brand PLC or third-party programmable controller program to run on a Logix programmable automation control system, or the SLC 500/MicroLogix or PLC-5 programmable controllers.

Legacy products are often expensive to support and are difficult to repair, which can increase downtime and decrease production. For this reason, Rockwell Automation Customer Support now offers Program Conversion Services. These services are designed to reduce the cost and the time it takes to migrate from a legacy PLC to one of our current PAC or PLC-control platform families.

One-stop PLC Program Conversion Services

Migration to a current Allen-Bradley control platform from a legacy product will improve your manufacturing process, system reliability and flexibility, give you more access to application processing power, and reduce equipment repair costs and spares inventory. With Program Conversion Services from Rockwell Automation Customer Support, your programmable controller program will be expertly and quickly converted to the new controller family. Rockwell Automation customer support engineers can help in the migration of legacy Allen-Bradley equipment or convert your PLC systems to Rockwell Automation products while minimizing downtime and maximizing operational success.

Service Benefits

Specialists for each of the product platforms will be involved during the program conversion process. There are no hard to find anomalies in the logic caused by typing errors. In most cases, the entire data table is reproduced and no data is lost, and the original documentation is preserved with no re-typing of comments and symbols. Original Allen-Bradley brand programs can be in 6200, APS, or AI series format. New programs will be in the RSLogix format.

Services Offered

Two program-conversion packages are available and project-specific custom packages are done on a case-by-case basis.

Basic Conversion Package

- The original programmable controller program will be converted to the ControlLogix, CompactLogix, PLC-5, or SLC 500/MicroLogix format.
- The package provides an error listing generated during the conversion that includes instructions that are not directly convertible and any addresses that may not have been converted, which could include pointers and indirect addressing.
- The program and error listing would be returned to the customer for manual debugging and correction.

Conversion Plus Initial Clean-up Package

- The original programmable controller program will be converted to the ControlLogix, PLC-5, or SLC 500/MicroLogix format.
- We will correct and convert any instruction and/or addressing errors to the new processor family.
- The completed program will then be returned to the customer for final startup and debugging.

Additional Options

Additional options include the following:

- Application-level telephone support during the start-up and debugging phase of the project.
- Consultation on system re-engineering, operator interface, architecture and communication strategies, to take full advantage of the new platform's control capabilities that are not part of a code translation effort, training, and onsite startup is available as an added value from your local Global Sales and Solutions (GSS) office.
- Complete turn-key migration or upgrades are available from your local GSS/Engineered Systems Office.

Additional Program Conversions Available

- PLC-2 format to ControlLogix, CompactLogix, PLC-5, SLC500/MicroLogix format
- PLC-3 format to ControlLogix, CompactLogix, or PLC-5 format
- PLC-5/250 format to ControlLogix or CompactLogix format
- Modicon – Quantum, 984, 584, 380, 381, 480, 485, 780, 785 to ControlLogix or CompactLogix format
- Siemens – S-5, S-7 to ControlLogix or CompactLogix format
- TI - 520, 520C, 525, 530, 530C, 535, 560, 560/565, 565, 560/560T, 560T, 545, 555, 575 to ControlLogix or CompactLogix format
- GE Series 6 to ControlLogix or CompactLogix format

Program conversions of other third-party programmable controllers to Allen-Bradley controller programs are also available. Contact technical support for details.

To schedule a conversion project, or learn more about the Program Conversion Services, contact your local Rockwell Automation sales office or authorized distributor: email us at raprogramconversion@ra.rockwell.com, or visit <http://www.rockwellautomation.com/support> and view KnowledgeBase Document G19154.

IMPORTANT

Use consultation services for re-engineering, typically to expand the system functionality and not to change out hardware due to obsolete or related reasons. SLC to Logix format and PLC-5 to Logix format conversions and PCE comment generation are built into RSLogix 5000 software.

Notes:

Controllers and I/O Modules

Introduction

The objective of this chapter is to provide a user with information for replacing Siemens S5 PLC hardware with Rockwell Automation's family of Logix Controllers and hardware.

In 2002, Siemens announced the gradual phasing out of the S5 PLC starting on October 1, 2003. Apart from the decline in the demand for the 20- year old technology, Siemens stated that it was becoming increasingly difficult to find suitably priced and technically matching components for an aging system.

This meant that new S5 hardware was available for approximately one year until October 2004. After that, the products will remain available on a spare parts basis for nine years until October 1, 2013. The dates for the Simatic S5-115, the medium performance range systems, are October 1, 2004, September 30, 2005 and September 30, 2014. The dates for the large-scale Simatic S5-135/155 systems are October 1, 2005, September 30, 2006 and September 30, 2015.

The chapter describes how to select controllers, local I/O, remote I/O, networks, and HMI. It also includes a section on distributed controller architecture, and provides hardware conversion examples for the most popular S5 modules.

S5 Controllers

When converting a control system from Siemens S5 to a ControlLogix system, it is essential to select a CPU that will have sufficient memory capacity to store the converted code.

S5 Controller	Memory Capacity
CPU 90U	4 KB
CPU 95U	16 KB
CPU 100	2 KB
CPU 102	4 KB
CPU 103	20 KB
CPU 921	22 KB
CPU 922	22 KB
CPU 928	110 KB
CPU 928	110 KB
CPU 928 3UA21	110 KB
CPU 928B	110 KB
CPU 928 3UB21	110 KB
CPU 941	18 KB
CPU 942	42 KB
CPU 943	48 KB
CPU 944	46 KB
CPU 945 7UA1	256 KB
CPU 945 7UA2	384 KB
CPU 946	896 KB
CPU 947	896 KB
CPU 948-1	640 KB
CPU 948-2	1664 KB
CPU 948B 3UA13	64 KB
CPU 948B 3UA23	64 KB

Logix Controllers

Logix Controller	Memory Capacity
1756-L60M03SE	750 KB
1756-L61	2 MB
1756-L61S (Safety)	2 MB 1 MB (Safety)
1756-L62	4 MB
1756-L62S (Safety)	4 MB 1 MB (Safety)
1756-L63	8 MB
1756-L63S	8 MB 1 MB (Safety)
1756-L64	16 MB
1756-L65	32 MB
1768-L43	2 MB
1768-L45	3 MB
1769-L31	512 KB
1769-L32E	750 KB
1769-L35E	1.5 MB

Care must be taken when estimating the memory requirements for a Logix project that has been converted from S5.

The S5 is an older generation PLC than the Logix PAC (Programmable Automation Controller) and comes from a period when memory storage was bulkier and more expensive than it is today. Thus the S5 controller stored only the PLC code and not the associated symbols and comments that were retained on the programming device. When uploading a PLC program from an S5 controller to a programming device that did not contain the original program, the resulting program would not contain any symbolic information.

Because memory is now considerably less expensive and physically smaller, Logix controllers can use tag naming and I/O aliasing for memory locations and store the information in the CPU. However, this results in the Logix controller using typically more memory to store a similar program than an S5 PLC.

Logix Controller Memory Use

The following table provides calculations for estimating the amount of memory needed to store a program in a Logix controller.

Additional recommendations for selecting ControlLogix system components can be found in the ControlLogix Selection Guide, publication [1756-SG001](#) and Logix5000 Controllers Design Considerations, publication [1756-RM094](#).

Logix Controller	Bytes
Controller Tasks (minimum 1 required)	____ x 4000 = ____ bytes
Digital I/O Points	____ x 400 = ____ bytes
Analog I/O Points	____ x 2600 = ____ bytes
DeviceNet Modules	____ x 7400 = ____ bytes
Other Communication Modules	____ x 2000 = ____ bytes
Motion Axis	____ x 8000 = ____ bytes
FactoryTalk Alarm Instruction	____ x 2200 = ____ bytes
FactoryTalk Subscriber	____ x 100,000 = ____ bytes

Example

To estimate the memory requirements of a program consisting of the following:

- 25 Controller Tasks
- 200 Digital I/O
- 50 Analog I/O
- 5 ControlNet Modules
- 1 Ethernet Module

Description	Memory Usage
Controller Tasks	25 x 4000 = 100 KB
Digital I/O Points	200 x 400 = 80 KB
Analog I/O Points	50 x 2600 = 130 KB
Communication Modules	(5+1) x 2000 = 12 KB
Total	322 KB
Spare Capacity 25%	80.5 KB
Required Memory	402.5 KB

S5 Controller Memory Use

Siemens recommends estimating 2 bytes of memory per each statement of code.

If it is estimated that the program in the above example requires 25 tasks containing an average of 100 lines of code each and each line of code has 10 statements, then the memory requirements would be:

$$25 \times 100 \times 10 \times 2 = 50,000 = 50 \text{ KB}$$

A spare capacity requirement of 25%:

$$50 \text{ KB} \times 1.25 = 62.5 \text{ KB}$$

Digital Input Modules

The following table lists the available Siemens S5 Digital DC Input Modules.

Siemens S5 Digital DC Input Modules

Cat. No.	Inputs	Voltage
6ES5 420-4UA13/14	32	24V DC
6ES5 430-4UA13/14	32	24V DC
6ES5 420-8MA11	4	24V DC
6ES5 421-8MA12	8	24V DC
6ES5 422-8MA11	16	24V DC
6ES5 430-8MB11	4	24...60V DC
6ES5 431-4UA12	16	24...60V DC
6ES5 431-8MA11	8	24V DC
6ES5 432-4UA12	32	24V DC/alarm
6ES5 433-8MA11	8	5...24V DC
6ES5 434-4UA12	32	5 ...15V DC

These are the available ControlLogix Digital DC Input Modules.

ControlLogix Digital DC Input Modules

Cat. No.	Inputs	Voltage
1756-IB16	16	12/24V DC sink
1756-IB16D	16	12/24V DC sink
1756-IB16I	16	12/24V DC sink/source
1756-IB16ISOE	16	24/48V DC sink/source
1756-IB32	32	12/24V DC sink
1756-IC16	16	30...60V DC sink
1756-IG16	16	5V DC TTL source
1756-IH16I	16	90...146V DC sink/source
1756-IH16ISOE	16	90...146V DC sink/source
1756-IV16	16	12/24V DC source
1756-IV32	32	12/24V DC source

Replacing 6ES5 420-4UA13/14 and 6ES5 430-4UA13/14 DI Modules

The simplest replacement for these modules would be to replace them with ControlLogix 1756-IB32 DI module.

If there are more than 16 spare inputs on the DI modules, it may be better to replace them with the ControlLogix 1756-IB16 module.

It is often the case that the density of wiring on a 32-channel module makes wiring and troubleshooting difficult. If this is the case, consider replacing these modules with two 16-channel ControlLogix 1756-IB16 modules.

If additional diagnostic features are required, then the 1756-IB32D module can be used. This module can detect an open wire condition for each I/O point in the field and set an associated diagnostic bit that can be monitored in the project.

The 1756-IB32I module would be a possible replacement if some/all inputs must be individually isolated.

Replacing 6ES5 431-4UA12 DI Modules

This module requires voltages from 24...60V to register an input as ON. The actual values of input voltage presented by the input signals should be reviewed and if they are all found to be within the range of 30...60V then this module can be replaced by the 1756-IC16 module, while any 24V inputs could be connected to a 1756-IB16 module.

Replacing 6ES5 432-4UA12 DI Modules

The 432 digital input module accepts 32 process signals and emits a process alarm if the logic state at one of its inputs changes.

This module could be replaced with a ControlLogix 1756-IB32 DI module where the process alarm could be generated by configuring a Module Input Data State Change as a Trigger for an Event Task. See “Event Task” on [page 34](#) for more information.

Replacing 6ES5 434-4UA12 DI Modules

This module requires voltages from 5V...15V to register an input as ON. The actual values of input voltage presented by the input signals should be reviewed to determine which of the following ControlLogix modules would be the best replacement.

- 1756-IG16 (5V TTL source)
- 1756-IV16/1756-IV32 (12/24 DC source)

The following table lists the available Siemens S5 Digital AC Input Modules.

Siemens S5 Digital AC Input Modules

Cat. No.	Inputs	Voltage
6ES5 435-4UA12	16	24...60 V AC
6ES5 436-4UA12	16	115...230 V AC
6ES5 436-4UB12	8	115...230 V AC

These are the available ControlLogix Digital AC Input Modules.

ControlLogix Digital AC Input Modules

Cat. No.	Inputs	Voltage
1756-IN16	16	10...30V AC
1756-IA8D	8	79...132V AC
1756-IA16	16	79...132V AC
1756-IA16I	16	79...132V AC
1756-IA32	32	79...132V AC
1756-IM16I	16	79...132V AC

Replacing 6ES5 435-4UA12 DI Modules

These modules have an operating voltage range of 24...60V AC and the nearest equivalent ControlLogix module is the 1756-IN16 that has an operating range of 10...30V AC.

The actual values of input voltage presented by the input signals should be reviewed and if they are all found to be within the range 10...30V AC then the 6ES5 435-4UA12 DI module can be replaced by the ControlLogix 1756-IN16 module. There is no ControlLogix DI module that operates with an input voltage in the range 30...60V AC.

Replacing 6ES5 436-4UA12 DI Modules

These modules could be replaced by a ControlLogix module 1756-IA16, or if individually isolated inputs are required, then the ControlLogix module 1756-IA16I can be used.

Two 16-channel 6ES5 436-4UA12 modules could be replaced by one 32-input channel 1756-IA32 module.

Replacing 6ES5 436-4UB12 DI Modules

These modules could be replaced by the ControlLogix module 1756-IA8D with individual diagnostic information for each input channel.

Two 8-channel 6ES5 436-4UB12 modules could be replaced by one 16-input channel 1756-IA16 module.

Digital Output Modules

The following table lists the available Siemens S5 Digital DC Output Modules.

Siemens S5 Digital DC Output Modules

Cat. No.	Outputs	Voltage
6ES5 440-8MA12	4	24V DC
6ES5 440-8MA22	4	24V DC
6ES5 441-8MA11	8	24V DC
6ES5 441-4UA13/14	32	24V DC
6ES5 450-8MB11	4	24...60V DC
6ES5 451-8MA11	8	24V DC
6ES5 451-4UA13/14	32	24V DC
6ES5 453-8MA11	8	5...24V DC (sink)
6ES5 453-4UA12	16	24V DC
6ES5 454-4UA13/14	16	24V DC
6ES5 457-4UA12	16	24...60V DC
6ES5 458-4UA12	16	24V Relay

These are the available ControlLogix Digital DC Output Modules.

ControlLogix Digital DC Output Modules

Cat. No.	Outputs	Voltage
1756-OB8	8	12/24V DC source
1756-OB8EI	8	12/24V DC source
1756-OB8I	8	12/24V DC source
1756-OB16D	16	12/24V DC source
1756-OB16E	16	12/24V DC source
1756-OB16I	16	12/24V DC sink/source
1756-OB16IS	16	12/24V DC sink/source
1756-OB32	32	12/24V DC source
1756-OC8	8	30...60V DC source
1756-OG16	16	5V DC TTL
1756-OH8I	8	120V DC sink/source
1756-OV16E	16	12/24V DC sink
1756-OB32E	32	12/24V DC sink

Siemens S5 Digital DC Output Modules with an operating voltage of 24V DC can be replaced with any of the ControlLogix Digital Output Modules with the same operating voltage. The ControlLogix Digital Output Modules can be electronically fused, individually isolated, and can provide additional diagnostic information to the project code as summarized in the table above.

The Siemens S5 6ES5 457-4UA12 Digital DC Output Module has an operating range of 24...60V DC. The actual values of input voltage presented by the input signals should be reviewed and if within the range 30...60V DC, then this module can be replaced by the ControlLogix 1756-OC8 module.

Siemens S5 Digital AC Output Modules

Cat. No.	Outputs	Voltage
6ES5 451-8MD11	4	115...230V AC
6ES5 455-4UA12	16	24...60V AC
6ES5 456-4UA12	16	115...230V AC
6ES5 456-4UB12	8	115...230V AC
6ES5 458-4UC11	16	250V AC (Relay)

Siemens S5 Digital Relay Modules

Cat. No.	Outputs	Voltage
6ES5 452-8MR11	4	Relay 30V DC / 230V AC
6ES5 452-8MR11	8	Relay 30V DC / 230V AC
6ES5 458-8MR12	16	Relay 250V AC

ControlLogix Digital AC Modules

Cat. No.	Outputs	Voltage
1756-ON8	8	10...30 V AC
1756-OW16I	16	10...265V AC 5...150V DC Relay
1756-OX8I	8	10...265V AC 5...150V DC Relay

Analog Input/Output Modules

When converting from a Siemens S5 Controller to a Rockwell Automation Logix Controller, it is essential to choose analog input/output modules that will be compatible with existing analog input/output devices.

Some of the S5-135U and S5-155U analog input modules are used in conjunction with range cards, with range cards being used depending on the type of signal being produced by the analog input device as shown in the following tables.

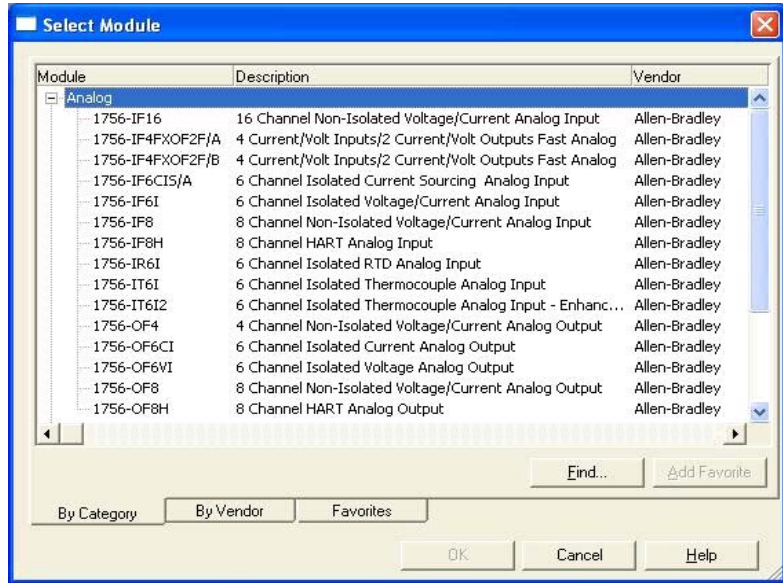
Siemens S5 Analog Input Modules

Cat. No.	Channels	Range / Sensor
460-4UA13	8 – single	See range cards
465-4UA12	16 – voltage/current 8 – Pt100 resistance	See range cards
465-4UA13	16 – voltage/current 8 – Pt100 resistance	See range cards
463-4UA12	4 – single	0...1V, 10V, 20 mA
463-4UB12	4 – single	4...20 mA
466-3LA11	16 – voltage/current 8 – Pt100 resistance	0...1.25V, 2.5V, 5V, 10V
466-4LA11	16 – voltage/current 8 – Pt100 resistance	0...20 mA 4...20 mA

Siemens S5 Analog Input Range Cards

Cat. No.	Range Cards	Range / Sensor
460-4UA13	498-1AA11	±12.5, 50, 500 mV, Pt100
465-4UA12	498-1AA11	± 50/500 mV, Pt100
	498-1AA21	± 1V
465-4UA13	498-1AA31	± 10V
	498-1AA41	± 20 mA
	498-1AA51	4...20 mA
	498-1AA61	± 5V
	498-1AA71	4...20 mA

All I/O configuration is done from the project tree of the RSLogix 5000 software. When selecting an analog module, a list of available analog modules is displayed as shown in the following dialog box.



The Logix 1756 Analog Input Modules are listed in the following table.

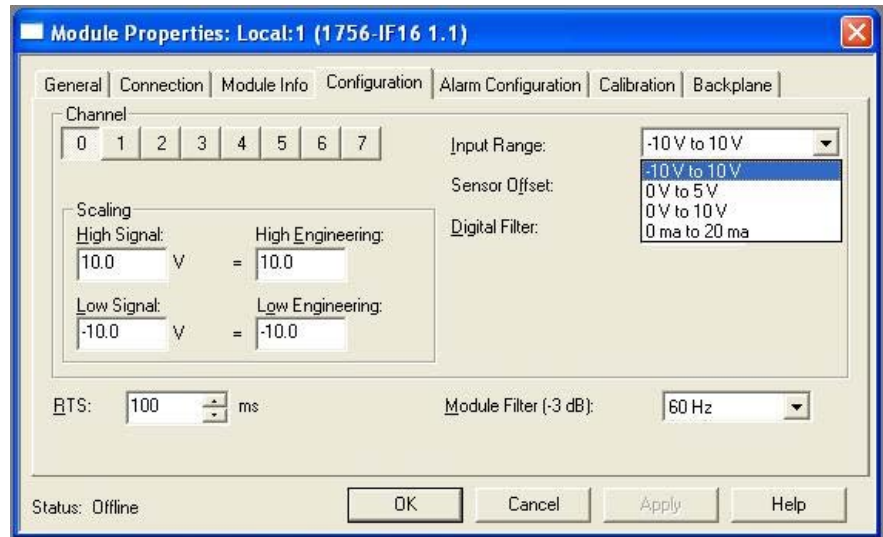
Logix Analog Input Modules

Cat. No.	Description
1756-IF8	8 channel non-isolated voltage/current
1756-IF6CIS	6 channel isolated current sourcing
1756-IF6I	6 channel isolated voltage/current
1756-IF16	16 channel non-isolated voltage/current
1756-IF4FXOF2F	4 current/volt inputs 2current/volt outputs
1756-IR6I	6 channel isolated RTD
1756-IT6I	6 channel isolated thermocouple
1756-IT6I2	6 channel isolated thermocouple enhanced

Logix analog input modules provide greater flexibility than the Siemens S5 counterparts because they can be configured in software, enabling them to receive input signals from the analog input devices. The dialog box on the next page shows that by accessing the properties of an analog module, the signal range, along with the high/low voltage current limits for the analog signal, can be modified for each channel.

The Logix analog input modules also allow scaling of the raw signal to the required engineering units and configure alarm limits via the Alarm Configuration tab.

These modules are supplied with standard calibration but can be recalibrated on a module or channel-by-channel basis for greater accuracy in applications.



Siemens S5 Analog Output Modules

Cat. No.	Channels	Range
470-4UA12	8	± 10V 0...20mA
470-4UA13		
470-4UB12	8	± 10V
470-4UB13		
470-4UC12	8	1...5V 4...20mA
470-4UC13		

The Logix 1756 Analog Output Modules are listed in the following table.

Logix Analog Output Modules

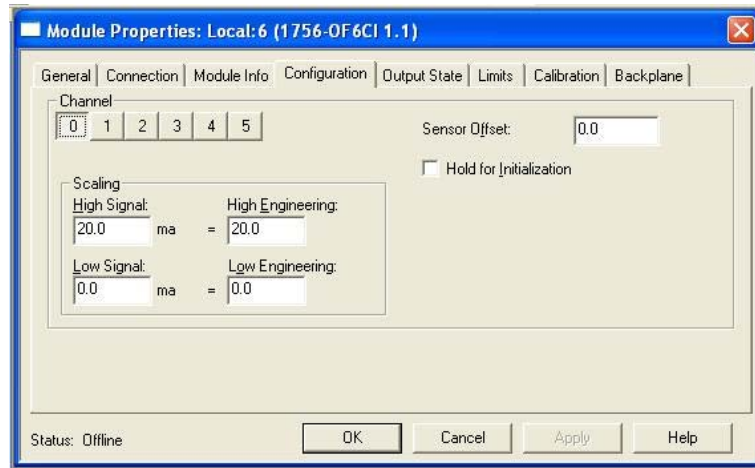
Cat. No.	Description
1756-OF4	4 channel non-isolated voltage/current
1756-OF6CI	6 channel isolated current
1756-OF6VI	6 channel isolated voltage
1756-OF8	8 channel non-isolated voltage/current
1756-OF8H	8 channel voltage/current (HART)

Logix analog output modules provide added flexibility because they can also be configured in software, allowing selection of the required output range for the process device.

The following dialog box shows how, by accessing the properties of an analog output module, the output signal range can be modified and the high/low voltage/current limits set for each analog output channel.

The Logix analog output modules also provide scaling to convert the engineering units to the required raw values and configuration of the alarm limits via the Alarm Configuration Tab.

These modules are supplied with standard calibration but can be recalibrated on a module or channel-by-channel basis for greater accuracy in applications.



The 1756-OF8H module includes the Highway Addressable Remote Transducer (HART) protocol that provides additional field data and device status information from HART devices connected to this module.

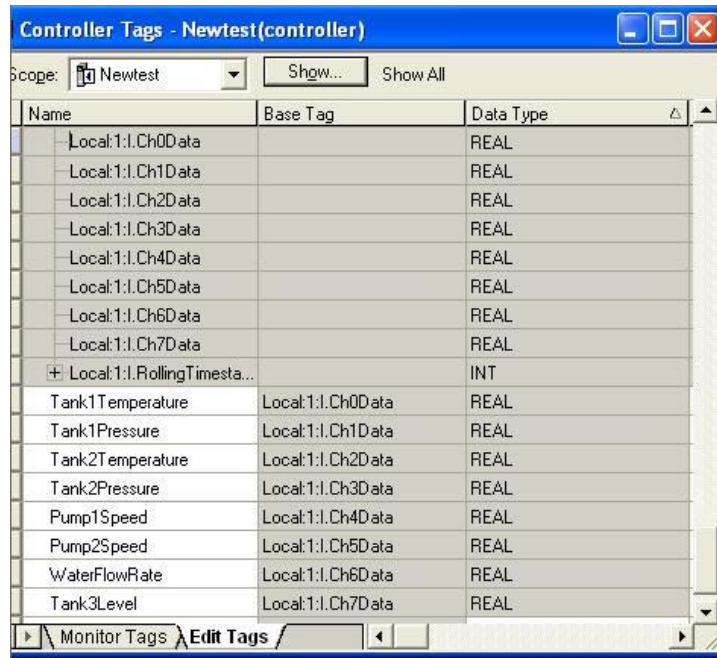
Addressing of the Siemens S5 Analog Modules is done by switches on the side of the modules.

Addressing of Logix Analog modules is done automatically. When an analog module is added, the tags that address the module are automatically created, as shown in the following dialog box.



The programmer can then create new tags with recognizable tag names that describe the associated process signals to alias the analog module tags. These new alias tags will contain the same data as the analog module tags they alias.

The following dialog box shows eight analog input channels from an analog input module for which alias tags were created.



Logix Controller with S5 I/O For customers to be able to migrate from S5 to current RA Technology within their time and budget constraints, while mitigating their risk of production downtime, a “phase-in” solution is possible by using variable ‘Siemens network’ linking modules (for example, Profibus DP network).

In the following section we will discuss some scenarios on how to connect to Legacy S5 hardware. The communication to S5 I/O and peer-to-peer communication is covered.

In these scenarios, the profibus scanners (in rack modules) are mostly used for communication to Legacy S5 I/O while the stand-alone modules are more for peer-to-peer communication between S5/S7 PLCs and Logix controllers.

CompactLogix controllers

A CompactLogix system can be used when converting a S5 system to a Logix-based platform, especially for small and standalone applications. For these applications, consider upgrading the S5 I/O immediately to Logix I/O together with the controller.

If the S5 I/O must be maintained temporarily, communication modules for the CompactLogix range can connect to all sorts of networks (such as Profibus DP). Suppliers of ControlLogix communication modules for third party networks usually provide communication modules for the CompactLogix platform, too. For more details, please examine the encompass partner websites. See Appendix A for available communication modules for the ControlLogix platform with similar functionality as the CompactLogix modules.

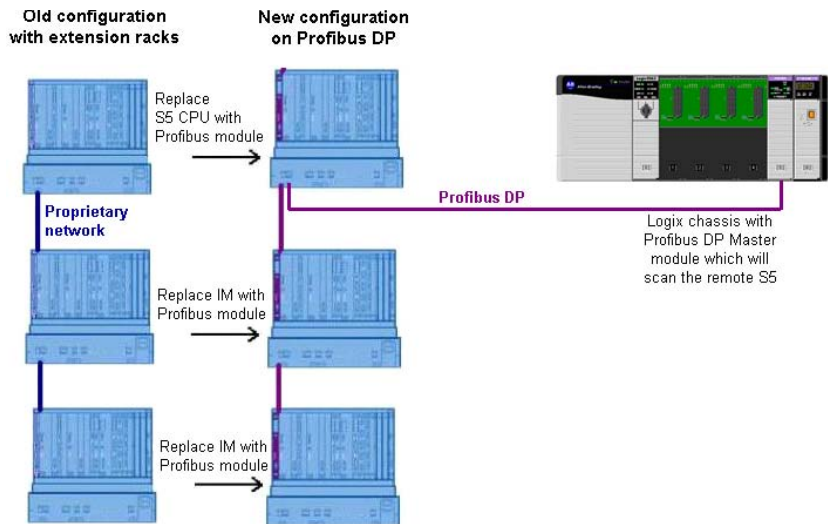
ControlLogix controllers

To limit migration risks, a phased migration can be done where the S5 controller is replaced by a ControlLogix controller, but where the S5 I/O (local S5 rack, S5 extension racks, S5 remote racks) is maintained temporarily. This approach allows for fast rollback scenarios and for shorter development cycles; that is, no installation, wiring, and testing of field I/O is required. The S5 I/O can be phased out gradually once the new controller is running.

Integrating the S5 hardware in the Logix architecture is possible through these three communication methods:

- I/O in local rack/extension rack

To be able to communicate with these I/O cards, they must be made available to a network. Therefore, new Profibus DP communication cards can be slotted in the S5 chassis to make each chassis available to the Profibus DP network. In the local (main) rack, simply replace the controller by a Profibus DP slave module. In the extension racks also replace the proprietary interface card by a Profibus DP slave module.



- Remote I/O

Most remote I/O racks are connected to the Profibus DP network, so no changes are needed. When the remote I/O connects to another network (for example, Legacy Siemens network), the interface module must be replaced by a Profibus DP slave module.

- Peer-to-peer communication

Because not all controllers can be migrated simultaneously, you may need to exchange data between the new ControlLogix controllers and other S5 controllers (for example, interlocking or handshaking). To accommodate this, several options are available.

- A Profibus DP communication module placed in the ControlLogix chassis to communicate with master and slaves on the Profibus DP network.
- A standalone module (gateway) that provides a transparent pass for the data between two networks (for example, Ethernet/IP to Profibus DP). They do not change the data sequence or the data itself.
- A module placed in the Logix chassis that connects and transfers monitoring and control data between islands of automation across diverse platforms, networks and protocols. Additional message-specific controller programming with this solution is unnecessary. The module can be configured for the right data mapping.

[See Appendix A](#) for more information on the possible modules to use.

S5 Controller with Logix I/O

Rockwell Automation 1794-FLEX I/O series is a small, modular I/O system for distributed applications that perform all functions of rack-based I/O. The FLEX I/O system consists of power supply, terminal base and I/O modules. By using this I/O system with the 1794-APB Flex I/O Profibus adapter, it is possible to make the I/O available on a Profibus network.

The S5 controller can be configured as a Profibus Master and communicate to the I/O via Profibus. The configuration of the S5 controller will be dependant on the model of S5 controller used.

Prosoft also provides modules to link Logix modules to Profibus.

Notes:

Software Conversions - Program Structure

Introduction

The Siemens S5 controller is programmed with the Step 5 software package, while Rockwell Automation's Logix controllers are programmed with the RSLogix 5000 software package.

This chapter explains the differences between the structure of a Step 5 PLC program and an RSLogix 5000 project and provides guidance in converting an S5 program to a RSLogix 5000 project.

There are two components in an S5 PLC program; the logic contained in modules such as function blocks, and the memory variables that store data related to this logic. The Code Area consists of Program Blocks (PB), Function Blocks (FB), Extension Function Blocks (FX) and Organization Blocks (OB). The Data Storage area consists of Data Blocks (DB), Extended Data Blocks (DX), and Flag Words (FW). This chapter looks at how these two components are implemented in Step 5 and RSLogix 5000 software, and seeks to give an understanding of the optimal method of converting the structure of a Step 5 program to an RSLogix 5000 project.

Organization Blocks/Logix Tasks

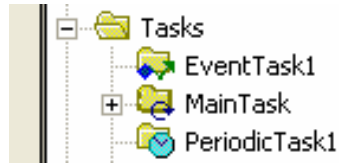
This comparison of Step 5 Organization Blocks and RSLogix 5000 tasks is designed to introduce the structure of an RSLogix 5000 project to the S5 user.

Step 5 uses Organization Blocks (OBs) as the top level in the hierarchy of program calls. These OBs are called by the controller's operating system and all user function blocks and program blocks are called from these OBs.

In RSLogix 5000 software, the equivalent project elements to these OBs are called Tasks. In Step 5 there are three types of OBs and the table below details the equivalent RSLogix 5000 tasks for each Step 5 OB type.

Step 5 OB	RSLogix 5000 Task	Description
Program Cycle OB	Continuous Task	The code is executed cyclically, recommencing executing the code from the beginning when the last instruction has been executed.
Cyclic Interrupt OB	Periodic Task	The code is executed at defined time intervals.
Hardware Interrupt OB	Event Task	The code is executed in response to a defined event.

The following window shows the layout of tasks in an RSLogix 5000 project and the folder icons used to indicate task types.



Continuous Task

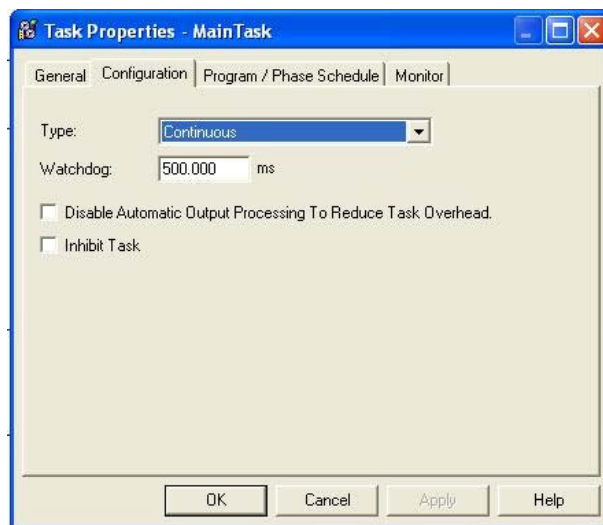
In Step 5 there is only one continuously executed Organization Block and it is always designated OB1. This Organization Block determines the sequence by which the program blocks, function blocks and data blocks of the PLC program are processed. The corresponding calls (conditional or unconditional) for the blocks required are written in Organization Block 1 (OB1).

In RSLogix 5000 software, the equivalent to the Step 5 OB1 is the continuous task that appears in an RSLogix 5000 project as follows:



Unlike Step 5, the existence of a continuous task is not compulsory in an RSLogix 5000 project.

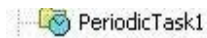
By right-clicking on the Main Task icon and choosing Properties, the configuration menu for the continuous task is displayed.



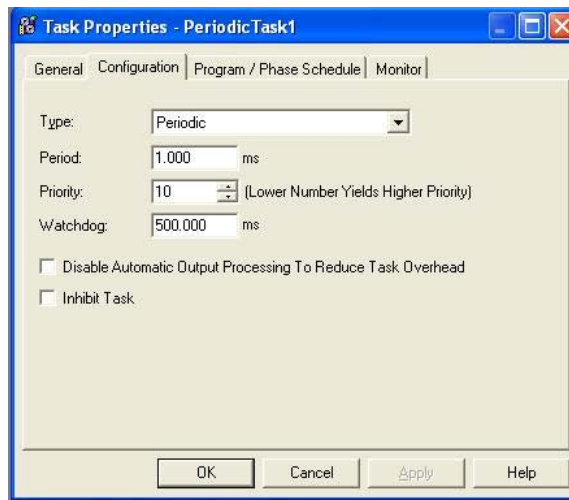
Periodic Task

Step 5 and RSLogix 5000 software allow tasks to be programmed that will execute at defined intervals. In the S5 PLC, the number of tasks and the permitted time intervals is dependent on the model of controller used. For example, in the S5-135U PLC Series, there are up to nine periodic tasks that can be executed. These programs reside in Organization Blocks OB10 to OB18 with each OB being executed, predefined-time intervals from 10 msec to 5 sec.

In RSLogix 5000 software, the equivalent to the S5 Periodic Task is indicated in an RSLogix 5000 project as follows:



By right-clicking on this icon and choosing Properties, the following configuration menu for the periodic task is displayed. From this menu, the executed task can be configured including task priority where tasks with lower numbers indicate a higher priority.



Event Task

Step 5 and RSLogix 5000 software allow tasks to be programmed that interrupt the normal cyclic sequence and process the logic contained in designated event task on detection of predefined event.

In Step 5, the predefined event is restricted to the change of state of an input while RSLogix 5000 permits event tasks to be initiated on the following:

- Event Instruction
- Module Input Data State Change
- Consumed Data
- Motion Control Events

In Step 5, the code to be executed on detection of an interrupt event is contained in predefined Organization Blocks. The number of event tasks that can be programmed depends on the model of CPU in use.

In RSLogix 5000 software, the equivalent to the Event Task is indicated in an RSLogix 5000 project as follows:



By right-clicking on this icon and choosing Properties, the following configuration menu for the event task is displayed. From this menu, the event that initiates the task can be determined and tasks priority can also be defined.

Task Properties

The following property settings can be configured for all tasks:

- Watchdog - timer that specifies how long a task can run before triggering a fault.
- Disable Automatic Output Processing To Reduce Task Overhead – this is an option that allows output processing to be turned off, reducing the elapsed time for that task. Normally at the end of a task the controller performs overhead operations (output processing) for the I/O modules in the system. While not the same as updating the modules, this output processing may affect the update of I/O in the system.
- Inhibit Task - this is an option that prevents a task from executing when the task trigger occurs. This option can be useful during testing/commissioning.

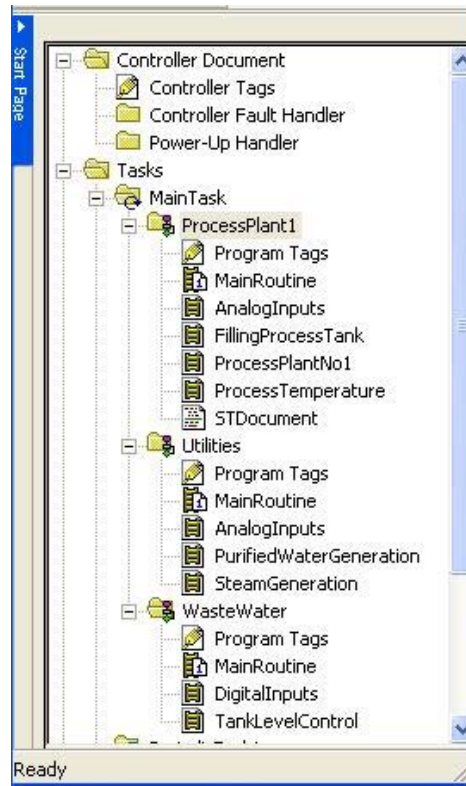
RSLogix 5000 Programs

The RSLogix 5000 tasks can be divided into a number of programs. This subdivision does not exist in Step 5, but can be used in RSLogix 5000 software to achieve a more modular software architectural design.

When converting Step 5 applications with many individual OBs and PBs, consider separating the application into one or more RSLogix 5000 programs.

The nature of the application will determine how the tasks are divided into programs. On smaller applications it may be suitable to include all the user code in one program. The following example shows how the continuous task is divided into the following three programs with each one containing its own routines.

- Process Plant1
- Utilities
- Wastewater





Blocks/Routine

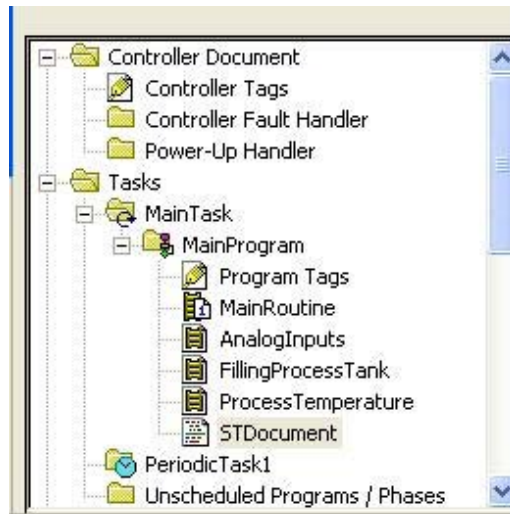
This section discusses anomalies involved in converting S5 Program Blocks (PBs) and Function Blocks (FBs) to RSLogix 5000 routines.

The next level down from the OB in the Step 5 program structure will usually contain Program Blocks (PBs) and possibly Function Blocks (FBs). Usually OB1 will call a number of PBs and/or FBs with each block executing a distinct function. The nature of the function depends on the particular application, but typically could include functions such as Scaling Analog Inputs, Filling Process Tank, or Controlling Process Temperature.

RSLogix 5000 software has a similar structure but uses different terminology. Instead of the term Block, RSLogix 5000 software uses the term Routine.

In Step 5, OB1 is the first code executed. In RSLogix 5000 software, the first routine executed can be selected by the programmer. It is indicated in the project tree by the icon  beside the Main Routine. When converting a PLC application from Step 5 to RSLogix 5000 software, function calls originally in S5 OB1, are commonly placed in the designated RSLogix 5000 Main Routine.

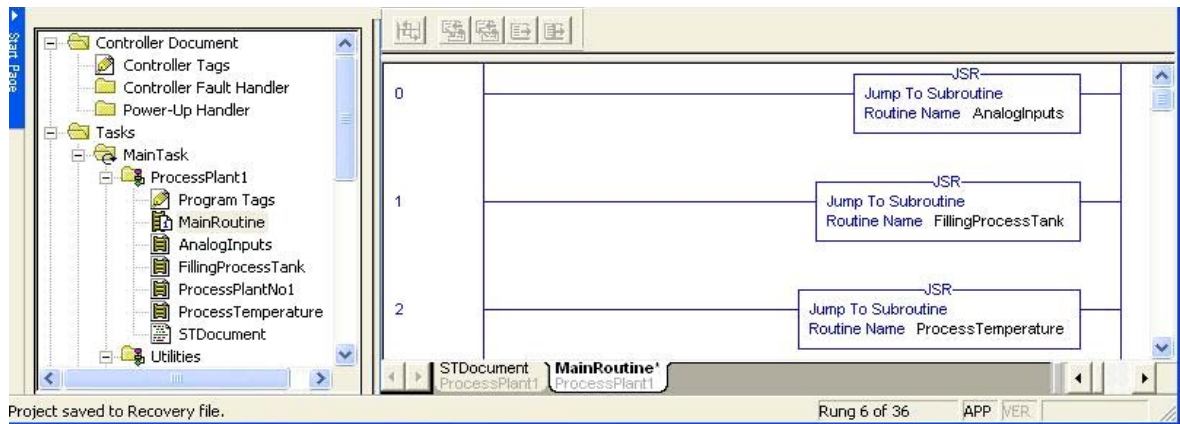
The following window shows how routines appear under the continuous task in the RSLogix 5000 project tree. The Main Routine designated as the start-up routine is indicated by the icon .



The following example shows how three PBs called in from OB1 could be converted into RSLogix 5000 software.

In S5 the PBs are given symbol names and called from OB1. In RSLogix 5000 software, the routines are called from the Main Routine.





More details on programming in RSLogix 5000 software can be found in the ControlLogix User Manual, publication [1756-UM001](#).

Function Blocks/Add-On Instructions

RSLogix 5000 software allows programming of Add-On Instructions that can replace and enhance the functionality of Step 5 Function Blocks.

The Add-On Instruction functionality allows the combination of commonly used logic into one function that can be called and executed multiple times as required in the program.

Add-On Instructions are programmed by Ladder Logic, Structured Text, or Function Block Diagrams.

The Add-On Instruction, like the Step 5 Function Block can be assigned input and output parameters.

Unlike Step 5, an Add-On Instruction can also have an InOut Parameter to define data used as input and output during instruction execution. An InOut Parameter is passed by reference, meaning the logic deals directly with the tag passed to the instruction. InOut parameters can be any data type because they are passed by reference. Any data structures or arrays that must be passed to an Add-On Instruction can be done with InOut Parameters.

The Add-On Instruction will automatically maintain a change history by recording a timestamp and the Windows user name at the time of the change. This is not available with Step 5 Function Blocks. The Add-On Instruction also automatically generates help files.

With the Add-On Instruction a pre-scan routine can be configured to run when the controller goes from Program mode to Run mode, or powers up in Run mode. Under these conditions, the pre-scan routine will run once, and can typically be used to initialize data. This is not available with Step 5 Function Blocks.

An Add-On Instruction can have an EnableInFalse routine, which will be called (if present) when the rung condition at the Add-On Instruction call is false. In this case, the input and output parameters will pass values.

Watchdog Timer

A watchdog timer can be configured for each RSLogix 5000 task. This timer specifies the maximum time a task can run before triggering a fault.

In S5 the user configures the cycle time in 54 DX0 or OB26. An error is triggered if a scan of the program is not executed within this time. There is no facility to set watchdog timers for individual blocks in S5. The S5 watchdog timer can be retriggered by calling OB 222.

I/O Addressing

In Step 5 and RSLogix 5000 software, the software address of a physical input from the plant is determined by the position of the input card and the channel to which the input is connected.

Similarly, the software address of a physical output to the plant is determined by the position of the output card and the channel to which the output is connected.

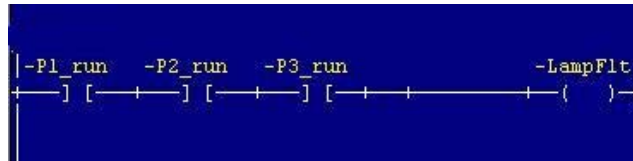
Example

In Step 5, if an input is connected into the 3rd channel on the first input card, it is assigned the address I0.2. (The first channel on this card is assigned address I0.0). Using the Step 5 symbols table, the programmer can assign a name to each input/output as shown below.

```
Symbols file : D:\TEST1020.INI
```

Operand	Symbol	Comment
I 1.0	P1_run	Pump 1 Running Status
I 1.2	P2_run	Pump 2 Running Status
I 1.3	P3_run	Pump 3 Running Status
Q 4.0	LampFlt	Fault Lamp

In the ladder code these symbols can be displayed instead of the assigned addresses as shown below.



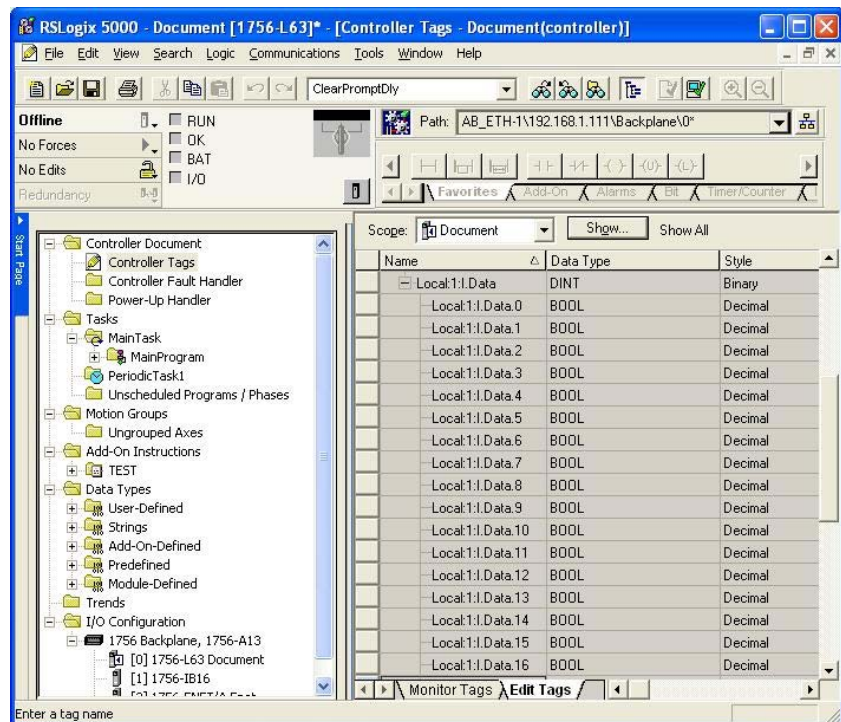
A limitation of Step 5 is that the length of a symbol name is by default restricted to 8 characters, extendable to maximum 20 characters. The comment is limited to 20 characters and is extendable to 48 characters.

In RSLogix 5000 software, a feature called ‘built-in device profiles’ means that when a digital input/output card is configured, the individual channels are automatically assigned addresses based on the digital module slot number and type of card.

This address can be seen from the Controller Tag screen that is accessed by clicking the Controller Tag icon on the project tree as shown below.

In the window below, a 16-channel digital input module is configured in slot 1. The channels are assigned addresses

Local:1:I.Data.0 to Local:1:I.Data.15



Logix Aliasing

The difference between Step 5 and RSLogix 5000 software is the RSLogix 5000 software uses a feature called aliasing to assign more intuitive names to I/O, which makes for more user-friendly programming.

An alias can be created for each I/O point and this alias will contain the same value as the associated I/O point.

When writing code, the programmer can use the alias name instead of the address location. The alias name can be up to 40 characters in length.

Use aliases in the following situations:

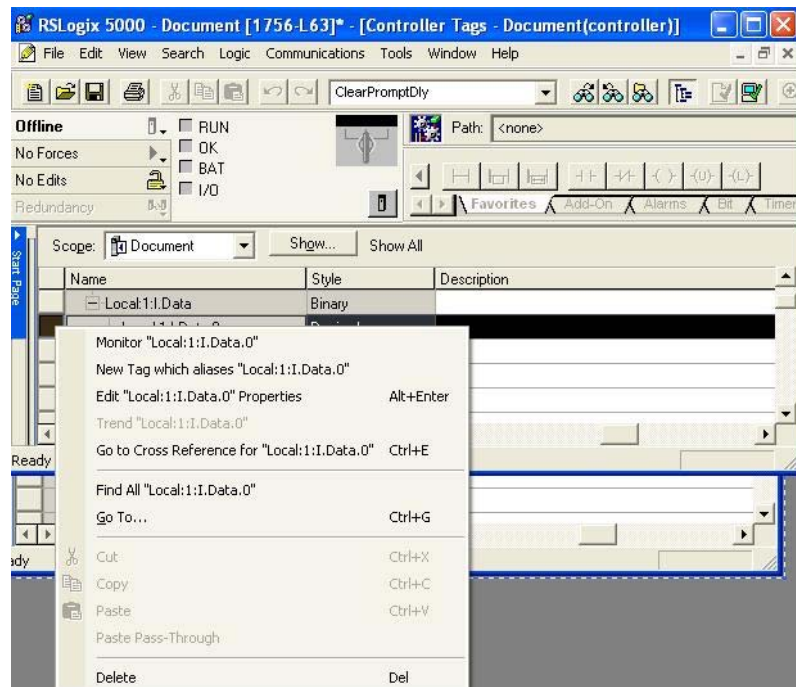
- Program logic in advance of wiring diagrams.
- Assign a descriptive name to an I/O device.
- Provide a simpler name for a complex tag.
- Use a descriptive name for an element of an array.

Example

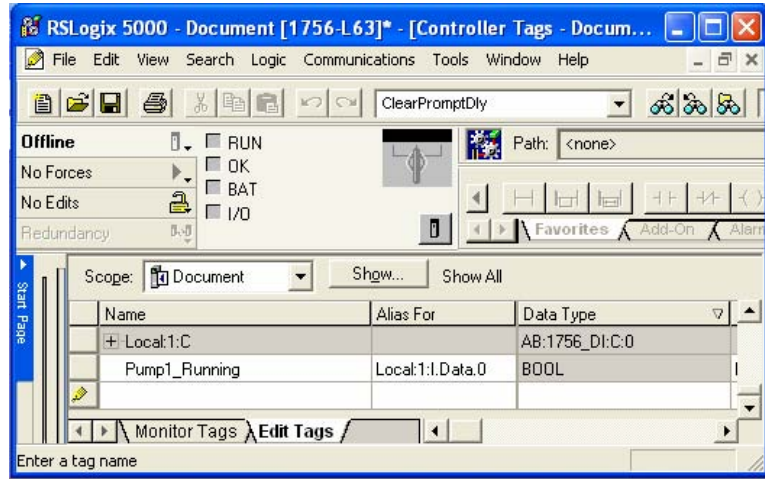
If the 1st input channel is connected to the feedback signal from Pump1 then the following alias can be created by right-clicking on the tag assigned to this address and choosing

New tag which aliases "Local:1:I.Data.0"

from the pull-down menu as shown in the window below.



When the alias tag is created, it will appear in the tag list as follows.



When the code references the feedback from the pump it can use the address *Pump1_Running* rather than *Local:1:I.Data.0* as shown below.



It is possible also to display the aliased input reference as shown below.



Internal Memory Addressing

Internal controller memory locations are available as flag words/bits or DataBlock (DB) words/bits.

Flag Words/Bits

Step 5 internal controller flag address locations have these formats:

- F3.5 for a bit
- FB7 for a byte
- FW10 for a word
- FD for a double word
- S for an extended memory location

These addresses can then be assigned symbolic names via the symbols table.

RSLogix 5000 software does not use this type of address format. Instead, memory locations are defined and referenced solely by the name assigned to them in the tag database.

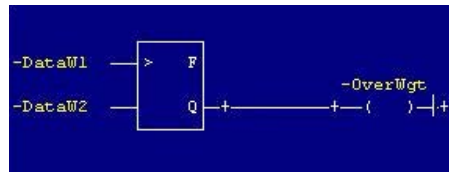
Example

In Step 5, an internal PLC location for storing tank weight could be defined as follows.

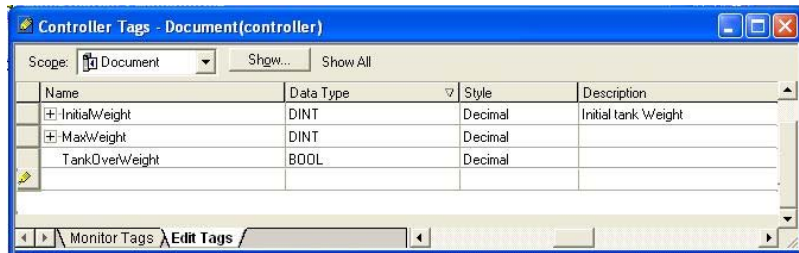
```
Symbols file : D:\TEST1@20.INI
```

Operand	Symbol	Comment
FW 20	DataW1	
FW 22	DataW2	
F 6.3	OverWgt	Tank Overweight

This can be used in the PLC logic to perform a comparison as shown below.



There is no address field in RSLogix 5000 software so the internal registers can be given any name as defined by the user. An example is shown below.



This can be used in the controller logic to perform a comparison as shown below.



Data Blocks

In S5, controller data blocks are often used to store the various data elements relating to a device type such as a pump, motor or valve. Some S5 controllers also have extended data blocks (DX) that store data in a similar manner to ordinary data blocks.

The members of the data blocks can be of various data types:

Data Type	Description
KM	for a binary pattern (16 bits)
KY	for two absolute numbers, one byte each, each in the 0 to 255 range
KH	for a hexadecimal pattern with a maximum of four digits
KS	for two alphanumeric characters
KT	for timer value (BCD coded) units .0 to .3 and values 0 to 999
KC	for a counter value 0 to 999

Example

Data Block DB1 could be assigned to store data relating to valves. The data block would store information such as the open/closed feedback status of the valves, the error status of the valves, or the auto/manual status of the valve.

RSLogix 5000 software allows the creation of User Defined Data Types that will often be useful in replacing Step 5 data blocks. An RSLogix 5000 User Defined Data Type is a custom data structure made up of individual standard data type elements.

Example

To create an RSLogix 5000 data structure to replace a Step 5 data block containing valve information, the following User Data Structure could be generated.



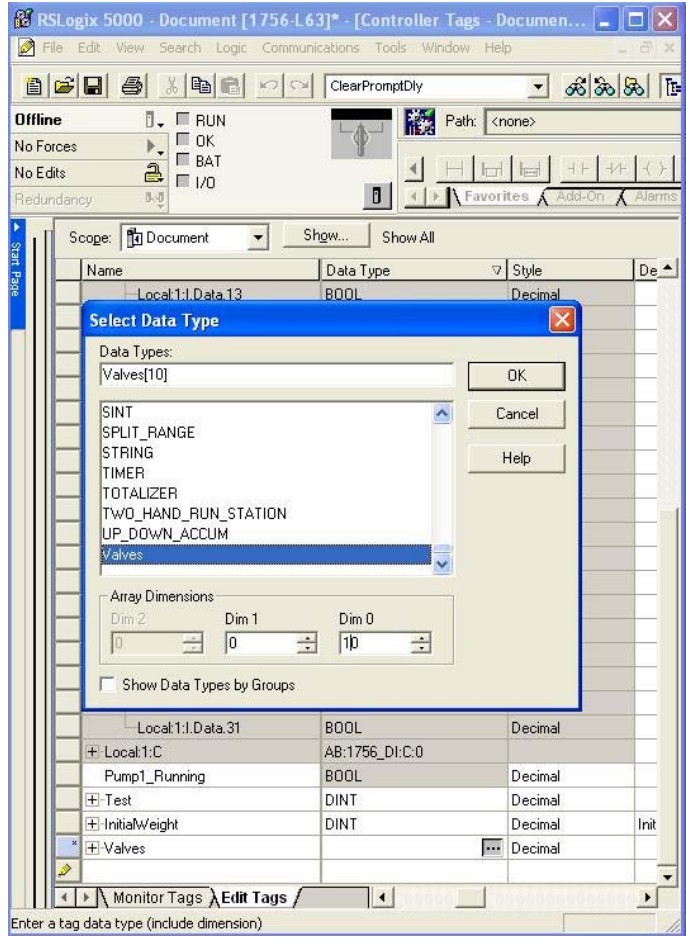
Name: Valves

Description: User Data Type for Valve Commands and Status

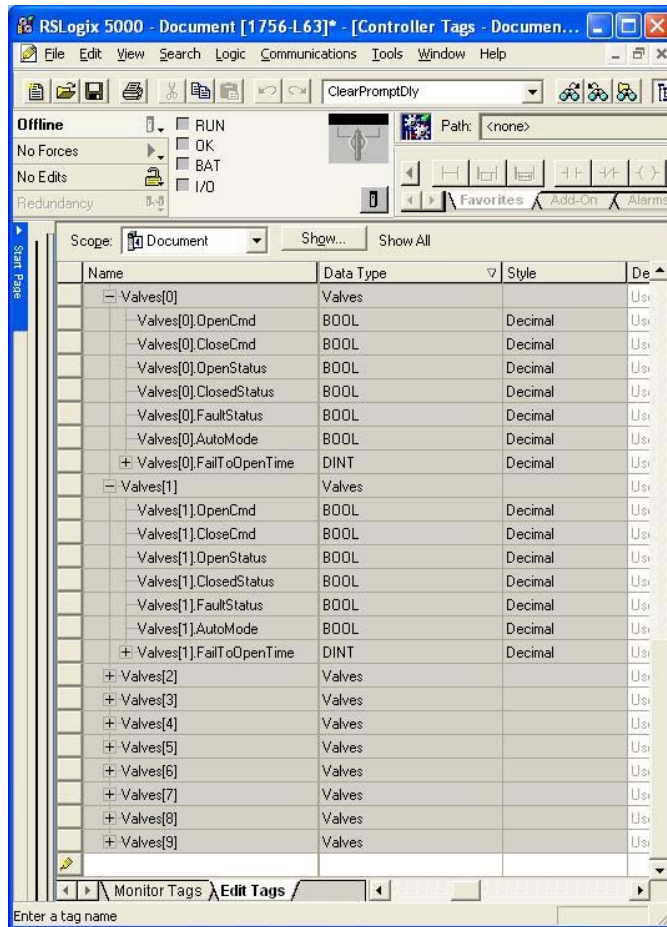
Members: Data Type Size: 8 byte(s)

Name	Data Type	Style	Description
OpenCmd	BOOL	Decimal	
CloseCmd	BOOL	Decimal	
OpenStatus	BOOL	Decimal	
ClosedStatus	BOOL	Decimal	
FaultStatus	BOOL	Decimal	
AutoMode	BOOL	Decimal	
FailToOpenTime	DINT	Decimal	

From the tag database, a tag with this User Data Type can then be created. Multiple instances are automatically generated by selecting the dimension of the array as shown below.



The tag database will automatically create the required addresses for the number of valves specified in the dimensions of the array as shown below.



Programming Languages

This section describes the programming languages that are available with Step 5 and RSLogix 5000 software. All languages are not standard and depends on the version of the software purchased. Selection of the RSLogix 5000 language most suitable to the task will result in easier program design, more rapid coding, and a program that is easier to understand.

There is one significant difference between the Step 5 and RSLogix 5000 languages. In Step 5, Statement List is the native language of the controller. This means that other languages are translated to STL. In RSLogix 5000 software, all the languages are native languages in the controller so each is compiled without reference to any of the others.

Step 5 has these three standard languages:

- Statement List (STL) - is a text-based programming language that uses statements to define what to execute. In German it is referred to as Anweisungsliste (AWL). A similar format, called Structured Text, exists in RSLogix 5000 software.
- Ladder Diagram (LAD) - is a programming method using graphic symbols in a schematic circuit diagram format. In German it is referred to as Kontaktplan (KOP).
- Control System Flowcharts (CFS) - is a programming method using graphic symbols. In German it is referred to as Funktionsplan (FUP). This format also exists in RSLogix 5000 software and is referred to as Function Block Diagram.

GRAPH 5 is an extension of Step 5 and a software package for graphical programming of sequential control systems. GRAPH 5 allows a sequence to be programmed by breaking it down into a series of steps and transitions, which are then programmed by using one of the standard languages (STL, LAD or CFS).

RSLogix 5000 software has four programming languages:

- Ladder Diagram (LD) – comparable to Step 5 LAD, with an expanded instruction set.
- Structured Text (ST) – comparable in structure to higher level computer languages such as C and VBA.
- Function Block Diagram (FBD) – Equivalent to Step 5 CFS.
- Sequential Function Chart (SFC) – comparable to Step 5 GRAPH 5/II.

When converting an application from S5 to Logix it is not necessary to choose the same language in RSLogix 5000 software as was used in Step 5.

RSLogix 5000 Ladder Diagram

Traditionally, Ladder Diagram is used for implementing Boolean combinational logic. In RSLogix 5000 software, it can also be used for sequential logic, motion, data manipulation, and mathematical calculations, although other languages may be more convenient for these tasks.

RSLogix 5000 Structured Text

Structured Text is a high level procedural language that is similar in structure to Visual Basic, Pascal and C. It is used primarily for data manipulation and mathematical calculations, although motion, combinational, and sequential logic can also be programmed in Structured Text.

RSLogix 5000 Function Block Diagram

Function Block Diagram describes graphically a Boolean or mathematical function relating input variables and output variables. Input and output variables are connected to blocks by connection lines. An output of a block may also be connected to an input of another block.

It is a good practice to program PID loops in FBD. It is the most convenient language for process control.

RSLogix 5000 Sequential Function Chart

SFC is a graphical tool for describing sequential logic as a set of states and transitions. Outputs may be assigned to a state, and conditions with Boolean results for transitions to other states defined.

Conversion of STEP 5 Code to RSLogix 5000 Software

For most applications the easiest conversion is to convert the Step 5 program to RSLogix 5000 software with the same programming language. So a Step 5 program written in ladder logic should be converted to an RSLogix 5000 ladder program. Similarly, a Step 5 Control System Flowchart should be converted to an RSLogix 5000 Function Block Diagram program, and a Step 5 Structured Text program should be converted to an RSLogix 5000 Statement List program. However, sometimes it is better to consider another language. In Step 5, PLC code was often programmed in STL for execution speed (no internal translation to native STL language needed) and because it offered more flexibility and more powerful programming compared to the other languages.

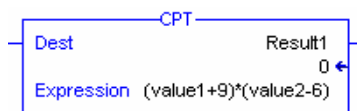
For example, consider RSLogix 5000 SFC programming language if the Step 5 application is mainly a series of discrete steps that must be executed in an easily defined sequence. Also, the one-shot instruction that did not exist in Step 5 and must be programmed in a STL can easily be done in an RSLogix 5000 Ladder diagram. Another consideration would be the PIDE block that only exists in FBD language in RSLogix 5000.

RSLogix 5000 programming languages are more flexible and contain a wider variety of functions than the older Step 5 package. You may reduce converted code complexity with RS Logix 5000 functions unavailable in Step 5.

For example, Step 5 does not allow mathematical calculations such as addition and subtraction to be performed on some data types. You cannot add a floating point number and an integer without first converting them to the same format. RSLogix 5000 software, however, can perform these mathematical calculations on other data types.

RSLogix 5000 software also lets the programmer perform multiple mathematical calculations in one instruction with the Compute (CPT) function to reduce code size and complexity when converting from Step 5.

Calculation in RSLogix5000 Ladder Diagram



To perform a mathematical operation on two operands in Step 5 Statement List, load these operands into Accumulator Registers, perform the required mathematical operation on the Accumulator Registers, and transfer the result from the Accumulator Register to the desired location. RSLogix 5000 Structured Text performs operations directly on the required tags and transfers the result directly to the required destination without using Accumulator Registers as in Step 5.

Calculation in RSLogix5000 Structured Text

```
Result1 := (value1+9) * (value2-6) ;
```

One of the first decisions in developing a Logix 5000 project is to define the functions or operations of your machine or process.

1. Identify the major functions/operations of your machine or process.
2. For each function, choose the programming language that best fits the function.
 - This may require revising the functions list to take advantage of other languages.
 - Use any combination of the languages in the same project.

If the function or group of functions represent	Use this language
Continuous or parallel execution of multiple operations (not sequenced)	Ladder Logic (LD)
Boolean or bit-based operations	
Complex logical operations	
Message and communication processing	
Machine interlocking	
Operations that service or maintenance personnel must interpret to troubleshoot the machine or process	Function Block Diagram (FBD)
Continuous process and drive control	
Loop control	
Calculations in circuit flow	Sequential Function Chart (SFC)
High-level management of multiple operations	
Repetitive sequences of operations	
Batch process	
Motion control with structured text	
State machine operations	Structured Text (ST)
Complex mathematical operations	
Specialized array or table loop processing	
ASCII string handling or protocol processing	

Notes:

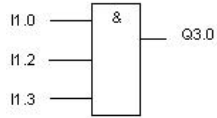
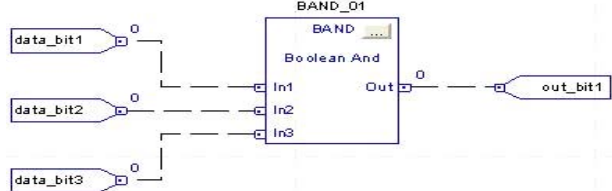
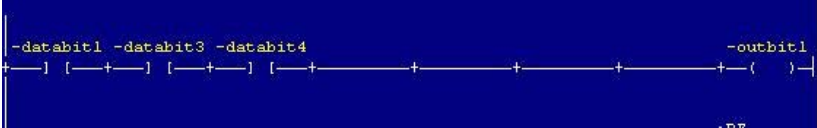

Software Conversions Instructions

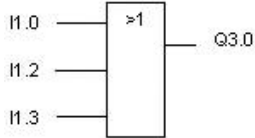
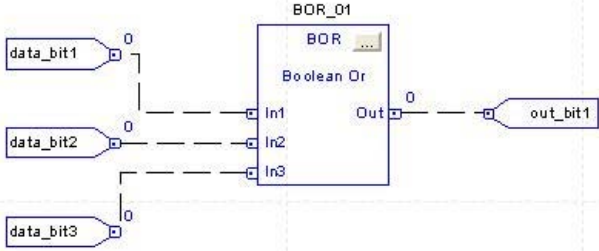
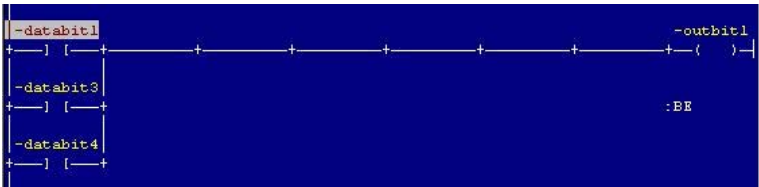

Introduction

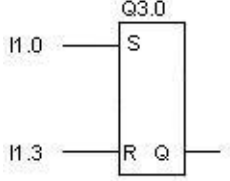
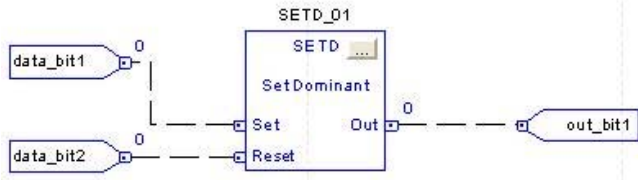
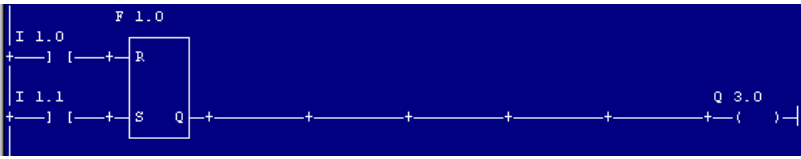
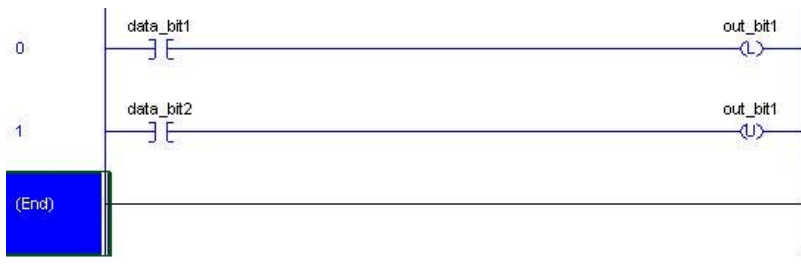
The object of this chapter is to demonstrate how typical programming instruction in S5 can be performed in RSLogix 5000 software. It examines examples of S5 instructions in other programming languages and seeks to find the best RSLogix equivalent in each case.


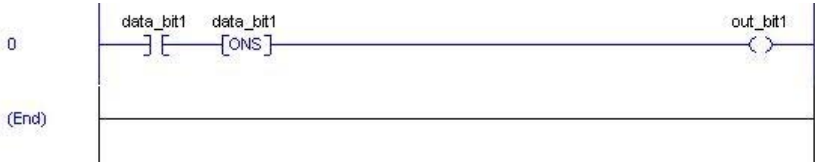
It is not advised to translate the code one on one to RSLogix 5000 software. Some advanced, predefined instructions in RSLogix 5000 software can replace whole parts of Step 5 code (instructions did not exist in Step 5). RSLogix 5000 software also offers more functionality such as aliasing, program/controller tags. Therefore consider other programming techniques rather than imitating the programming in Step 5. Optimizing code by using the standard Logix functionality can improve the scan time substantially.

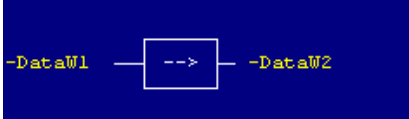
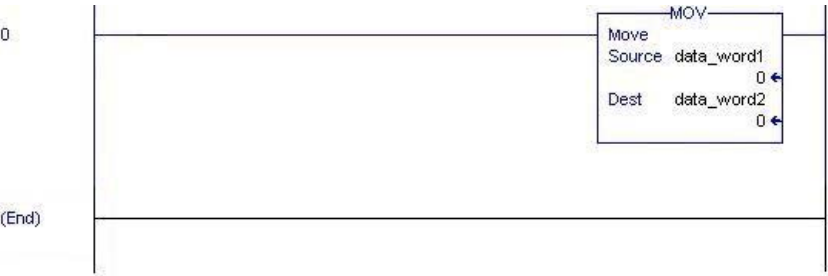
Also see Chapter 4 – [Additional RSLogix 5000 Features](#).

Binary Instruction: AND	
SIMATIC S5	RSLogix 5000
<p>S5 Control System Flowchart</p> 	<p>RSLogix Function Block Diagram</p> 
<p>S5 Statement List</p> <pre>A I1.0 A I1.2 A I1.3 =Q3.0</pre>	<p>RSLogix 5000 Structured Text</p> <pre>IF data_bit1 AND data_bit2 AND data_bit3 THEN out_bit1:= 1; END_IF;</pre>
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

Binary Instruction: OR	
SIMATIC S5	RSLogix 5000
<p>S5 Control System Flowchart</p> 	<p>RSLogix Function Block Diagram</p> 
<p>S5 Statement List</p> <pre> O I1.0 O I1.2 O I1.3 =Q3.0 </pre>	<p>RSLogix 5000 Structured Text</p> <pre> IF data_bit1 OR data_bit2 OR data_bit3 THEN out_bit1:= 1; END_IF; </pre>
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

Binary Instruction: SET/RESET	
SIMATIC S5	RSLogix 5000
<p>S5 Control System Flowchart</p> 	<p>RSLogix Function Block Diagram</p> 
<p>S5 Statement List</p> <pre> A I1.0 S Q3.0 A I1.1 R Q3.0 </pre>	<p>RSLogix 5000 Structured Text</p> <pre> IF data_bit1 THEN out_bit1:= 1; END_IF; IF data_bit2 THEN out_bit1:= 0; END_IF; </pre>
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

Binary Instruction: PULSE	
SIMATIC S5	RSLogix 5000
<p>The pulse instruction is used to detect a change of state of a digital flag and it generates a pulse that is ON for once scan cycle when this change of state occurs. RSLogix 5000 software has a dedicated pulse function while in S5 the function can be constructed as shown below. This is a good example of where it may be an option to use another language in RSLogix 5000 software than in the original S5 project.</p>	
<p>S5 Statement List</p> <p>(F2.0 = 1 for one cycle after I1.0 =1)</p> <pre> A I1.0 AN F4.0 =F2.0 A I.0 =F4.0 </pre>	<p>RSLogix 5000 Structured Text</p> <p>(PulseOutput =1 for one cycle after PulseInput goes from 0 to 1)</p> <pre> IF PulseInput and NOT PulseFlag THEN PulseOutput :=1; ELSE PulseOutput:=0; END_IF; PulseFlag :=PulseInput; </pre>
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

Binary Instruction: Transfer/Move	
SIMATIC S5	RSLogix 5000
S5 Statement List L FW10 T FW12	RSLogix 5000 Structured Text data_word2 := data_word1;
SIMATIC S5 - Ladder	
 <p>The diagram shows a blue rectangular background. On the left, the text '-DataW1' is written in yellow. A horizontal line connects this text to a white square box containing a yellow double arrow pointing to the right. Another horizontal line connects the right side of the box to the text '-DataW2' on the right, also in yellow.</p>	
RSLogix 5000 - Ladder	
 <p>The diagram shows a standard ladder logic structure. On the left, a vertical line has a '0' at the top and '(End)' at the bottom. A horizontal line connects the '0' to a rectangular box labeled 'MOV' at the top. Inside the box, the text 'Move' is at the top, followed by 'Source data_word1' and '0' with a left-pointing arrow. Below that, 'Dest data_word2' and '0' with a left-pointing arrow are shown. A horizontal line continues from the right side of the box to a second vertical line on the right.</p>	

On Delay Timer

The following example shows how a ON DELAY function in SIMATIC S5 can be converted into a similar function in RSLogix 5000 software for Statement List and Ladder formats.

In S5 the time base is determined by the digit after the decimal point in the statement L KT xxx.y, where xxx denotes the preset of the timer and y denotes the time base as one of the following:

- .0 => 0.01sec
- .1 => 0.1sec
- .2 => 1sec
- .3 => 10sec

In RSLogix 5000 software, the time base is in milliseconds.

The RSLogix 5000-equivalent bit/word for each bit/word used in the S5 in the following example is shown below.

Usage of the RES instruction is optional.

Function	S5	RSLogix 5000
Timer Enable	I1.0	data_bit1
Reset	I1.1	data_bit2
Output	Q4.0	out_bit1

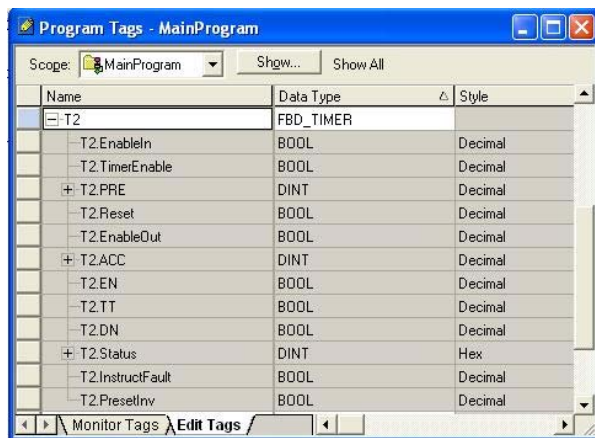
S5 Statement List

```
A I1.0
L KT9.2
SR T3
A I1.1
R T3
A T3
= Q4.0
```

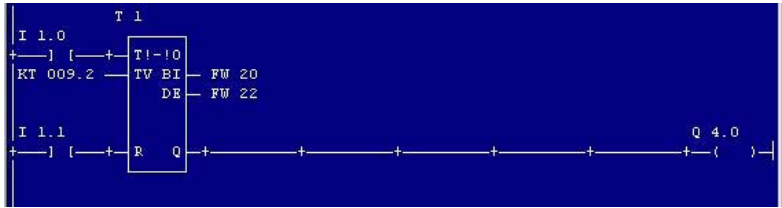
RSLogix 5000 Structured Text

```
T2.Pre := 9000;
T2.Reset := data_bit2;
T2.TimerEnable := data_bit1;
out_bit1 := T2.DN;
```

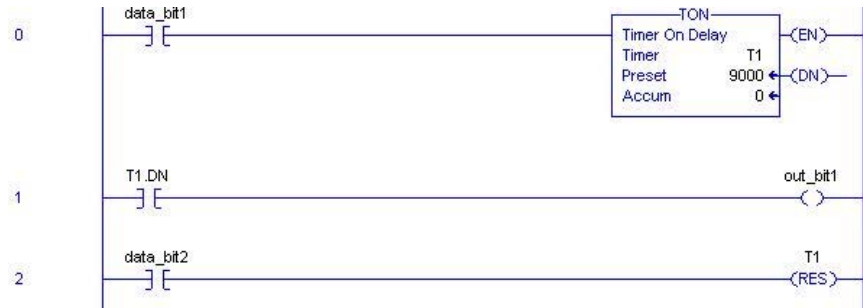
where timer T2 is defined as data type FBD_TIMER in the program tags as follows.



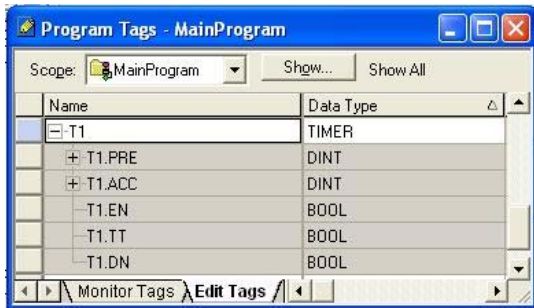
SIMATIC S5 - Ladder



RSLogix 5000 - Ladder



where timer T1 is defined as data type TIMER in the program tags as follows.



Off Delay Timer

The following example shows how an OFF DELAY function in SIMATIC S5 can be converted into a similar function in RSLogix 5000 for Statement List and Ladder formats.

The RSLogix 5000-equivalent bit/word for each bit/word used in the S5 in the following example is shown below.

Usage of the RES instruction is optional.

Function	S5	RSLogix 5000
Timer Enable	I1.0	data_bit1
Reset	I1.1	data_bit1
Output	Q4.0	out_bit1

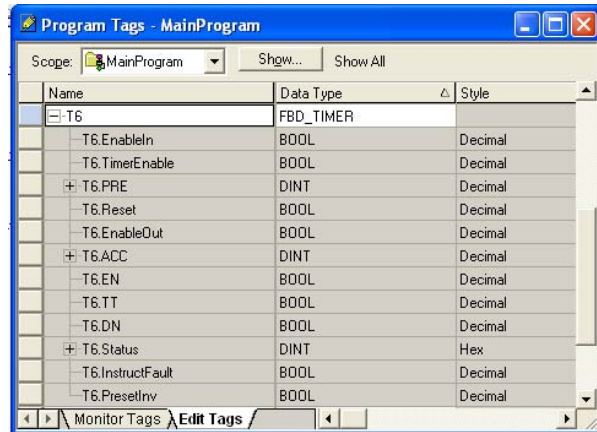
S5 Statement List

```
AN I1.0
L KT9.2
SF T3
A I1.1
R T3
A T3
= Q4.0
```

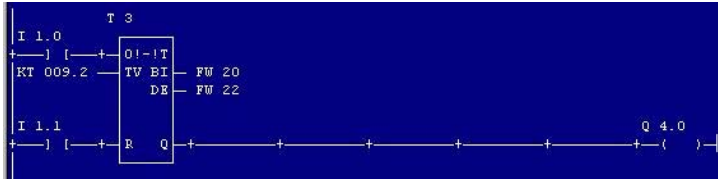
RSLogix 5000 Structured Text

```
T1.Pre := 9000;
T1.Reset := data_bit2;
T1.TimerEnable := data_bit1;
TOFR(T1);
out_bit1 := T1.DN;
```

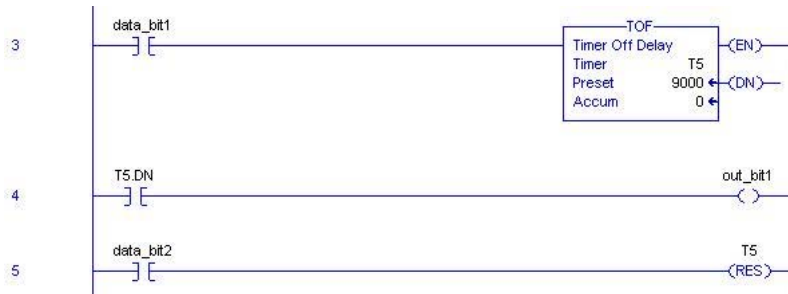
where timer T1 is defined as data type FBD_TIMER in the program tags as follows.



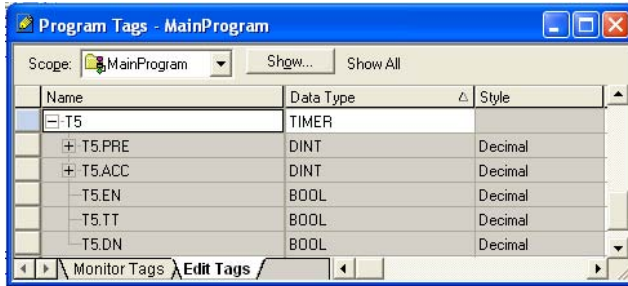
SIMATIC S5 - Ladder



RSLogix 5000 - Ladder



where timer T5 is defined as data type TIMER in the program tags as follows.



Counter - UP

The following example shows how a COUNTUP function in SIMATIC S5 can be converted into a similar function in RSLogix 5000 for Statement List and Ladder formats.

The RSLogix 5000-equivalent bit/word for each bit/word used in the S5 in the following example is shown below.

Function	S5	RSLogix 5000
Counter Preset	DW20	data_word1
Counter Reset	I4.1	data_bit2
Count Up	I4.2	data_bit1
Output	Q2.0	out_bit1

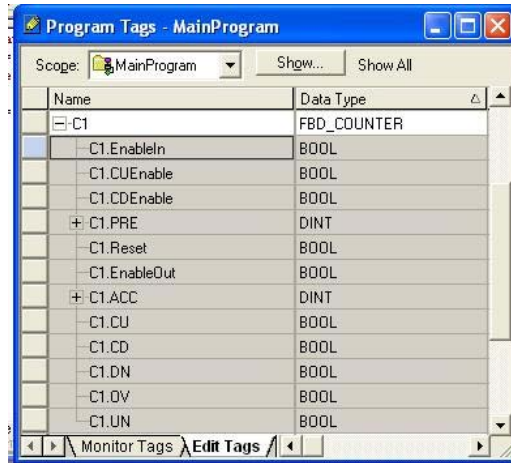
S5 Statement List

```
A I4.0
L DW20
S C1
A I4.1
R C1
A I4.2
CU C1
A C1
= Q4.0
```

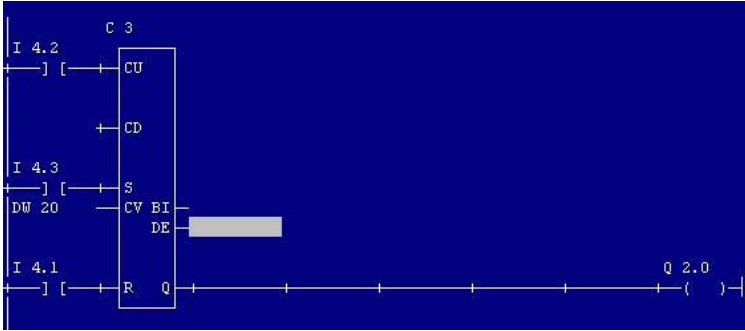
RSLogix 5000 Structured Text

```
C1.Pre := data_word1;
C1.Reset := data_bit2;
C1.CUEnable := data_bit1;
CTUD(C1);
out_bit1 := C1.DN;
```

where timer C1 is defined as data type FBD_COUNTER in the program tags as follows.

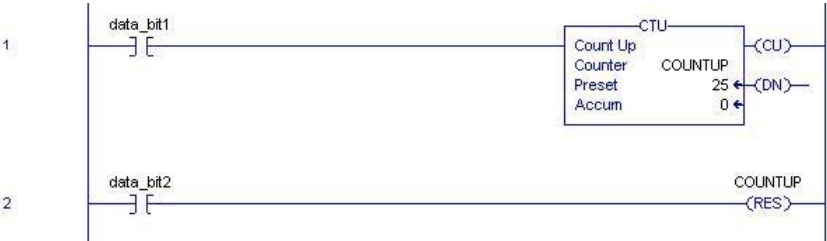


SIMATIC S5 - Ladder



The diagram shows a SIMATIC S5 ladder logic network. It features a counter block 'C 3' with inputs for 'CU' (Count Up) and 'CD' (Count Down). The 'CU' input is connected to a normally open contact labeled 'I 4.2'. The 'CD' input is connected to a normally open contact labeled 'I 4.3'. The 'S' (Set) input is connected to a normally open contact labeled 'DW 20'. The 'CV BI' (Current Value Binary Input) is connected to a normally open contact labeled 'I 4.1'. The 'DE' (Direct Enable) input is connected to a normally open contact labeled 'Q 2.0'. The 'R' (Reset) input is connected to a normally open contact labeled 'Q 2.0'. The output of the counter block is connected to a coil labeled 'Q 2.0'.

RSLogix 5000 - Ladder



The diagram shows an RSLogix 5000 ladder logic network with two rungs. Rung 1 has a normally open contact labeled 'data_bit1' connected to a 'Count Up' coil labeled 'CTU'. The 'CTU' block has a 'Counter' input labeled 'COUNTUP' and a 'Preset' input labeled '25'. Rung 2 has a normally open contact labeled 'data_bit2' connected to a 'Count Down' coil labeled 'COUNTUP'. The 'COUNTUP' block has a 'RES' (Reset) input.

where timer COUNTER is defined as data type COUNTER in the program tags as follows.

Program Tags - MainProgram

Scope: MainProgram Show... Show All

Name	Data Type
- COUNTUP	COUNTER
+ COUNTUP.PRE	DINT
+ COUNTUP.ACC	DINT
- COUNTUP.CU	BOOL
- COUNTUP.CD	BOOL
- COUNTUP.DN	BOOL
- COUNTUP.OV	BOOL
- COUNTUP.UN	BOOL

Monitor Tags Edit Tags

Counter - DOWN

The following example shows how a COUNTDOWN function in SIMATIC S5 can be converted into a similar function in RSLogix 5000 for Statement List and Ladder formats.

The RSLogix 5000-equivalent bit/word for each bit/word used in the S5 in the following example is shown below.

Function	S5	RSLogix 5000
Counter Preset	DW20	data_word1
Counter Reset	I4.1	data_bit2
Count Down	I4.2	data_bit1
Output	Q2.0	out_bit1

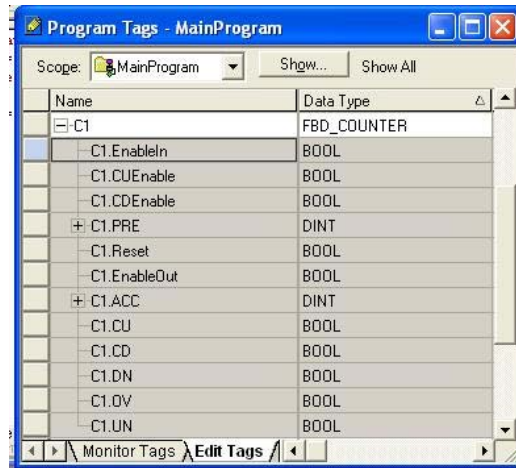
S5 Statement List

```
A I4.0
L DW20
S C1
A I4.1
R C1
A I4.2
CD C1
A C1
= Q2.0
```

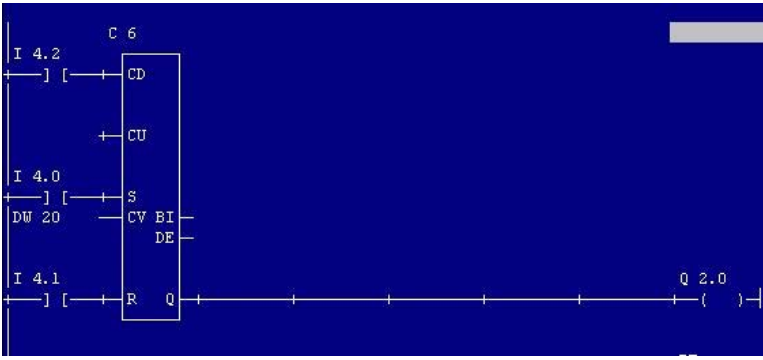
RSLogix 5000 Structured Text

```
C1.Pre := data_word1; => space before :=
C1.Reset := data_bit2;
C1.CDEnable := data_bit1;
CTUD(C1);
out_bit1 := C1.DN;
```

where counter C1 is defined as data type FBD_COUNTER is the program tags as follows.

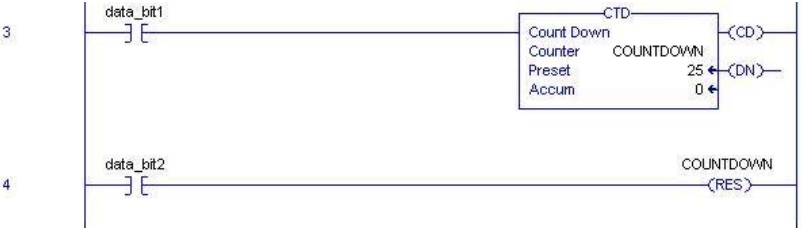


SIMATIC S5 - Ladder



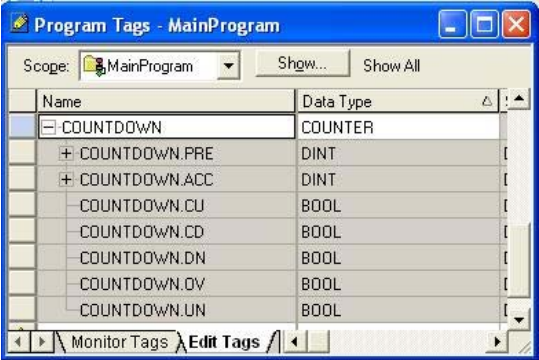
The diagram shows a SIMATIC S5 ladder logic network. It features a counter block labeled 'C 6'. The counter is set to 'CD' (Count Down) and is triggered by input 'I 4.2'. The counter's output is connected to output 'Q 2.0'. Other inputs include 'I 4.0' connected to 'S' (Set), 'DW 20' connected to 'CV BI' (Current Value Binary Input), and 'I 4.1' connected to 'R' (Reset). The counter also has a 'DE' (Direct Enable) input.

RSLogix 5000 - Ladder



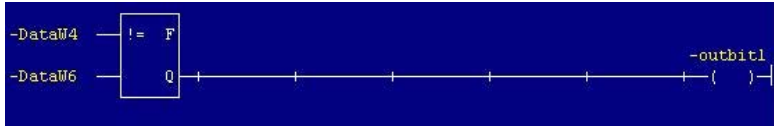

The diagram shows an RSLogix 5000 ladder logic network. It features a counter block labeled 'CTD' (Count Down). The counter is set to 'COUNTDOWN' and is triggered by input 'data_bit1'. The counter's output is connected to output 'COUNTDOWN'. The counter's preset value is set to '25' and is triggered by input 'data_bit2'. The counter's accumulated value is set to '0'. The counter also has a 'RES' (Reset) input.

where counter COUNTDOWN is defined as data type COUNTER is the program tags as follows.



The screenshot shows the 'Program Tags - MainProgram' window. It displays a list of tags and their data types:

Name	Data Type
COUNTDOWN	COUNTER
+ COUNTDOWN.PRE	DINT
+ COUNTDOWN.ACC	DINT
- COUNTDOWN.CU	BOOL
- COUNTDOWN.CD	BOOL
- COUNTDOWN.DN	BOOL
- COUNTDOWN.OV	BOOL
- COUNTDOWN.UN	BOOL

Comparison - Equal		
<p>In the following comparison function the two operands are compared and if found to be equal then the output is set to 1.</p> <p>The RSLogix 5000-equivalent bit/word for each bit/word used in the S5 in the following example is shown below:</p>		
Function	S5	RSLogix 5000
Operand1	IB19	data_word1
Operand2	IB20	data_word2
Output	Q3.0	out_bit1
S5 Statement List	RSLogix 5000 Structured Text	
<pre>L IB19 L IB20 !=F = Q3.0</pre>	<pre>If data_word1 = data_word2 THEN out_bit1 := 1; else out_bit1 := 0; END_IF;</pre>	
SIMATIC S5 - Ladder		
 <p>The diagram shows a SIMATIC S5 ladder logic network. It starts with two normally open contacts labeled '-DataW4' and '-DataW6'. These contacts are connected to a coil labeled '!= F'. The output of this coil is connected to a coil labeled 'Q'. The output of coil 'Q' is connected to a coil labeled '-outbit1'.</p>		
RSLogix 5000 - Ladder		
 <p>The diagram shows an RSLogix 5000 ladder logic network. It starts with a normally open contact labeled '10'. This contact is connected to a coil labeled 'EQU'. The coil 'EQU' has two input fields: 'Source A data_word1' and 'Source B data_word2', each with a '0' and a left-pointing arrow below it. The output of coil 'EQU' is connected to a coil labeled 'out_bit1'.</p>		

Comparison - Greater Than

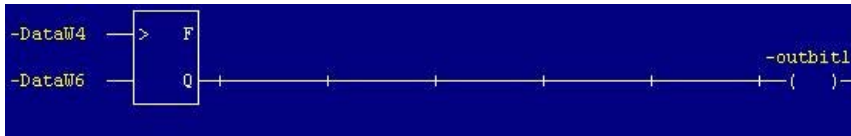
In the following comparison function the Operand 1 is compared to Operand 2 and if Operand 1 > Operand 2 then the output is set to 1.

The RSLogix 5000 equivalent bit/word for each bit/word used in the S5 in the following example is shown below:

Function	S5	RSLogix 5000
Operand 1	IB19	data_word1
Operand 2	IB20	data_word2
Output	Q3.0	out_bit1

S5 Statement List	RSLogix 5000 Structured Text
<pre>L IB19 L IB20 > F = Q3.0</pre>	<pre>If data_word1 > data_word2 THEN out_bit1 := 1; else out_bit1 := 0; END_IF;</pre>

SIMATIC S5 - Ladder



RSLogix 5000 - Ladder



Comparison - Less Than

In the following comparison function the Operand 1 is compared to Operand 2 and if Operand 1 < Operand 2 then the output is set to 1.

The RSLogix 5000 equivalent bit/word for each bit/word used in the S5 in the following example is shown below:

Function	S5	RSLogix 5000
Operand 1	IB19	data_word1
Operand 2	IB20	data_word2
Output	Q3.0	out_bit1

S5 Statement List

```
L IB19
L IB20
< F
= Q3.0
```

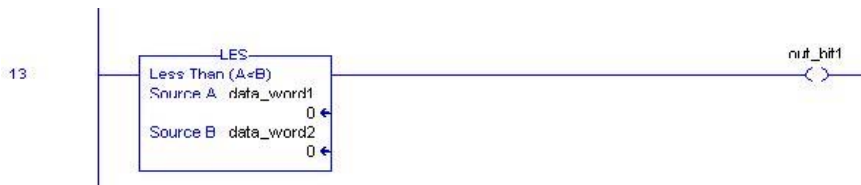
RSLogix 5000 Structured Text

```
If data_word1 < data_word2 THEN
  out_bit1 := 1;
else
  out_bit1 := 0;
END_IF;
```

SIMATIC S5 - Ladder



RSLogix 5000 - Ladder



Comparison - Not Equal

In the following comparison function the two operands are compared and if found to be NOT equal then the output is set to 1.

The RSLogix 5000 equivalent bit/word for each bit/word used in the S5 in the following example is shown below:

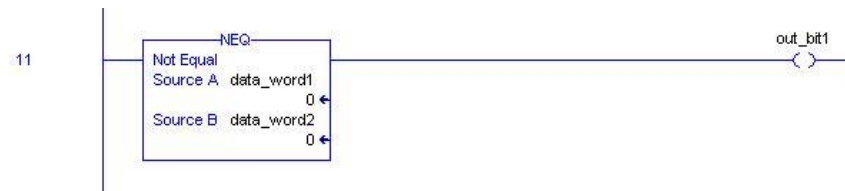
Function	S5	RSLogix 5000
Operand1	IB19	data_word1
Operand2	IB20	data_word2
Output	Q3.0	out_bit1

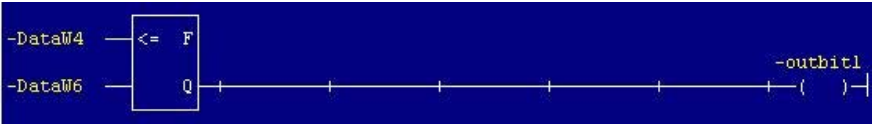
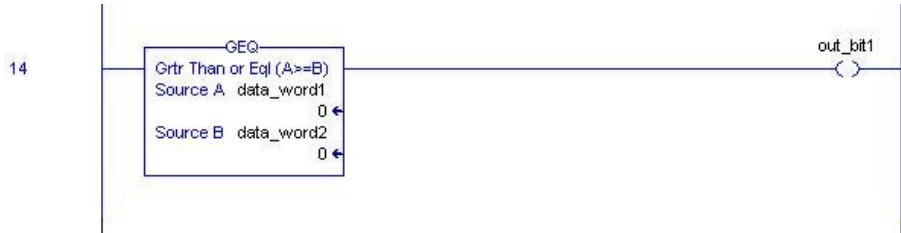
S5 Statement List	RSLogix 5000 Structured Text
<pre>L IB19 L IB20 <>F = Q3.0</pre>	<pre>If data_word1 <> data_word2 THEN out_bit1 := 1; else out_bit1 := 0; END_IF;</pre>

SIMATIC S5 - Ladder



RSLogix 5000 - Ladder



Comparison - Greater Than or Equal To		
<p>In the following comparison function the Operand 1 is compared to Operand 2 and if Operand 1 >= Operand 2 then the output is set to 1.</p> <p>The RSLogix 5000 equivalent bit/word for each bit/word used in the S5 in the following example is shown below:</p>		
Function	S5	RSLogix 5000
Operand1		IB19data_word1
Operand2	IB20	data_word2
Output	Q3.0	out_bit1
S5 Statement List	RSLogix 5000 Structured Text	
<pre>L IB19 L IB20 >= F = Q3.0</pre>	<pre>If data_word1 >= data_word2 THEN out_bit1 := 1; else out_bit1 := 0; END_IF;</pre>	
SIMATIC S5 - Ladder		
 <p>The diagram shows a blue ladder logic rungs. The first rung contains a comparison function block with a left input labeled '-DataW4', a right input labeled '-DataW6', and a function symbol '<= F'. The output of this block is labeled 'Q'. The second rung is a coil labeled '-outbit1'.</p>		
RSLogix 5000 - Ladder		
 <p>The diagram shows a white ladder logic rung. On the left, a vertical bus is labeled '14'. A function block is connected to this bus. The block is titled 'GEG' and 'Grtr Than or Eq (A>=B)'. It has two source inputs: 'Source A data_word1' and 'Source B data_word2', both with '0' and a left-pointing arrow next to them. The output of the block is connected to a coil labeled 'out_bit1'.</p>		

Comparison - Less Than or Equal To

In the following comparison function the Operand 1 is compared to Operand 2 and if Operand 1 <= Operand 2 then the output is set to 1.

The RSLogix 5000 equivalent bit/word for each bit/word used in the S5 in the following example is shown below:

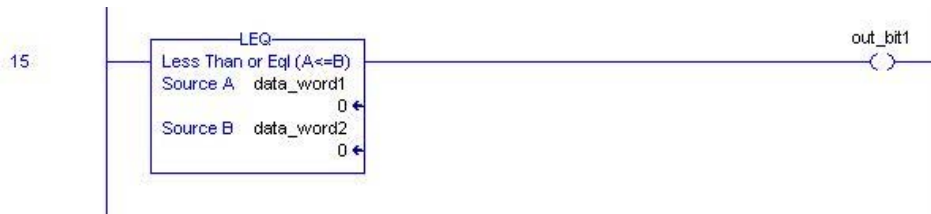
Function	S5	RSLogix 5000
Operand1	IB19	data_word1
Operand2	IB20	data_word2
Output	Q3.0	out_bit1

S5 Statement List	RSLogix 5000 Structured Text
<pre>L IB19 L IB20 <= F = Q3.0</pre>	<pre>If data_word1 <= data_word2 THEN out_bit1 := 1; else out_bit1 := 0; END_IF;</pre>

SIMATIC S5 - Ladder



RSLogix 5000 - Ladder



Jump Operations

Jump operations programmable in S5 can easily be converted to RSLogix 5000 software. The function of these jump instructions is to select to include/exclude the execution of code sections based on the TRUE/FALSE result of other instructions (for example, COMPARE or AND).

Besides these jump functions, RSLogix 5000 software also contains the following instructions that can be used in Structured Text to repeatedly execute a section of code before advancing to the next instruction.

- FOR...DO – executes a section of code a specified number of times.
- WHILE...DO – continually execute a portion of code as long as certain conditions are true.
- REPEAT...UNTIL - continually execute a portion of code until certain conditions are true.

Other similar functions that are available in RSLogix 5000 software are:

- CASE OF – select what instruction to execute based on a numerical value.
- IF...THEN...ELSE – executes a section of code if conditions are true.

Conditional Jump		
<p>In the following example of the Conditional Jump function, the instruction that turns on the output if the input is on is executed only if the two data words are NOT equal.</p>		
Function	S5	RSLogix 5000
Operand 1	IB19	data_word1
Operand 2	IB20	data_word2
Input	I1.0	data_bit1
Output	Q4.0	out_bit1
S5 Statement List	RSLogix 5000 Structured Text	
<pre>L IB19 L IB20 <> F JC =DESA A I1.0 = Q4.0 DESA: NOP 0</pre>	<p>RSLogix 5000 Structured Text has instructions that allow the program to jump to other subroutines but does not permit the program to jump to another point in the same subroutine.</p> <p>The alternative structure is: If data_word1 <> data_word2 then ... (code that is after the NOP 0) else Out_bit1 := data_bit1; End_if;</p>	
SIMATIC S5 - Ladder		
<p>S5 Ladder has instructions that allows the program to jump to other subroutines but does not permit the program to jump to another point in the same subroutine.</p>		
RSLogix 5000 - Ladder		
<p>The diagram shows three rungs of a ladder logic network:</p> <ul style="list-style-type: none"> Rung 16: A normally closed contact labeled 'NEQ' (Not Equal) with 'Source A' as 'data_word1' and 'Source B' as 'data_word2'. The contact is connected to a coil labeled 'DESA' with a '(JMP)' symbol. Rung 17: A normally open contact labeled 'data_bit1' is connected to a coil labeled 'out_bit1'. Rung 18: A coil labeled 'DESA' with a '[LBL]' symbol is connected to a coil labeled 'NOP'. 		

Unconditional Jump		
<p>In the following example of the Unconditional Jump function, the instruction that activates the output (if the input is on) is never executed regardless of whether the two data words are equal.</p>		
Function	S5	RSLogix 5000
Operand 1	IB19	data_word1
Operand 2	IB20	data_word2
Input	I1.0	data_bit1
Output	Q4.0	out_bit1
S5 Statement List		RSLogix 5000 Structured Text
<pre>L IB19 L IB20 >> F JU = DESA A I1.0 = Q4.0 DESA: NOP</pre>		<p>RSLogix 5000 Structured Text has instructions that allow the program to jump to other subroutines but does not permit the program to jump to another point in the same subroutine.</p>
SIMATIC S5 - Ladder		
<p>S5 Ladder has instructions that allow the program to jump to other subroutines but does not permit the program to jump to another point in the same subroutine.</p>		
RSLogix 5000 - Ladder		

Miscellaneous Jump		
<p>Execution of comparison operations in S5 sets condition codes CC0 & CC1.</p>		
S5 Instruction	Description	RSLogix 5000 Implementation
JZ	ACCU 2 = ACCU 1	Equivalent to the equal to comparison instruction (EQ)
JN	ACCU 2 <> ACCU 1	Equivalent to the not equal to comparison instruction (NEQ).
JP	ACCU 2 > ACCU 1	Equivalent to the greater than comparison instruction (GT).
JM	ACCU 2 < ACCU 1	Equivalent to the less than comparison instruction (LT).

Overflow Jump Function JO

With S5 you can work with decimal number in the range -32768 to +32768. If the result of the arithmetic operation is outside this range then S5 sets the overflow bit, OV. The instructions JO can be used to program a jump if this bit is set. This is often used in programs to run error handling code when values exceed their expected range.

With RSLogix 5000 software, the data type DINT can be used to store a 32-bit signed integer value between the range -2,147,483,648 to +2,147,483,647 therefore overflow of data is not usually an issue. However, if required, a combination of a greater than (GT) or less than (LT) comparison instruction and a jump (JMP) instruction can be used in RSLogix to exceed a jump if values exceed a specified range. RSLogix 5000 software also includes an arithmetic overflow status flag (S:V) that is set if a value cannot fit into the destination because the value is greater than the maximum value for the destination or less than the minimum value for the destination. This overflow bit can be used in conjunction with the JUMP (JMP) instruction to replicate the functionality of the S5 JO instruction.

Arithmetic Operations

Arithmetic operations programmable in S5 can easily be converted to RSLogix 5000 software. The main difference between the two systems is that in S5 Statement List each operand is loaded into an accumulator before the operation is performed and then the result can be transferred to the required destination. The corresponding RSLogix 5000 Structured Text code is simpler as it executes the operation on the operands directly. In S5 all operands for an arithmetic operation must be of the same type (that is, 16 bit fixed point, 32 bit fixed point or 32 bit floating point) but in RSLogix 5000 operands of other types can be used in the same instruction.

The following RSLogix 5000 arithmetic operations have no corresponding instruction in S5.

- MOD – Divides one operand by another and places the remainder in the specified destination.
- ABS – Takes the absolute value of the source and places it in the destination.
- CPT – Allows the programmer to define multiple arithmetic operations in one instruction

The following RSLogix 5000 advance arithmetic operations have no corresponding instruction in S5.

- LN – Computes the natural log of an operand and places it in the specified destination.
- LOG – Computes the base 10 log of an operand and places it in the specified destination.
- XPV – Computes the value of operand 1 (X) to the power of operand 2 (Y) and places the result in the specified destination.

S5 has increment (INC) and decrement (DEC) instructions that are used to increment and decrement the value in the accumulator. RSLogix 5000 software does not use the accumulator in the same way as S5 so this functionality can be achieved with the add or subtract instructions in the RSLogix 5000 software.

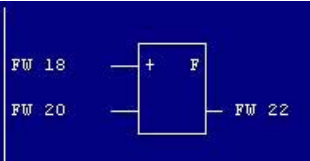
Examples

The following examples illustrate how to convert the following arithmetic functions from S5 to RSLogix 5000.

- Addition
- Subtraction
- Multiplication
- Division

Addition		
<p>S5 uses three addition instructions depending on whether the operands are 16 bit integers, 32 bit integers or floating point numbers. RSLogix 5000 software has one addition instruction that can be used regardless of the format of the operands.</p> <p>Note that you can mix data types, but loss of accuracy and rounding error might occur and the instruction takes more time to execute. You can check the S:V bit to see whether the rest was truncated.</p>		
Function	S5	RSLogix 5000
Operand 1 (16 bit)	FW18	data_word1
Operand 1 (32 bit)	FD30	
Operand 1 (floating point)	FD42	
Operand 2 (16 bit)	FW20	data_word2
Operand 2 (32 bit)	FD34	
Operand 2 (floating point)	FD46	
Output (16 bit)	FW22	data_word3
Output (32 bit)	FD38	
Operand (floating point)	FD50	
S5 Statement List 16 bit fixed point numbers		RSLogix 5000 Structured Text
<pre>L FW18 L FW20 +F T FW22</pre>		<pre>data_word3 := data_word1 + data_word2;</pre>
S5 Statement List 32 bit fixed point numbers		
<pre>L FD30 L FD34 +D T FD38</pre>		
S5 Statement List 32 bit floating point numbers		
<pre>L FD42 L FD46 +G T FD50</pre>		

SIMATIC S5 - Ladder
16 bit fixed point numbers

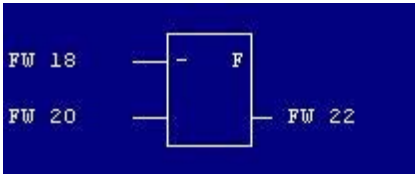


RSLogix 5000 - Ladder



Subtraction		
<p>S5 uses three subtraction instructions depending on whether the operands are 16 bit integers, 32 bit integers or floating point numbers. RSLogix 5000 software has one subtraction instruction that can be used regardless of the format of the operands.</p> <p>Note that you can mix data types, but loss of accuracy and rounding error might occur and the instruction takes more time to execute. You can check the S:V bit to see whether the rest was truncated.</p>		
Function	S5	RSLogix 5000
Operand 1 (16 bit)	FW18	data_word1
Operand 1 (32 bit)	FD30	
Operand 1 (floating point)	FD42	
Operand 2 (16 bit)	FW20	data_word2
Operand 2 (32 bit)	FD34	
Operand 2 (floating point)	FD46	
Output (16 bit)	FW22	data_word3
Output (32 bit)	FD38	
Operand (floating point)	FD50	
S5 Statement List 16 bit fixed point numbers		RSLogix 5000 Structured Text
<pre>L FW18 L FW20 -F T FW22</pre>		<pre>data_word3 := data_word1 - data_word2;</pre>
S5 Statement List 32 bit fixed point numbers		
<pre>L FD30 L FD34 -D T FD38</pre>		
S5 Statement List 32 bit floating point numbers		
<pre>L FD42 L FD46 -G T FD50</pre>		

SIMATIC S5 - Ladder
16 bit fixed point numbers

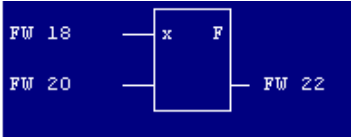


RSLogix 5000 - Ladder

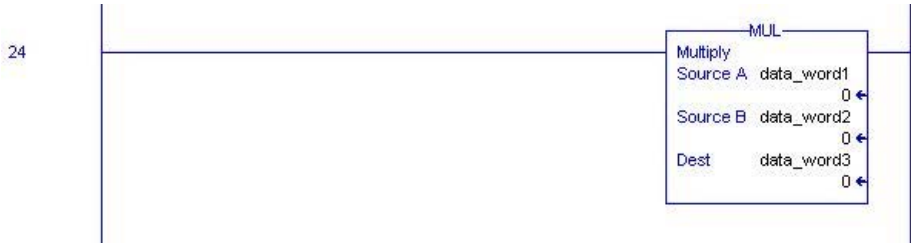


Multiplication		
<p>S5 uses two multiplication instructions depending on whether the operands are 16 bit integers or floating point numbers. RSLogix 5000 software has one multiplication instruction that can be used regardless of the format of the operands.</p> <p>Note that you can mix data types, but loss of accuracy and rounding error might occur and the instruction takes more time to execute. You can check the S:V bit to see whether the rest was truncated.</p>		
Function	S5	RSLogix 5000
Operand 1 (16 bit)	FW18	data_word1
Operand 1 (floating point)	FD30	
Operand 2 (16 bit)	FW20	data_word2
Operand 2 (floating point)	FD34	
Output (16 bit)	FW22	data_word3
Operand (floating point)	FD48	
S5 Statement List 16 bit fixed point numbers		RSLogix 5000 Structured Text
<pre>L FW18 L FW20 xF L FW18 L FW20 xF T FW22</pre>		<pre>data_word3 := data_word1 + data_word2;</pre>
S5 Statement List 32 bit fixed point numbers		
<p>This function can not be performed in S5. Numbers must first be converted to floating point format.</p>		
S5 Statement List 32 bit floating point numbers		<pre>data_word3 := data_word1 * data_word2;</pre>
<pre>L FD30 L FD34 xG T FD38</pre>		

SIMATIC S5 - Ladder
16 bit fixed point numbers

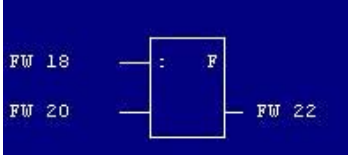


RSLogix 5000 - Ladder



Division		
<p>S5 uses two division instructions depending on whether the operands are 16 bit integers or floating point numbers. RSLogix 5000 software has one division instruction that can be used regardless of the format of the operands.</p> <p>Note that you can mix data types, but loss of accuracy and rounding error might occur and the instruction takes more time to execute. You can check the S:V bit to see whether the rest was truncated.</p>		
Function	S5	RSLogix 5000
Operand 1 (16 bit)	FW18	data_word1
Operand 1 (floating point)	FD30	
Operand 2 (16 bit)	FW20	data_word2
Operand 2 (floating point)	FD34	
Output (16 bit)	FW22	data_word3
Operand (floating point)	FD38	
S5 Statement List 16 bit fixed point numbers		RSLogix 5000 Structured Text
<pre>L FW18 L FW20 :F L FW18 T FW22</pre>		<pre>data_word3 := data_word1 / data_word2;</pre>
S5 Statement List 32 bit fixed point numbers		
<p>This function can not be performed in S5. Numbers must first be converted to floating point format.</p>		
S5 Statement List 32 bit floating point numbers		
<pre>L FD30 L FD34 :G T FD38</pre>		

SIMATIC S5 - Ladder
16 bit fixed point numbers



RSLogix 5000 - Ladder



Block Call Operations

S5 refers to independent portions of code as blocks while RSLogix 5000 software refers to these as subroutines.

S5 uses two block call instructions, Jump Conditional (JC) and Jump Unconditional (JU), to call other program blocks.

RSLogix 5000 software has a similar JumpToSubroutine (JSR) instruction to implement this operation. Comparison or Boolean logic can be placed in front on the RSLogix 5000 JSR instructional to achieve the functionality of the S5 conditional block call function.

At the end of each block S5 has the following instructions:

- Block End (BE) -Identifying the end of a function/program block
- Block End Conditional (BEC) - Causes the current program/function block to end if the proceeding conditions are true.
- Block End Unconditional (BEU) - Causes the current program block to end unconditionally

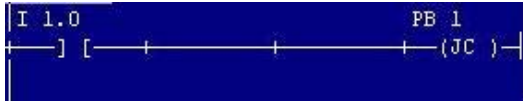
RSLogix 5000 software has a similar ReturnFromSubroutine (RET) instruction to implement this operation. Comparison or Boolean logic can be placed in front on the RSLogix 5000 RET instructional to achieve the functionality of the S5 Block End Conditional and Block End Unconditional function.

It is not necessary to specify the end of an RSLogix 5000 program subroutine. If it is not specified then the code assumes that the routine ends when the last instruction in the routine has been executed.

Block Calls		
Function	S5	RSLogix
Operand 1	F10.3	data_bit1
Operand 2	PB10	Alarms
S5 Statement List 16 bit fixed point numbers		RSLogix 5000 Structured Text
In the following example the code will jump to PB 10 if F10.3 =1. A F10.3 JC PB10		In the following example the code will jump to the Alarms routine if data_bit1 =1. if data_bit1 then JSR (Alarms); end_if;
In the following example the code will jump to PB 10 regardless of the state of F10.3 A F10.3 JU PB10		In the following example the code will jump to the Alarms routine unconditionally. JSR (Alarms);

SIMATIC S5 - Ladder

Conditional Jump



Unconditional Jump



RSLogix 5000 - Ladder

Conditional Jump



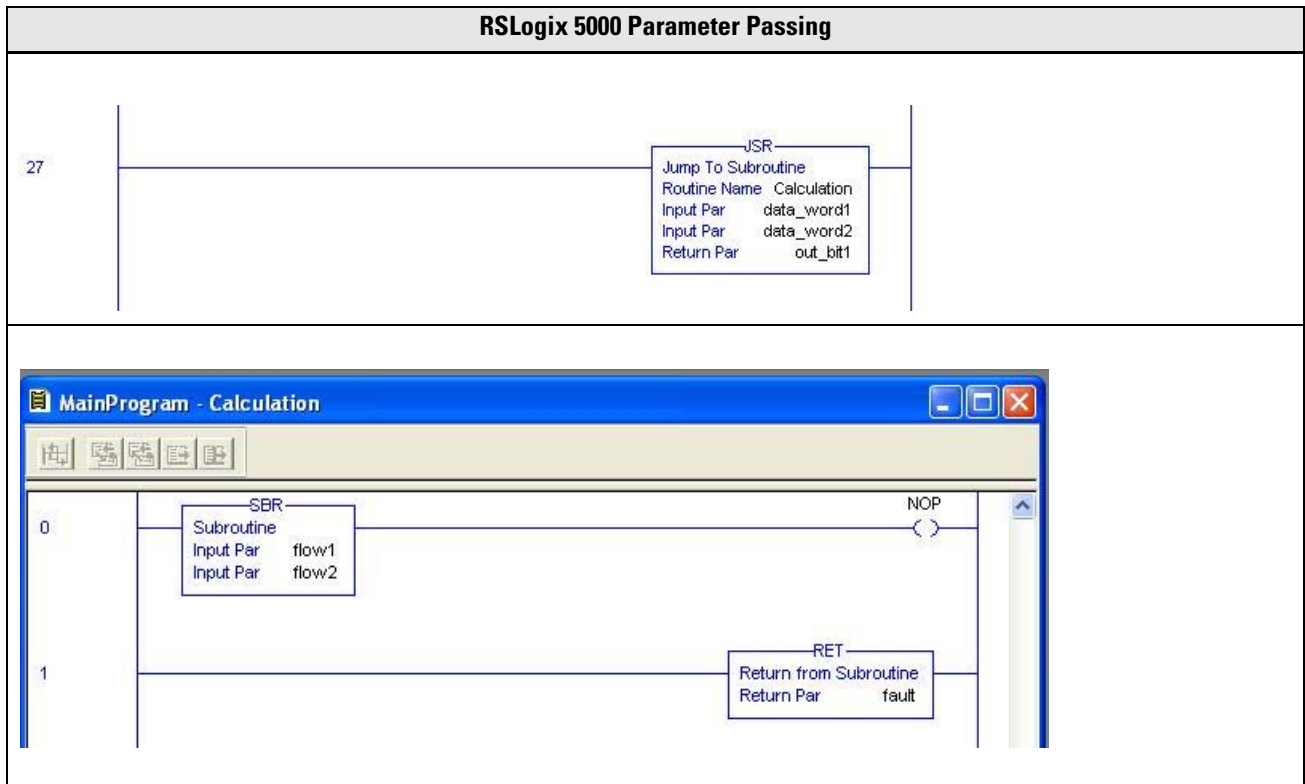
Unconditional Jump



RSLogix 5000 software allows for the option of parameters to be passed to subroutines and also allows for the subroutine to return a result to the calling routine by using the RET function.

The following RSLogix 5000 example illustrates how the JSR/SBR instructions can be used to pass parameters to the called subroutine and return a parameter from the subroutine to the main routine. In this example, the data_word1 value is passed to the parameter flow1 while the data_word2 value is passed to the parameter flow2. The called subroutine, Calculation, evaluates the parameter fault and this value is passed to the parameter out_bit1 in the calling routine.

Instead of using a subroutine, another option is to use the Add-On Instruction functions as discussed in Chapter 2.



Shift Operations

S5 has the following instructions that will shift the contents of the accumulator to the left/right by the number of bits specified by the parameter.



Instruction	Description
SLW	Shifts the contents of the accumulator ACCU1-L (word) to the left by the number of bits specified in the parameter.
SLD	Shifts the contents of the accumulator ACCU1 (double word) to the left by the number of bits specified in the parameter.
SRW	Shifts the contents of the accumulator ACCU1-L (word) to the right by the number of bits specified in the parameter.
SRD	Shifts the contents of the accumulator ACCU1 (double word) to the right by the number of bits specified in the parameter.

This functionality can be achieved in RSLogix 5000 Ladder by using the Bit Field Distribute (BFD) function. In this function the following parameters are specified.

Parameter	Description
Source	the register that contains the bit to be shifted
Source bit	the number of the lowest bit to be moved
Destination	the register that contains the bits after the shift is performed
Length	number of bits to be shifted

The BFD function provides greater flexibility than the S5 shift instructions because with the S5 shift instructions it is only possible to shift the entire contents of a register while with the RSLogix 5000 BFD function it is possible to shift any specified subset of bits.

This BFD function is not available in RSLogix 5000 Structured Text but a similar function called Bit Field Distribute with Target (BFD'T) can produce the same results.


Shift Left	
<p>The following example shows how S5 shifts the bits in FW18 by 4 places to the left and how the equivalent functionality is achieved by using RSLogix software.</p>	
S5	RSLogix 5000
FW18	data_word1
FW20	data_word2
S5 Statement List 16 bit fixed point numbers	RSLogix 5000 Structured Text
<pre>L FW18 SLW 4 T FW20</pre>	<pre>BTD_01.Source := data_word1; BTD_01.SourceBit := 0; BTD_01.Length := 32; BTD_01.DestBit := 4; BTD_01.Target := data_word2; BTD(BTD_01); data_word3 := BTD_01.Dest;</pre>
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

Conversion Operations

Siemens S5 uses the two's complement system in which negative numbers are represented by the two's complement of the absolute value. The following instructions change the sign of a 16 bit word and a 32 bit double word integer.



S5 Instruction	Description
CSW	Form two's complement of ACCU 1-L (bits 0 to 15)
CSD	Form two's complement of ACCU1 (bits 0 to 31)

The RSLogix 5000 Negate (Neg) instruction performs the same function.

Negate											
The following example shows how S5 changes the number sign in DW64 and places the result in DW78, and how the equivalent functionality is achieved by using RSLogix software.											
<table style="width: 100%; border: none;"> <tr> <td style="text-align: center; width: 50%;">S5</td> <td style="text-align: center; width: 50%;">RSLogix 5000</td> </tr> <tr> <td style="text-align: center;">DW64</td> <td style="text-align: center;">data_word1</td> </tr> <tr> <td style="text-align: center;">DW78</td> <td style="text-align: center;">data_word2</td> </tr> </table>	S5	RSLogix 5000	DW64	data_word1	DW78	data_word2					
S5	RSLogix 5000										
DW64	data_word1										
DW78	data_word2										
S5 Statement List 16 bit fixed point numbers	RSLogix 5000 Structured Text										
L DW64 CSW T DW78	data_word2 := -data_word1;										
SIMATIC S5 - Ladder											
											
RSLogix 5000 - Ladder											
<table style="width: 100%; border: none;"> <tr> <td style="width: 10%; vertical-align: middle;">31</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; font-weight: bold; font-size: small;">NEG</td> </tr> <tr> <td style="font-size: x-small;">Negate</td> <td style="font-size: x-small;"> </td> </tr> <tr> <td style="font-size: x-small;">Source</td> <td style="font-size: x-small;">data_word1 0 ←</td> </tr> <tr> <td style="font-size: x-small;">Dest</td> <td style="font-size: x-small;">data_word2 0 ←</td> </tr> </table> </td> </tr> </table>		31	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; font-weight: bold; font-size: small;">NEG</td> </tr> <tr> <td style="font-size: x-small;">Negate</td> <td style="font-size: x-small;"> </td> </tr> <tr> <td style="font-size: x-small;">Source</td> <td style="font-size: x-small;">data_word1 0 ←</td> </tr> <tr> <td style="font-size: x-small;">Dest</td> <td style="font-size: x-small;">data_word2 0 ←</td> </tr> </table>	NEG		Negate		Source	data_word1 0 ←	Dest	data_word2 0 ←
31	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; font-weight: bold; font-size: small;">NEG</td> </tr> <tr> <td style="font-size: x-small;">Negate</td> <td style="font-size: x-small;"> </td> </tr> <tr> <td style="font-size: x-small;">Source</td> <td style="font-size: x-small;">data_word1 0 ←</td> </tr> <tr> <td style="font-size: x-small;">Dest</td> <td style="font-size: x-small;">data_word2 0 ←</td> </tr> </table>	NEG		Negate		Source	data_word1 0 ←	Dest	data_word2 0 ←		
NEG											
Negate											
Source	data_word1 0 ←										
Dest	data_word2 0 ←										



S5 Instruction	Description
DEF	Convert a 16 bit fixed point from BCD into binary
DED	Convert a 16 bit fixed point from BCD into binary

The RSLogix 5000 Convert to Integer (FRD) instruction performs the same function.

BCD to Integer	
S5	RSLogix 5000
DW64	data_word1
DW78	data_word2
S5 Statement List 16 bit fixed point numbers	RSLogix 5000 Structured Text
L DW64 DEF T DW78	This instruction is not available in structured text format.
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

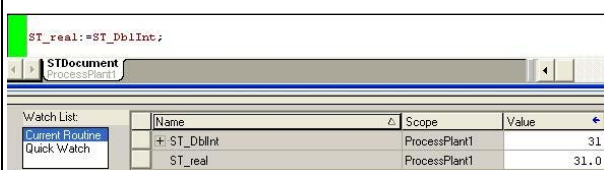

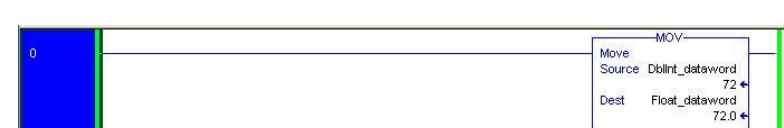
S5 Instruction	Description
DUF	Convert a 16 bit fixed point from binary to BCD
DUD	Convert a 32 bit fixed point from binary to BCD

The RSLogix 5000 Convert to BCD (TOD) instruction performs the same function.

BCD to Integer	
S5	RSLogix 5000
DW64	data_word1
DW78	data_word2
S5 Statement List 16 bit fixed point numbers	RSLogix 5000 Structured Text
L DW64 DUF T DW78	This instruction is not available in structured text format.
SIMATIC S5 - Ladder	
	
RSLogix 5000 - Ladder	
	

FDG- Convert a 16 bit fixed point from binary to BCD



In RSLogix 5000 software, this functionality is achieved by defining a register as having floating point format and moving an integer into it by using the MOVE (MOV) instruction in Ladder or the := instruction in Structured Text.

Integer to Floating Point															
<p>S5</p> <p>DD64</p> <p>DD78</p>	<p>RSLogix 5000</p> <p>data_word1</p> <p>data_floatingpoint</p>														
S5 Statement List 16 bit fixed point numbers	RSLogix 5000 Structured Text														
<pre>L DD64 FDG T DD78</pre>	 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Watch List</th> <th>Name</th> <th>Scope</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Current Routine Quick Watch</td> <td>ST_DblInt</td> <td>ProcessPlant1</td> <td>31</td> </tr> <tr> <td>ST_real</td> <td>ProcessPlant1</td> <td>31.0</td> </tr> </tbody> </table>	Watch List	Name	Scope	Value	Current Routine Quick Watch	ST_DblInt	ProcessPlant1	31	ST_real	ProcessPlant1	31.0			
Watch List	Name	Scope	Value												
Current Routine Quick Watch	ST_DblInt	ProcessPlant1	31												
	ST_real	ProcessPlant1	31.0												
SIMATIC S5 - Ladder															
															
RSLogix 5000 - Ladder															
 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Watch List</th> <th>Name</th> <th>Scope</th> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Current Routine Quick Watch</td> <td>DblInt_dataword</td> <td>ProcessPlant1</td> <td>72</td> <td></td> </tr> <tr> <td>Float_dataword</td> <td>ProcessPlant1</td> <td>72.0</td> <td></td> </tr> </tbody> </table>		Watch List	Name	Scope	Value	Description	Current Routine Quick Watch	DblInt_dataword	ProcessPlant1	72		Float_dataword	ProcessPlant1	72.0	
Watch List	Name	Scope	Value	Description											
Current Routine Quick Watch	DblInt_dataword	ProcessPlant1	72												
	Float_dataword	ProcessPlant1	72.0												

DUD - Convert a floating point number to a 32 bit fixed point format.

In RSLogix 5000 software, this functionality is achieved by defining a tag with Data Type = REAL (as shown below) and transferring it to a tag with Data Type = DINT by using the MOVE (MOV) instruction in ladder or the := instruction in Structured Text.

Name	Data Type	Style
data_floatingpoint	REAL	Float

Floating Point to Integer													
<p>S5 RSLogix 5000</p> <p>DD64 data_floatingpoint</p> <p>DD78 data_word2</p>													
S5 Statement List 16 bit fixed point numbers	RSLogix 5000 Structured Text												
<pre>L DD64 DUD T DD78</pre>	<pre>ST_DblInt:=ST_real;</pre> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Scope</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>ST_DblInt</td> <td>ProcessPlant1</td> <td>23</td> </tr> <tr> <td>ST_real</td> <td>ProcessPlant1</td> <td>23.2</td> </tr> </tbody> </table>	Name	Scope	Value	ST_DblInt	ProcessPlant1	23	ST_real	ProcessPlant1	23.2			
Name	Scope	Value											
ST_DblInt	ProcessPlant1	23											
ST_real	ProcessPlant1	23.2											
SIMATIC S5 - Ladder													
													
RSLogix 5000 - Ladder													
													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Scope</th> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>DblInt_dataword</td> <td>ProcessPlant1</td> <td>17</td> <td></td> </tr> <tr> <td>Float_dataword</td> <td>ProcessPlant1</td> <td>17.1</td> <td></td> </tr> </tbody> </table>		Name	Scope	Value	Description	DblInt_dataword	ProcessPlant1	17		Float_dataword	ProcessPlant1	17.1	
Name	Scope	Value	Description										
DblInt_dataword	ProcessPlant1	17											
Float_dataword	ProcessPlant1	17.1											

Notes:

Additional RSLogix 5000 Features

Introduction

The objective of this chapter is to describe RSLogix 5000 features unavailable in S5. Using these features can result in a more unified RSLogix 5000 application structure than one created by directly converting S5 program components.

There are programming and structure elements in RSLogix 5000 software that differ from S5. The goal of this chapter is to highlight these differences so the programmer understands these features and can use them to achieve the most efficient conversion of an application from S5 to RSLogix 5000 software.

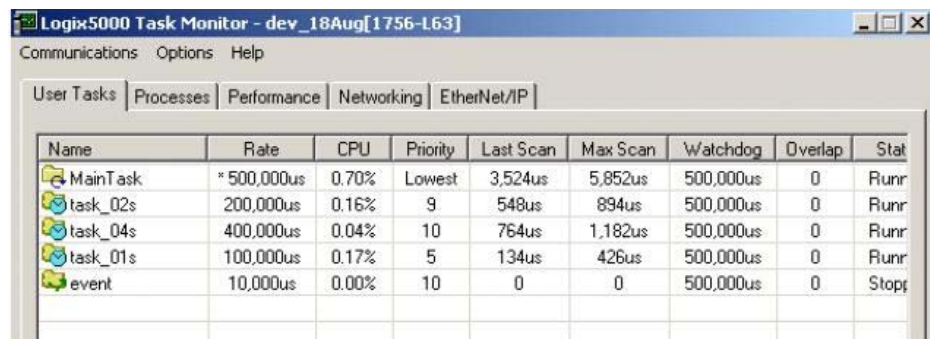
The goal of this chapter is to:

- provide information to the S5 user to make the design process easier and quicker.
- highlight RSLogix 5000 capabilities to avoid programming controller features.

Task Monitor

RSLogix 5000 software includes a tool called Task Monitor that helps analyze scheduled tasks, and much more. The screen shot below shows how information about your controller's tasks can be viewed in one table.

The other tabs provide additional system-level information on your controller's performance. The tool is included as standard on the RSLogix 5000 installation disk.



The screenshot shows the 'Logix5000 Task Monitor' window with the 'Performance' tab selected. The table displays the following data:

Name	Rate	CPU	Priority	Last Scan	Max Scan	Watchdog	Overlap	Stat
MainTask	* 500,000us	0.70%	Lowest	3,524us	5,852us	500,000us	0	Runr
task_02s	200,000us	0.16%	9	548us	894us	500,000us	0	Runr
task_04s	400,000us	0.04%	10	764us	1,182us	500,000us	0	Runr
task_01s	100,000us	0.17%	5	134us	426us	500,000us	0	Runr
event	10,000us	0.00%	10	0	0	500,000us	0	Stop

Asynchronous I/O Updating

In Logix systems, I/O is updated asynchronously with respect to program execution periods. In contrast, S5 uses the traditional PLC approach where an I/O image table is updated at the start of the cycle and input values do not change during an execution of the program.

When converting from S5 to Logix, consider if you must buffer input data so its value remains constant during a program-scan cycle. To keep the value constant, copy the inputs to memory variables at the start of the program and then use these memory variables, instead of the inputs, throughout the code.

DINT Data Type

Logix controllers operate on DINT (32 bit integer) tags more efficiently than on INT (16 bit integer) or SINT (8 bit integer). Use DINT whenever possible, even if the range of values would fit in an INT or a SINT. These data types are provided for IEC61131-3 compatibility reasons, but are internally converted to DINTs before being used by the program. Therefore, code will execute more efficiently in most situations.

String Data Type

The S5 permits ASCII characters to be stored in datablock locations but there are no instructions available in S5 to manipulate text strings. In RSLogix 5000 software, it is possible to define a variable of type STRING and to manipulate these string variables with the following instructions:

- Find –finds the starting position of a string within another string.
- Insert –inserts ASCII characters at a specified location in a string.
- Concat –adds ASCII characters to the end of a string.
- Mid –copies a specified number of characters from a string and stores them in another string.
- Delete –removes ASCII characters from a string.
- File Search and Compare (FSC) – finds a string in an array of strings

In RSLogix 5000 software, the following comparison operations can be performed on string variables and numerical variables. The hexadecimal value of each character is used to determine the value of a string where the hexadecimal values are determined from the ASCII character code table.

- Compare (CMP)
- Equal (EQU)
- Not Equal (NEQ)
- Greater Than Or Equal (GEQ)
- Greater Than (GRT)
- Less Than or Equal (LEQ)
- Less Than (LES)

RSLogix 5000 software also allows the programmer to perform the following conversions on string type variables.

- DINT to String (DTOS) – converts a numeric value to its ASCII equivalent.
- String to Dint (STOD) – converts an ASCII representation to its equivalent integer value.
- Real to String (RTOS) – converts a real value to its ASCII equivalent.
- String to Real (STOR) – converts an ASCII representation to its equivalent real value.
- Upper – converts all the characters in a string to uppercase.
- Lower – converts all the characters in a string to lowercase.

RSLogix 5000 software also provides an instruction called Swap Byte (SWPB) that rearranges the bytes order in a source and places the result in a destination. This instruction cannot be executed on string type variables but on double integers by using the ASCII display format to rearrange characters.

Coordinated System Time (CST)

All S5 controllers, except S5-90U, have a system clock that set and read the current date and time.

RSLogix 5000 software provides the GetSystemValue (GSV) and SetSystemValue (SSV) instructions to access system status data, including date and time. It does this by accessing the data assigned to objects. The two date/time related objects are Coordinated System Time Object and WallClockTime.

Coordinated System Time (CST) Object

The CST is a 64 bit, free-running timer with microsecond granularity on a ControlLogix backplane. A time master, for example, a ControlLogix controller or 1756-SYNCH module generates the timer and sets it on the backplane. All other modules in the chassis have read-only access to the CST and adjust their time reference based on the backplane time reference. CST serves as a relative time reference.

WallClockTime

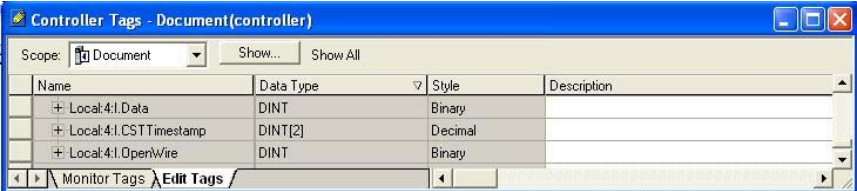
By setting parameters, this object keeps the current UTC time internally while allowing access to other time zones and adjustment for daylight-savings time.

The RSLogix 5000 time-based objects provide the foundation for clock synchronization for multi-CPU systems, accurate motion control functionality, scheduled output switching to 100 μ s accurate, input-event timestamping, scheduled analog sampling, safety I/O monitoring and communication, motion cam position calculations, and Wall Clock Time.

Timestamped Inputs

Timestamp is a RSLogix 5000 functionality that records a change in input data with a relative time of when that change occurred (CST time). With digital input modules, you can configure a timestamp for changes of data and use the timestamp to compare the relative time between data samples. This feature allows for accurate calculations between events to help identify the sequence of events in fault conditions or during normal I/O operations.

Because only one timestamp value is returned to the controller when any input channel changes state, only use timestamping on one input point per module. This value will appear in one of the tags automatically generated when the module is configured as show below.



Name	Data Type	Style	Description
Local:4:I.Data	DINT	Binary	
Local:4:I.CSTTimestamp	DINT[2]	Decimal	
Local:4:I.OpenWire	DINT	Binary	

Timestamping on multiple input points can be achieved with a Logix Sequence of Events module. This is a DC input module that provides timestamping on a per-point basis and the normal functionality of a digital input module. There are two type of Logix Sequence of Events Modules namely:

- 1756-IB16IS0E – 24/48V DC inputs.
- 1756-IH16IS0E – 125V DC inputs.

Scheduled Outputs

With digital output modules, you can configure the module to set the outputs at a scheduled time with reference to the CST.

Scheduled outputs can be used in conjunction with the timestamped inputs feature so that after input data changes state and a timestamp is generated, an output point will actuate at some configured time in the future. You can schedule outputs up to 16.7 seconds into the future.

This allows the user to achieve unparalleled accuracy in linking outputs to time references for applications such as axis positions in motion control, or process control functions, without putting a huge burden on the communication and logic processing systems and related application code.

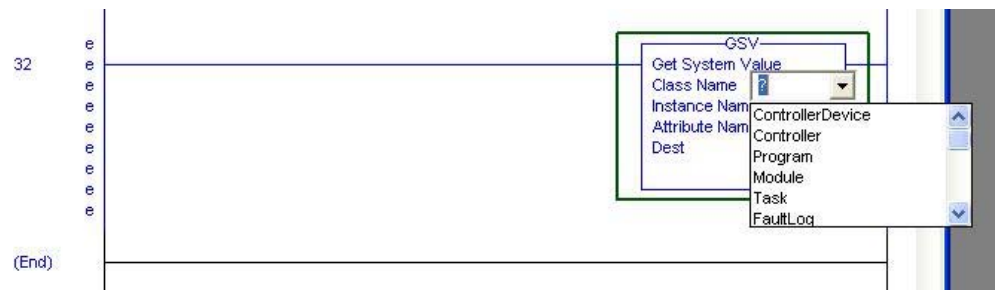
Conversion of System Functions

Introduction

The purpose of this chapter is to examine the most commonly used S5 System Functions and their RSLogix 5000 equivalents, and to show the range of system functions available in RSLogix 5000 software.

Logix System Values

In RSLogix 5000 software, the equivalent of most S5 System Functions will be the GSV (Get System Value) and the SSV (Set System Value) instructions. These instructions access a hierarchy of objects (Classes, Instances and Attributes) built-in to Logix controllers. The GSV and SSV pull-down menus will guide you through parameter selection.



When you enter a GSV/SSV instruction, specify the object and its attribute that you want to access. If there are multiple instances of the same object type, it is necessary to specify the object name. For example, there can be several tasks in your application. Each task has its own TASK object that can be accessed by the task name.

The RSLogix 5000 online help provides detailed information on how to access these system values and how to select the required Attribute Name in the GSV/SSV instruction.

The following objects can be accessed by the GSV/SSV instruction.

Objects	Description
Axis	Provides status information about a servo module axis.
Controller	Provides status information about a controller's execution.
Controller Device	Identifies the physical hardware of the controller.
Coordinate System	The coordinate system tag is used to set the attribute values to be used by the Multi-Axis Coordinated Motion instructions in your motion applications.
CST (CoordinatedSystemTime)	Provides coordinated system time for the devices in one chassis.
DF1	Provides an interface to the DF1 communication driver that you can configure for the serial port.
FaultLog	Provides fault information about the controller.
Message	Provides an interface to set up and trigger peer-to-peer communication.
Module	Provides status information about a module.
Motion Group	Provides status information about a group of axes for the servo module.
Program	Provides status information about a program.
Redundancy	Provides status information about the redundancy system.
Routine	Provides status information about a routine.
Serial Port	Provides an interface to the serial communication port.
Task	Provides status information about a task.
WallClockTime	Provides access to the current time information of the controller's wall clock timer.

Read/Write System Time

In the S5 PLC series the system clock can be read by using FB129 (name: SST:UHR) in conjunction with Data Block 5. A function identified of 1/0 is placed in DW0 and it determines whether the date/time is set/read.

To achieve this functionality with RSLogix 5000 the GSV (Read)/SSV (Write) instruction is used to access the Coordinated System Time (CST) object or the WallClockTime object.

RSLogix 5000 software allows for time intervals of microseconds to be read/set while S5 the smallest time interval is a second. The WallClockTime Object allows the user to specify the time zone and whether to apply daylight savings time. An GSV instruction on the WallClockTime object can then be used to read the Coordinated Universal Time (UTC) or the local time.

Example: Reading the Time

The following ladder diagram shows how the GSV function is used to set the system clock.



The following window shows the result of the above GSV instruction with the date and time stored in the elements of the array GetDateTime[7].

GetDateTime[0] - year

GetDateTime[1] - month

GetDateTime[2] - day

GetDateTime[3] - hour

GetDateTime[4] - min

GetDateTime[5] - sec

GetDateTime[6] - microsecond

The screenshot shows a software window titled 'Controller Tags - Document(controller)'. It features a table with columns for 'Name', 'Value', 'Force Mask', and 'Style'. The 'GetDateTime' array is expanded, showing the following values:

Name	Value	Force Mask	Style
GetDateTime	{...}	{...}	Decimal
+ GetDateTime[0]	2009		Decimal
+ GetDateTime[1]	4		Decimal
+ GetDateTime[2]	10		Decimal
+ GetDateTime[3]	16		Decimal
+ GetDateTime[4]	36		Decimal
+ GetDateTime[5]	37		Decimal
+ GetDateTime[6]	961229		Decimal
+ GetDateTime[7]	0		Decimal
+ SetDateTime	{...}	{...}	Decimal

Example: Setting the Time

The following ladder diagram shows a SSV instruction on the WallClock object being used to set the date and time with the values stored in the elements of the array SetDateTime[7].

SetDateTime[0] - year

SetDateTime[1] - month

SetDateTime[2] - day

SetDateTime[3] - hour

SetDateTime[4] – min

SetDateTime[5] – sec

SetDateTime[6] – microsecond



Save/Load Scratchpad Area

The S5 standard function blocks and user-generated function blocks may use bytes FW200-FW254 to generate and store intermediate results.

FB 38 is used to save this data area if the cyclic program is interrupted by a time/interrupt driven process.

FB 39 is called at the end of the time/event driven process and is used to reload the values to these memory locations.

The reason for these save/load functions is that the time/event driven process could overwrite this temporary storage area and that could result in incorrect intermediate results being present in this area when the normal cyclic process is resumed after a time/interrupt process has been completed.

RSLogix 5000 software does not use a shared temporary shared memory location. Add-On Instructions can have temporary data tags but these will be exclusive to the Add-On Instruction to which they are assigned.

Communication and Networking

S5 Serial Communication

See these sections for information on communication and networks.

S5 Serial Communication Processors (CP 521, CP 523, CP 524, and CP 525)

These S5 communication modules can be used for point-to-point serial data communication with a peripheral device by using a current loop or RS-232 interface. They are often used to send messages to a serial printer. The CP 524 and CP 525 can be configured to communicate with another controller in another rack.

Serial Communication Memory Submodule

The communication processors have receptacles for memory submodules that provide the extra memory capacity required for parameter assignment data and message texts. The configured data is transferred from the memory submodule to the communication processors on startup. These memory modules have a maximum capacity of 16 KB.

Serial Interface

The S5 communication processors connect to a peripheral device over a serial interface that can be a TTY current loop interface or an RS-232C (V.24) interface. The TTY 20mA current loop interface is no longer used in modern peripheral devices.

The signals of both interfaces are transmitted over a common subminiature D socket connector.

Logix Serial Communication

The S5 serial communication modules were designed to communicate with peripheral devices mainly by using RS-232 or the current-based TTY protocols. Advances in communication technology and protocols has made communication from a PLC to a peripheral device with RS-232 uncommon, and TTY nearly obsolete. Compared to modern communication methods, RS-232 and TTY are slow, susceptible to electrical noise, and restrictive in the distance over which they operate. Furthermore, many modern peripheral devices do not provide interfaces to connect by using RS232 or TTY. Even the Siemens SIMATIC HMI Windows-based panel does not provide a TTY interface to communicate with a SIMATIC S5 PLC (with the exception of the MP 370 HMI panel that has a port that can be configured to be RS-232 or TTY in the BIOS).

In converting from a S5 system to a Logix system, the built-in RS-232 port on the Logix controller can replace the S5 communication modules in communicating to a peripheral device.

The following Logix controllers have a built-in serial RS-232 port that eliminates the need for the additional serial communication module as required by the Siemens S5.

- 1756 ControlLogix and 1756 GuardLogix
- 1768 CompactLogix
- 1769 CompactLogix
- PowerFlex 700S Phase 2 with DriveLogix

The most modern peripheral devices communicate via Ethernet and any of the following Logix Ethernet modules can communicate over an Ethernet network.

- 1756-EN2T
- 1756-ENBT
- 1756-EN2F

TIP

The 1756-EN2F provides fiber-optic communication.

The Logix EtherNet/IP module, 1756-EWEB provides remote access via an Internet Browser to tags in a Logix controller.

If it is required to retain a peripheral device with TTY current loop interfaces, a RS-232 to TTY conversion device should be used in conjunction with the RS-232 serial port on the Logix controller.

An example of a possible converter device is the following supplied by ELV:

- ELV V24-20mA interface converter (Order No. 68-176-10)
- ELV AT adapter cable Sub-B 9-pin female to Sub-D 25-pin male (Order No. 68-078-86)

S5 Distributed Control Systems

The Siemens S5 allows for remote I/O to be connected to a centralized CPU with interface modules installed in the centralized rack and expansion racks.

The S5 series programmable controllers can comprise a central controller (CC) and, depending on the configuration, one or more expansion units (EUs). Expansion units are used when there are insufficient central controller slots, or to locate I/O modules as closely as possible to the associated field devices.

To permit communication between the central controller and the expansion unit, a central controller interface module is installed in the central controller rack and an expansion unit interface module is installed in the expansion unit. The type of interface module used is dependent on the type of expansion units racks. The interface modules for the central controller rack and the expansion rack come in pairs. For example, IM300 for the central controller rack combined with the IM310 for expansion racks.

S5 Central Controller Rack Interface Modules	S5 Expansion Unit Rack Interface Modules
IM 300-3	IM306
IM 300-5 (LB)	IM 310
IM 301-3	IM 311
IM 301-5	IM 312-3
IM 302	IM 312-5
IM 304	IM 313
IM 307	IM 314
IM 308 (B)	IM317
	IM318
	IM324

S5 has limited capacity for connecting large amounts of I/O to controllers as shown in the following table.

S5 Controllers 922/928/948	I/O Capacity
DI with process image	1024
DI without process image	7168
AI	448
DO with process image	1024
DO without process image	7168
AO	448

S5 Network Communication

S5 has a low-performance Local Area Network (LAN) called SINEC L1 that operates at 9600 bps and uses the CP 530 communication module. There is also a S5 high-performance LAN called SINEC H1 that operates at 10 Mbps and uses the CP 143 network communication module.

CP 530 Network Communication Module

The CP 530 Network Communication Module is used to interconnect S5 PLC networks to achieve a distributed and hierarchical automation LAN. It executes all the central control functions of the LAN.

The SINEC L1 LAN operates in a master/slave configuration. The master S5 controller must contain a CP530 module and the slave S5 controller may communicate via the controller programming port or via a CP530 module depending on the model of the slave controller.

The Master coordinates and monitors the entire data traffic in the SINEC L1 network, as well as routing through and monitoring of programming functions via the bus. The other programmable controllers participating in the network are automatically designated the slave controllers. These are two ways of sending data packets in the SINEC L1 network:

- From any SINEC L1 node to another
- From any SINEC L1 node simultaneously to all other nodes connected (broadcasting)

The data contained in the data packet may be the status of inputs, outputs, and flags or the contents of data words.

The SIMATIC S5 connects to the LAN bus as follows:

- Master Controller (S5-115U, S5-135U, S5-150U) via the CP 530 module
- S5-115U as slave via the programmer port of the controller or via the CP 530 module
- S5-135U and S5-150U as slave via the CP 530 module
- S5-101U as slave via the programmer port of the computer
- S5-100U (from 102 CPU onwards) as slave via the computer programmer port

Data is transferred between controllers by using SEND/RECEIVE functions.

CP 143 Network Communication Module

The S5 SINEC H1 is an industrial standard LAN enabling S5 controllers and PCs with SINEC H1 communication cards to be connected via transceivers and drop cables to a central bus cable. Individual bus cables can be connected with repeaters, bridges, or star couplers.

The SINEC H1FO LAN allows network segments to be connected via fiber optic cables. These segments permit distances up to 2500 m to be covered.

The S5 CP 143 module fits into the S5 rack and permits the S5-115, S5-135 and S5-155 PLC CPUs to communicate on the SINEC H1 LAN. It operates as a slave processor to the controller and communicates with it via a dual port RAM and the S5 backplane bus. The CP143 configures and sends data to other a partner system vial the SINEC H1/H1FO network

NetLinx Networks

Siemens S5 remote I/O networks can be replaced by Logix networks that permit communication between I/O in multiple platforms, distributed in many locations and connected over multiple I/O links.

NetLinx is the term identifying the Rockwell Automation solution in the area of networking technologies. The following are the primary networks used in Logix systems.

- EtherNet/IP
- ControlNet
- DeviceNet

These networks have a variety of notable features. All are designed under the Common Industrial Protocol (CIP), which enables you to control, configure, and collect data over any of the NetLinx networks. As a result, data can flow between networks without protocol translation software or proxies.

Choosing a ControlNet Network or EtherNet/IP Network for communication between controllers will depend on the requirements of each particular application.

Use the NetLinx Selection Guide, publication [NETS-SG001](#) to determine which network is the best fit for your application.

In converting from an S5 LAN with multiple controllers to a Logix network, the possibility of performing all the required control functions in a central rack that connects to remote I/O racks via ControlNet, EtherNet/IP and/or DeviceNet should be considered.

The Logix controllers have larger memory capacity than the older S5 CPUs so it is possible that software code previously distributed over several controllers could now be executed by one Logix controller or multiple controllers located in a central chassis.

The cabling for an S5 LAN network consisted of a dedicated 4-core cable that had a maximum length between two LAN nodes defined by the cable and bus terminal used.

Logix controllers can communicate over an Ethernet network by using network switches to connect over a wider area with greater speed and reliability.

The SINEC L1 LAN operates at a baud rate of 9.6 Kbps compared to Logix EtherNet cards that can operate at speeds of 10/100 Mbps.

Remote I/O can be connected to a centralized controller by using ControlNet communication modules or remote communication modules.

Logix uses the following ControlNet bridge modules to interface a ControlNet network to a Logix backplane:

- 1756-CN2
- 1756-CNB

The Logix system provides redundant media for ControlNet networks. Redundancy requires no additional programming and is transparent to any devices connected over a ControlNet network. Redundant networks use the following ControlNet bridge modules:

- 1756-CN2R
- 1756-CNBR

NetLinx networks use connections to establish communication between two devices. The number of connections allowed depends on the type of modules and the type of connections that are established.

Remote connections depend on the communication module. The number of connections supported by the module will determine how many connections the controller can access through that module. See the EtherNet/IP Performance User Manual, publication [ENET-AP001](#), for detailed guidance on connections used by network components.

Produced/Consumed Tags in RSLogix

In Logix, the tag values are communicated between networked controllers by configuring the tags as produced tags and consumed tags.

The value of a produced tag can be multicast or unicast across an Ethernet network where it can be read by consumed tags in other controllers on the same network.

Part of the configuration of a consumed tag defines the controller that contains the equivalent produced tag.

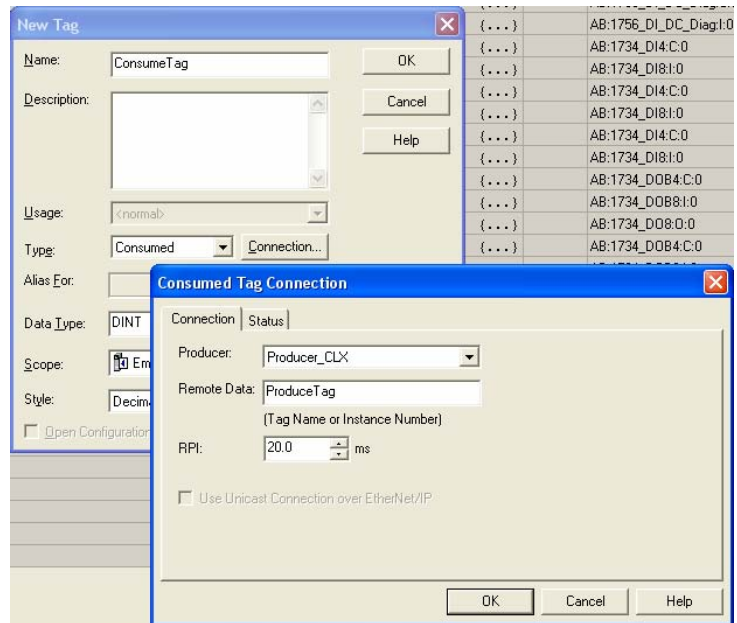
Besides communicating over an Ethernet network, values of produced and consumed tags can be transmitted via ControlNet and the backplane of Logix controllers.

Changing the value of a consuming tag will not affect the producing tag.

No programming is required to configure produce/consume connections in contrast to the complex programming to communicate with S5 communication modules.

The dialog box below provides an overview for configuring a consumed tag.

First, create a tag type of Consumed from the drop-down menu. Select the Connection button to prompt the Consumed Tag Connection dialog box. On the Connection tab, select the producer controller, provide the produced tag name, and set the requested packet interval (RPI). The configuration for a produced tag is similar, but no connection needs to be specified.



Fiber Optic Communication

IM 307/IM 317 Fiber Optic Interface Modules

These S5 modules are used to connect expansion racks to a central rack containing a controller and allows for a transmission link between two nodes of up to 1500 m.

In modern control architecture this arrangement would typically be implemented by using Ethernet modules, such as the Logix 1756-ENBT, in local and remote controller racks. These Ethernet modules would connect to local network switches that would then interconnect with fiber optic links. One advantage of connecting network switches via fiber optic rather than S5 modules is that the same fiber optic link can simultaneously interconnect business computers and servers. A fiber optic link allows for more extensive use, as it is not exclusively reserved for controller use.

Alternatively, Logix 1756-EN2F Ethernet modules can connect local and remote racks. This module uses a fiber optic link that can be used to connect racks up to 2 km apart.

The Logix 1756-SYNCH module uses fiber optic links to transfer data at high speed from one controller chassis to multiple chassis. The high speed data transfer makes it particularly suitable for applications that require data to be transferred at a high speed and in a synchronized manner. Such applications may include distributed motion control and coordinated drive control applications where the rate of data transfer between networked nodes is critical. The SYNCHLINK modules transfer data unidirectional. Therefore they cannot serve as a replacement for communication cards, for example, EN2T, when exchanging data with remote I/O cards.

S5 CP 580 & CP 581 Communication Processor Modules

Although these modules are referred to as communication processors they are mostly used in conjunction with S5 controllers to provide additional computing performance. Direct communication between these modules and the S5 controller takes place via the S5 backplane. These modules are suitable for tasks that require a sizable amount of data handling, such as analyzing and/or processing data or the storage of recipe data. This frees up processing capabilities in the S5 controller. These modules also contain the MS-DOS operating software that allows MS-DOS commands from the S5 controller to activate programs for execution on the CPU 580.

The additional processing power available in the Logix controllers as compared to the S5 controllers means that all the required processing can now be done in the controllers. Multiple Logix controllers can be placed in the same rack or distributed in remote racks to provide as much processing capacity as required.

The relative cost of computers and servers has decreased since the 1980s. Their storage capacity has greatly increased, and the speed and reliability of data transfer also has improved. This means that extensive data storage can now be easily and cheaply achieved by connecting a computer/server to the controller via an Ethernet card, and using the computer/server to store data that can then be monitored by using a SCADA application such as FTVIEW.

IP 265 High Speed Sub Control

This is essentially a high-speed programmable I/O module that was used in S5 to execute small segments of code requiring a rapid execution time.

In a program conversion from S5 to RSLogix 5000 software, functions of this special module can be transferred to and executed by the Logix controller. With its quicker execution times, the Logix controller eliminates the need for a specialized module. If necessary, a dedicated Logix controller could be used to execute any functions requiring a short execution cycle.

Depending on the application, consider transferring the functions of the S5 High Speed Sub Control Module to an Event Task with a short time interval in a Logix controller.

Drive and Loop Control

Encoder/Positioning Modules

These are intelligent I/O modules that can calculate the position of a rotary or linear axis based on electronic encoder input signals. The module then processes this information to calculate and produce the required output for the positional control of speed drives.

The modules determine the position of a rotary cam by counting and processing the high speed pulses generated by an encoder. These high speed pulses are connected to the input channel of the module.

These pulses are at too high a frequency to be processed by standard digital input modules. The encoder/positioning modules are rated for pulse input signals up to 500 kHz, while a standard digital input module is suitable for counting pulses up to a maximum frequency of 100 Hz.

Furthermore, the modules have their own microprocessors that allow them to handle these position control tasks autonomously thus reducing the load to the S5 central processor.

Logix Servo Drive Controllers

The Logix controllers use a range of modules called SERCOS interface modules that serve as a link between the Logix platform and intelligent servo drives.

The SERCOS interface is an open, controller-to-digital drive interface designed for high-speed, real time, serial communication by using noise-immune, fiber-optic cables.

The following Logix Motion Interface Modules can be used to control servomotors.

- 1756-M16SE
- 1756-M08SE
- 1756-M03SE
- 1756-M02AE
- 1756-HYD02
- 1756-M02AS

Using these modules and the RSLogix 5000 motion instruction set, detailed drive-status information can be sent from drive to controller and from controller to drive.

The RSLogix 5000 motion instruction set provides commands for a variety of operations to control servo drives including the following:

- Change the state of an axis – enabling/disabling an axis or setting the output voltage to an axis.
- Control the axis position – stopping/jogging an axis or sending an axis to a home position. Changing the speed, acceleration or deceleration of an axis.
- Initiate action on all axis – perform certain actions, such as shutdown, on all axis.
- Tune an axis – calculate the gain and dynamic limits of an axis.
- Control Multi-Axis Coordinated Motion – start a linear/circulate move for axis of a coordinated system. Calculate the position of one co-coordinated system with respect to another coordinate system.

Replacing S5 Positioning Module with Logix

The S5 servomotor control module relies on hard-wired input signals from an encoder to determine the position of an axis and it outputs a hard-wired analog $\pm 10V$ signal to a motor drive to control the motor. An S5 servomotor control module can control only one motor and can receive only encoder reference data from one encoder.

In RSLogix 5000 software, commands to the motor and status information from the motor are all transmitted along a fiber-optic link. This fiber-optic link can network to multiple drives so that the speed and position of several axis of movement is easily controlled in a coordinated way.

In upgrading S5 servomotor control to a Logix controlled system, it is likely that the encoder and drive must be upgraded to an intelligent drive compatible with the Logix fiber-optic network and its speed and position controlled with a RSLogix 5000 motion interface modules.

In some applications it may be possible to retain the encoder and wire it to an Allen Bradley encoder/counter module (model no.1734-IJ or 1734-IK). The servomotor could then be a controller via an analog output module. However, this would not give the same degree of accuracy or consistency achieved by using a Logix SERCOS drive interface module.

Controller with Motion

The 1756-L60M03SE controller combines a 1756-L6x controller and a SERCOS motion module in a two-slot module. This controller is ideal for small motion systems and can control three SERCOS axes with the included interface. This controller can control as many as six axes if you add an additional motion module,1756-L60M03SE.

S5 Analog Control Modules

S5 has a number of intelligent I/O modules dedicated to the control of analog signals.

IP 243 Analog Module

This module is used for the quick processing of analog input signals. It has eight high-speed analog input channels with a maximum conversion time of 35 μ s. It includes two comparators that allow the comparison of analog values with each other. It is generally used in process control applications where the use of standard analog input and output modules would be too slow to provide the required response.

IP 244 Temperature Controller Module

This module is used for closed-loop PID temperature control and limit monitoring of analog signals. There are 16 analog input channels, 12 of which are configured for thermocouples. The manipulated variables are output in binary form by using pulse width modulation.

IP 252 Closed Loop Control Module

This module can be configured for dedicated fast control loop applications in the S5 system. It has eight analog inputs and eight analog outputs. It can execute eight closed-loop applications in parallel. Because it processes these loops internally, it does not draw on the S5 CPU resources. However, it is possible to transfer data between the CPU and the IP 252. It also has inputs for two tachogenerators whose pulse trains can be processed by the IP 252 to evaluate actual motor speed.

IP 260 Loop Controller Module

This is a single-channel loop controller that is used in the control of process variables such as temperature, pressure, and flow. All the process variables required for closed-loop control are acquired via the four analog input channels.

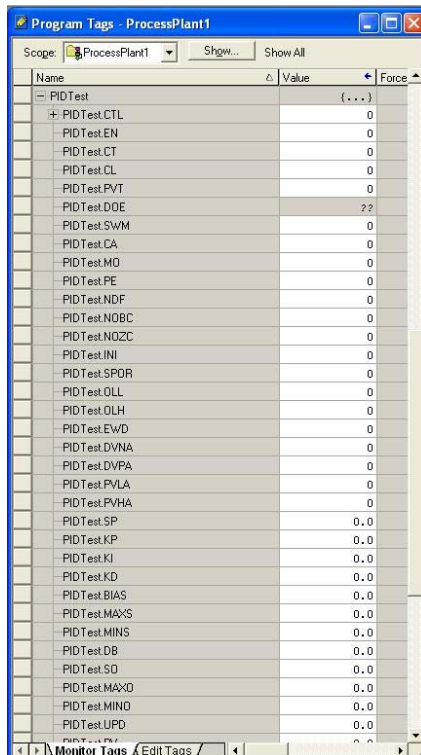
PID control is performed on the process signal and the resultant-calculated manipulated variable is output to the process via two digital controller outputs or via the analog output, depending on the controller type.

Logix Analog Control

The RSLogix 5000 instruction set includes a PID block that executes control over a process variable. The PID calculations are performed on an analog signal coming from a standard analog input module. The PID output is transferred to the corresponding control variable via a standard analog output module. The PID calculations are performed by the controller so no dedicated hardware module is required.



A PID Tag is defined in the tag database and it automatically generates all the required PID elements.



In the FBD language, an enhanced PIDE block is also available.

Logix Replacements for S5 Components

Introduction

This chapter lists S5 components and provides information for selecting suitable Logix replacements.

It is not possible to give a definitive table specifying which Logix controller should replace each S5 controller. The best replacement will be determined only after analyzing the requirements of each particular application.

For example, a stand-alone CPU 943 controller may be replaced by a Logix 1756-L62 controller. If several S5 CPU 943s are being used to control plant areas and they are communicating over a serial interface, it may be advisable to replace all these controllers with a Logix CPU such as a 1756-L62, 1756-L63, or 1756-L64.

The defining criterion for selecting the most suitable model of controller for a particular application is usually memory capacity. Guidance on the amount of memory needed to implement an application for Logix and S5 controllers is provided in Chapter 1.

Controller Memory Use

This section covers the various Logix module types available for replacing S5 components and their advantages.

CPU Modules

The following table suggests which Logix controllers may be suitable replacements for S5 controllers based on the principle of replacing low end capacity S5 controllers with the lowest end capacity Logix controllers, medium level S5 controllers with medium level Logix controllers and high end capacity S5 controllers with high end capacity Logix controllers.

If replacing an S5-90U or S5-95U controller, consider one of the Logix CompactLogix range of controllers. There are three controllers in the 1769 CompactLogix family.

- 1769-L31 – 512 KB Memory and maximum 4 tasks
- 1769-L32 – 700 KB Memory and maximum 6 tasks
- 1769-L35 – 1.5 MB Memory and maximum 8 tasks

There are also two controllers in the 1768 CompactLogix family.

- 1768-L43 – 2 MB Memory and maximum 16 tasks
- 1768-L45 – 3 MB Memory and maximum 16 tasks

S5-90U/95U/100U CPU		Logix CPU	
Cat. No.	Memory	Cat. No.	Memory
CPU 90	4 KB	1769-L3x	2 MB
CPU 95	16 KB	1769-L3x	2 MB
CPU 100	2 KB	1756-L61	2 MB
CPU 102	4 KB	1756-L61	2 MB
CPU 103	20 KB	1756-L61	2 MB
S5-110S CPU		Logix CPU	
Cat. No.	Memory	Cat. No.	Memory
CPU 110S	84 KB	1756-L61S	2 MB
S5-115U CPU		Logix CPU	
Cat. No.	Memory	Cat. No.	Memory
CPU 941	18 KB	1756-L61	2 MB
CPU 942	42 KB	1756-L62	4 MB
CPU 943	48 KB	1756-L62	4 MB
CPU 944	96 KB	1756-L63	8 MB
S5-135U/155U CPU		Logix CPU	
Cat. No.	Memory	Cat. No.	Memory
CPU 921	74 KB	1756-L63	8 MB
CPU 922	88 KB	1756-L63	8 MB
CPU 928	110 KB	1756-L63	8 MB
CPU 928 3UA21	110 KB	1756-L63	8 MB
CPU 928B	110 KB	1756-L63	8 MB
CPU 928B 3UB21	110 KB	1756-L63	8 MB
CPU 948-1	640 KB	1756-L64	16 MB
CPU 948-2	1664 KB	1756-L64	16 MB
CPU 948B 3UA13	64 KB	1756-L63	8 MB
CPU 948B 3UA23	64 KB	1756-L63	8 MB
CPU 945 7UA1	256 KB	1756-L64	16 MB
CPU 945 7UA2	384 KB	1756-L64	16 MB
CPU 946	896 KB	1756-L64	16 MB
CPU 947	896 KB	1756-L64	16 MB

Digital Input Modules

The following table suggests Logix replacements for S5 digital DC input modules based on number of channels and input voltages.

Siemens S5 Digital Input Modules			Logix Digital Input Modules		
Cat. No.	Inputs	V DC	Cat. No.	Inputs	V DC
6ES5 420-4UA13/14	32	24	1756-IB32	32	12/24
6ES5 430-4UA13/14	32	24	1756-IB32	32	12/24
6ES5 420-8MA11	4	24	1756-IB16	16	12/24
6ES5 421-8MA12	8	24	1756-IB16	16	12/24
6ES5 422-8MA11	16	24	1756-IB16	16	12/24
6ES5 430-8MB11	4	24...60	1756-IB16	16	12/24
6ES5 431-4UA12	16	24...60	1756-IB16	16	12/24
			1756-IC16	16	30...60
6ES5 431-8MA11	8	24	1756-IB16	16	12/24
6ES5 432-4UA12	32	24 / alarm	1756-IB16	16	12/24
6ES5 433-8MA11	8	5...24	1756-IG16	16	5V
6ES5 434-4UA12	32	5...15	1756-IG16	15	5V
			1756-IV16	16	10...30

The following table suggests Logix replacements for S5 digital AC input modules based on number of channels and input voltages.

Siemens S5 Digital Input Modules			Logix Digital Input Modules		
Cat. No.	Inputs	V AC	Cat. No.	Inputs	V AC
6ES5 435-4UA12	16	24...60	1756-IN16	16	10...30
6ES5 436-4UA12	16	115...230	1756-IA16	16	79...132
6ES5 436-4UB12	8	115...230	1756-IA8D	8	79...132

Digital Output Modules

The following table suggests Logix replacements for S5 digital DC output modules based on numbers of channels and output voltages.

Siemens S5 Digital Outputs Modules (DC)			Logix Digital Outputs Modules (DC)		
Cat. No.	Outputs	V DC	Cat. No.	Outputs	V DC
6ES5 440-8MA12	4	24V	1756-OB8	8	12/24V
6ES5 440-8MA22	4	24V	1756-OB8	8	12/24V
6ES5 441-8MA11	8	24V	1756-OB8	8	12/24V
6ES5 441-4UA13/14	32	24V	1756-OB32	32	12/24V
6ES5 450-8MB11	4	24...60V	1756-OC8	8	30...60V
6ES5 451-8MA11	8	24V	1756-OB8	8	12/24V
6ES5 451-4UA13/14	32	24V	1756-OB32	32	12/24V DC
6ES5 453-8MA11	8	5...24V	1756-OG16	16	5V DC TTL
6ES5 453-4UA12	16	24V	1756-OB16	16	12/24V
6ES5 454-4UA13/14	16	24V	1756-OB16	16	12/24V
6ES5 457-4UA12	16	24...60V	1756-OC8	8	30...60V
6ES5 458-4UA12	16	24V Relay	1756-OW16I	16	10...265V AC 5...150V DC (Relay)

Siemens S5 Digital Output Modules (AC)			Logix Digital Output Modules (AC)		
Cat. No.	Outputs	V AC	Cat. No.	Outputs	V AC
6ES5 451-8MD11	4	115...230V	1756-OX8I	8	10...265V
6ES5 455-4UA12	16	24...60V	1756-ON8	8	10...30 V
6ES5 456-4UA12	16	115...230V	1756-OW16I	16	10...265V
6ES5 456-4UB12	8	115...230V	1756-OX8I	8	10...265V
6ES5 458-4UC11	16	250V AC (Relay)	1756-OW16I	16	10...265V AC 5...150V DC (Relay)

Digital Relay Output Modules

The following table lists the Logix replacements for S5 digital relay output modules based on number of channels and output voltages.

Siemens S5 Digital Relay Output Modules (DC/AC)			Logix Digital Relay Output Modules (DC/AC)		
Cat. No.	Outputs	V DC	Cat. No.	Outputs	V DC
6ES5 452-8MR11	4	(Relay) 30V DC / 230V AC	1756-OX8I	8	10...265V AC 5...150V DC (Relay)
6ES5 452-8MR11	8	(Relay) 30V DC / 230V AC	1756-OX8I	8	10...265V AC 5...150V DC (Relay)
6ES5 458-8MR12	16	(Relay) 250V AC	1756-OW16I	16	10...265V AC 5...150V DC (Relay)

Analog Input Modules

The following table suggests Logix replacements for S5 analog input modules based on number of channels and input ranges.

Siemens S5 Analog Input Modules			Logix Analog Input Modules		
Cat. No.	Channels	Range	Cat. No.	Channels	Range
460-4UA13	8 – single	See range cards	1756-IF8	8	± 10.25V 0...10.25V 0...5.125V 0...20.5 mA
465-4UA12	16 – voltage/current	See range cards	1756-IF16	16	±± 10.25V 0...10.25V 0...5.125V 0...20.5 mA
	8 – Pt100 resistance	See range cards	1756-IR6I	6	1...487 Ω 2...1000 Ω 4...2000 Ω 8...4080 Ω
465-4UA13	16 – voltage/current	See range cards	1756-IF16	16	± 10.25V 0...10.25V 0...5.125V 0...20.5 mA
	8 – Pt100 resistance	See range cards	1756-IR6I	6	1...487 Ω 2...1000 Ω 4...2000 Ω 8...4080 Ω
463-4UA12	4 – single	0...1V, 10V, 20mA	1756-IF8	8	± 10.25V 0...10.25V 0...5.125V 0...20.5 mA
463-4UB12	4 - single	4...20mA	1756-IF8	8	± 10.25V 0...10.25V 0...5.125V 0...20.5 mA

Siemens S5 Analog Input Modules			Logix Analog Input Modules		
Cat. No.	Channels	Range	Cat. No.	Channels	Range
466-3LA11	16 – voltage/current	0...1.25V, 2.5V, 5V, 10V	1756-IF16	16	± 10.25V 0...10.25V 0...5.125V 0...20.5 mA
	8 – Pt100 resistance	See range cards	1756-IR6I	6	1...487 Ω 2...1000 Ω 4...2000 Ω 8...4080 Ω
466-4LA11	16 – voltage/current	0...20mA 4...20mA	1756-IF16	16	± 10.25V 0...10.25V 0...5.125V 0...20.5 mA
	8 – Pt100 resistance	See range cards	1756-IR6I	6	1...487 Ω 2...1000 Ω 4...2000 Ω 8...4080 Ω

Siemens S5 Analog Input Range Cards

These do not have Logix replacements as the ranges in the Logix analog cards are software configurable. Choose a Logix card that meets the specifications.

Cat. No.	Range Card	Range/Sensor
460-4UA13	498-1AA11	±12.5, 50, 500 mV, Pt100
465-4UA12 465-4UA13	498-1AA11	± 50/500 mV, Pt100
	498-1AA21	± 1V
	498-1AA31	± 10V
	498-1AA41	± 20 mA
	498-1AA51	4...20 mA
	498-1AA61	± 5V
	498-1AA71	4...20 mA

Analog Output Modules

The following table suggests Logix replacements for S5 Analog Output Modules based on numbers of channels and output ranges.

Siemens S5 Analog Output Modules			Logix Analog Output Modules		
Cat. No.	Channels	Range	Cat. No.	Outputs	Range
470-4UA12	8	± 10V 0...20 mA	1756-OF8	8	± 10.4V 0...21 mA
470-4UA13	8	± 10V 0...20 mA	1756-OF8	8	± 10.4V 0...21 mA
470-4UB12	8	± 10V	1756-OF8	8	± 10.4V 0...21 mA
470-4UB13	8	± 10V	1756-OF8	8	± 10.4V 0...21 mA
470-4UC12	8	1...5V 4...20 mA	1756-OF8	8	± 10.4V 0...21 mA
470-4UC13	8	1...5V 4...20 mA	1756-OF8	8	± 10.4V 0...21 mA

Other Communication Modules

Introduction

This section contains an overview of the available modules that provide communication to S5 I/O and peer-to-peer communication.

If you are using a protocol to which you want to connect but is not described in this manual, please check the websites of our encompass partners (such as Woodhead/Molex, ProSoft, or HMS).

In-rack Modules

This section contains an overview of the available ControlLogix in-rack modules that provide communication to S5 networks.

Profibus Network

This section describes modules that connects to a Profibus network.

Woodhead/Molex SST-PFB-CLX-RLL DP

One way of connecting to S5 I/O on Profibus is using a PROFIBUS DP Scanner for the Logix Controller in the main rack. This is a third-party module produced by Brad Communications (Molex/ Woodhead, Part Number SST-PFB-CLX-RLL), which works with RSLinx software 2.2 OEM/Professional or later.

This module slots directly into a local or remote ControlLogix chassis. It enables communication between a CLX processor and DP remote I/O devices on a PROFIBUS network, acting as a PROFIBUS DP remote I/O scanner.

The SST-PFB-CLX-RLL DP supports a maximum of 125 slaves with a maximum of 244 bytes of input data and 244 bytes of output data per slave and it operates on all Profibus data rates up to 12 Mbps. As a Master, it supports up to a maximum of 1984 bytes of input data and 1968 bytes of output data within four I/O pages. The SST-PFB-CLX-RLL can act on a PROFIBUS network as a DPV1 Class 1 & Class 2 master via generic CIP messages (manages up to 16 simultaneous Class 1 commands and up to 16 simultaneous Class 2 commands and up to 32 simultaneous Class 2 connections). The module can also act as a DPV0 slave.

The Profibus DP/PA Network can be configured by the free SST Configuration Tool. The firmware is field-upgradeable with the HyperTerminal.

More information on this module can be found at the website <http://www.woodhead.com>. For software/firmware/manual downloads, go to <http://www.mysst.com/download>.



Profibus DP Scanner Module

ProSoft MVI56-PDPMV1

The MVI56 Profibus DPV1 Master Communication Module is a powerful communication interface for ControlLogix platform processors.

The module supports complete Master specifications according to IEC 61158. Acyclic parameter data can be transferred with Class 1 or Class 2 DPV1 services, so processors can easily communicate with slave devices supporting Profibus DPV0/V1 protocol.

The module acts as an input/output module between the Profibus network and the ControlLogix processor. Data transfer from the processor is asynchronous with



the I/O data transfer on the Profibus network. It supports all standardized baud rates, up to 12 Mbps.

MVI56-PDPMV1 Specifications

Specification Type	Description
General	Single Slot-1756 backplane compatible.
	The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module.
	Ladder Logic is used for data transfer between module and processor. Sample ladder file (add-on instruction) included.
	Local or remote rack.
	Configurable I/O backplane sizes allow optimal performance for remote rack applications.
	CIPconnect (tm) enabled. Module supports communication with ProSoft Configuration Builder and FDT comDTM via Ethernet routing by using Rockwell Automation Ethernet interface.
Profibus DP Master	Up to 125 Slaves can be connected with a repeater.
	Up to 1536 cyclic bytes input and 1536 bytes output data.
	Supports Extended Diagnostic Data.
	RS-485 optically isolated Profibus Interface with on board DC-DC converter.
	Acyclic communication (DPV1), Read and Write.
	Alarm Handling (DPV1).
	Supports Sync and Freeze commands.
	Supports PROFIdrive 3.1 compliant parameter read and write operations.
Supports Multicast and Broadcast telegrams (DPV1).	

Siemens Industrial Ethernet/Modbus TCP/IP

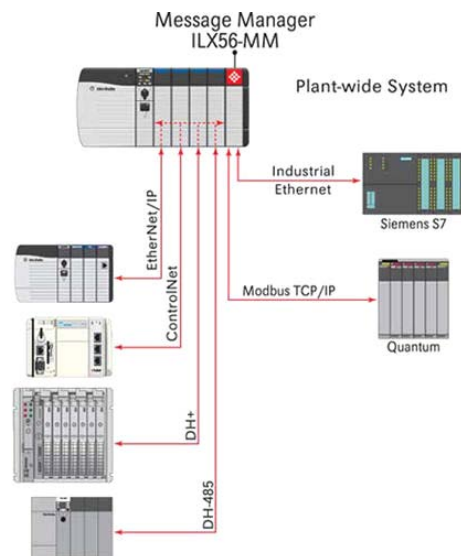
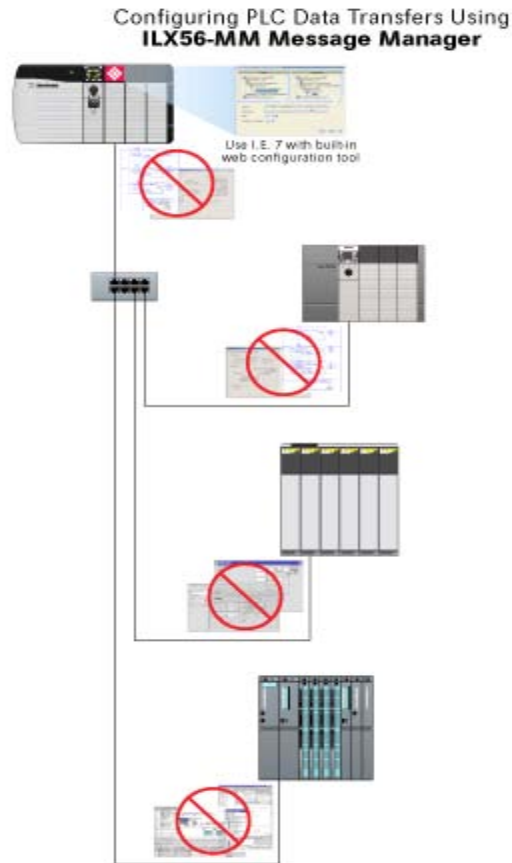
This section describes modules that connects to a Siemens Industrial Ethernet/Modbus network.

ProSoft ILX56-MM

The ProSoft Technology inRAx Message Manager is a single-slot solution that connects and transfers factory and plant process-automation monitoring and control data between islands of automation across diverse platforms, networks and protocols.

The inRAx Message Manager is an easy-to-use data transfer engine for the ControlLogix 1756 platform, supporting Rockwell Automation Programmable Logic Controller (PLC) and Programmable Automation Controller (PAC) network protocols, and platforms and devices that use Siemens Industrial Ethernet and Modbus TCP/IP protocols.

The ILX56-MM makes it easy for system integrators and end users to reduce the engineering effort required to coordinate data transfers between multiple control systems in Factory and Plant Automation applications. The Message Manager simplifies and streamlines data connection and transfer operations by enabling PLC/PAC data exchanges without the need for additional message-specific PLC programming.



Protocols Supported

Modbus TCP/IP and Siemens Industrial Ethernet supported via the built-in Ethernet ports. EtherNet/IP, ControlNet, Data Highway+ (DH+) and Data Highway 485 (DH485) are supported through Rockwell Automation (RA) 1756 communication bridge modules.

Data can be transferred between up to 16 individually-configurable communication-interface connections. A transfer list for data exchange can be created using source and destination tags.

ILX56-MM Specifications

Specification Type	Description
General	Built-in web-based configuration utility.
	Requires Internet Explorer version 7 (or later) for full implementation.
	Single slot-1756 backplane compatibility.
	Remove and Insert Under Power (RIUP).
	Local or remote rack.
	Scrolling display for status and diagnostics.

3964R Protocol

This section describes modules that connect to 3964R protocol devices.

ProSoft MVI56-S3964R

The MVI56 S3964R Protocol module from ProSoft Technology allows point-to-point communication between a ControlLogix controller (1756-L1, -L1Mx, L55, L6x) and a partner with 3964R (with or without RK512) communication capability.

The 3964R protocol (with or without RK512) was designed by Siemens for bidirectional data communication through a point-to-point connection. It is a peer-to-peer protocol with read and write access. A BCC checksum ensures transfer safety.

The MVI56-S3964R can communicate with or without RK512 devices by using two RS-232, RS-422 communication lines (ports). The ports operate independently from each other.

Ladder logic programming in the ControlLogix processor is required to enable and support the Siemens 3964R protocol functionality. The ladder program handles data encoding/decoding transferred from the module and the initiation of protocol-specific functionality in the module. Example ladder programs are provided with the module to ease the implementation of the module in the user application.

MVI56 S3964R Specifications

Specification Type	Description
Functional	Single-slot, ControlLogix backplane compatible.
	Multiple modules can be placed in a rack up to the chassis power supply limit.
	While in 3964R with RK512 mode on PRT1 and PRT2, DB-SEND and DB-FETCH instructions can be initiated from the ControlLogix processor or received by the communication partner.
	While in 3964R without RK512 mode on PRT1 and PRT2, data can be sent from the ControlLogix processor or be received from the communication partner.
	The maximum transfer rate is 512 bytes in any mode.
	Data byte swapping can be configured to adjust to a word format.
	While in 3964R with RK512 mode on PRT1 and PRT2, evaluation of the header information DB (data block), DW (data word) and coordination bytes 9 and 10 is possible.
	Communication activity and diagnostics are available through LEDs and acknowledgment telegrams.

Standalone Modules

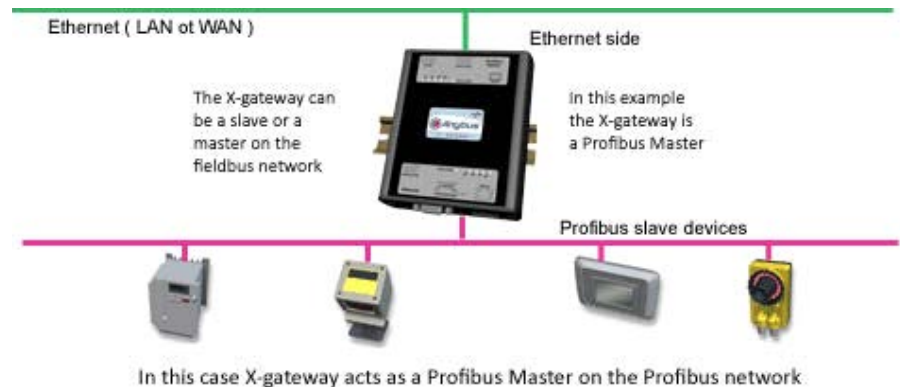
This section contains an overview of the available standalone modules that provide communication to S5 network.

HMS

HMS provides several linking devices to exchange data from one network to another. The device below can be used to pass data between the Ethernet/IP and Profibus DP network.

Anybus X-Gateway

The X-gateways are compact devices for standard DIN rail mounting and 24 Volt industrial-power supply. They are based on the proven Anybus communication modules. Embedded into a small metal housing, the modules form an intelligent link between two industrial networks.



The Anybus X-gateways provide a transparent pass for the data between the two networks. They do not change the sequence of the data nor do they change the data itself. Input data on network 1 will appear as output data on network 2. Output data on network 1 will appear as Input data on network 2. The total amount of data transferred is limited to a maximum 512 bytes Input and 512 bytes Output data. Fast I/O data transfer with average transit times between the two networks of: 10...15 ms. The X-gateway is handling the data consistently. In-built simple configuration interface accessible via Windows Hyper Terminal. IT functions such as dynamic web server, supporting downloadable customer-specific web pages. Remote monitoring and diagnostic features for most Ethernet versions based on an integrated web server. Optional control status information added to I/O data for diagnostic purposes. Included Anybus OPC server version supports OPC Data Access (DA) v1.0, 2.05, 3.0.

The X-gateways primary focus is with the transfer of cyclic I/O data between two networks. Depending on network combinations, acyclic parameter data can also be transferred (such as SDO, DPV1, PCP, or Explicit messaging).

ProSoft

ProSoft provides several linking devices to exchange data from one network to another. The device below can be used to pass data between the Ethernet/ IP and profibus DP network.

EtherNet/IP to Profibus DP Master 5204-DFNT-PDPM

The DFNT-PDPM modules are ideal for applications where EtherNet/IP connectivity can be used to integrate a Profibus DP slave device into a system. The module is a stand-alone DIN rail-mounted protocol gateway that provides one Ethernet port and one Profibus DP Slave configurable DB9F port. The EtherNet/IP (DFNT) protocol driver supports the Explicit Messaging implementation of the protocol. User-configurable as a Client and a Server, the EtherNet/IP port is a very powerful data-transfer tool. The Profibus DP Master protocol driver supports Master implementations of the protocol on a Mono-Master or Multi-Master network.



The Ethernet communication port provides a high-speed client and server connection. Data is exchanged between the protocol drivers via an internal data file. The internal database that consists of areas for application data, status information, and configuration information. The internal database is shared between all ports on the module and is used as a conduit to pass information from a device on one network to one or more devices on another network. All functions of a Profibus Class 1 Master are supported, that is, the module acts as a central controller that cyclically exchanges information with the slaves in a defined message cycle.

The Profibus Master protocol driver exists as a single-port implementation. The driver can be configured as a Class 1 Profibus Master to interface with other Profibus slave devices.

The unit is also used for configuration of the nodes on the PROFIBUS network. It provides access to standard and extended diagnostic information. The PROFIBUS master port can be used to continuously interface with PROFIBUS slave devices over a serial communication interface (RS-485).

5204-DFNT-PDPM Specifications

Specification Type	Description
Profibus DP Master	Multi-drop on a Profibus DP link with other DP compatible devices
	Software configurable parameters: <ul style="list-style-type: none"> • Baud rate: 9.6 Kbps...12 Mbps • Address: 0...125
	Data I/O: Up to 512 cyclic bytes (256 words) input and 512 bytes (256 words) output data
	Autobaud detection at all Profibus DP rates
	9-pin D-shell female Profibus DP connector
	Supports extended diagnostic data
	Can be used as active Profibus Master with Profibus
	PA slaves on the network
	Supports Sync and Freeze commands
	Supports Multicast and Broadcast telegrams
Ethernet/IP	Messaging: <ul style="list-style-type: none"> • PCCC on CIP • Explicit Messaging supported
	Five independent TCP server sockets permit remote clients to interact with all data contained in the module
	125-word read-and-write data lengths
	Floating point data supported

The ProLinx Gateway also features built-in web server functionality. All Profibus and A-B/ControlLogix processor data can be accessed by any standard browser with customer-created web pages.

ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) provides a quick and easy way to manage module configuration files customized to meet your application needs. PCB is not only a powerful solution for new configuration files, but also lets you to import information from previously installed (known-working) configurations to new projects.

In-rack Interface Cards for S5 rack

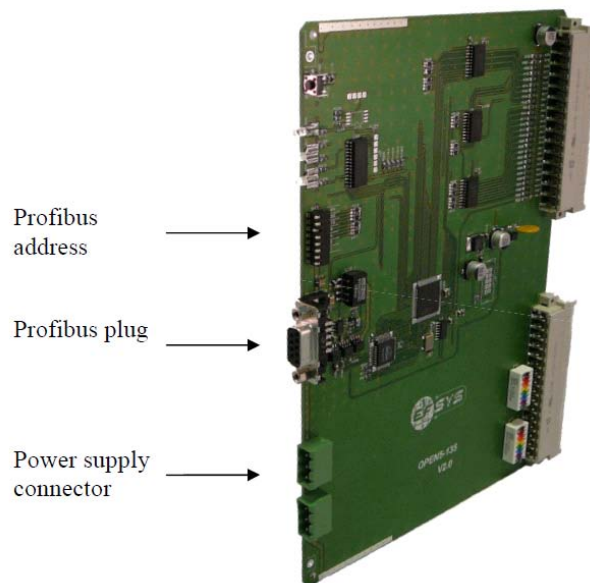
To put the remote S5 I/O on the Profibus DP network, a Profibus DP slave card has to be installed in the remote S5 rack.

If a S5 IM-308-C module is installed, the addressing needs to be configured with the Com Profibus software package. Because these cards are obsolete, third-party modules can also be used to put the S5 I/O on the Profibus DP network.

Efsys / A2i

To provide a S5 rack connectivity to a Profibus DP network, the following module can be used.

Open5-135U



This interface card can be slotted in a S5 rack to connect to S5 I/O modules not yet available on the Profibus network. The OPEN5 board supports most S5 digital input and output modules. However, intelligent I/O cards are not compatible with the OPEN5 Board.

The card supports DPV0. The words that can be read/written simultaneously depends on the number of I/O words (number and type of cards). The limit is 244 bytes per partner (DPV0 limitation) and it operates on all Profibus data rates up to 12 Mbps. A maximum total input data size of 1984 bytes and output size of 1976 bytes combined, for DP master and slave.

The response time depends on the speed of communication, the number of partners on the bus, and the number of I/O cards in the S5 rack; the minimum time is 10 ms.

One OPEN5 card is needed per 135U rack (main and extension rack). Insert the OPEN5 Board into the extension slot (extreme right-hand side-Slot 163).

Profibus Limitations

The Profibus exchange table limits a maximum of five boards of 16 analog I/O on the analog Inputs-Outputs.

An interface card for the S5 115U processor is also available.

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the web to assist you in using its products. At <http://www.rockwellautomation.com/literature>, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect Support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit <http://www.rockwellautomation.com/support>.

Installation Assistance

If you experience a problem with a hardware module within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your module up and running.

United States	1.440.646.3223 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

New Product Satisfaction Return

Rockwell tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning, it may need to be returned.

United States	Contact your distributor. You must provide a Customer Support case number (see phone number above to obtain one) for your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for return procedure.

RSLogix, RSLogix 5000, Logix5000, RSLinx, FactoryTalk View, CompactLogix, ControlLogix, GuardLogix, FlexLogix, DriveLogix, SoftLogix, Compact I/O, POINT I/O, ArmorPoint I/O, ArmorBlock MaXum, FLEX I/O, FLEX Ex, NetLinx, Integrated Architecture, Encompass, SLC 500, MicroLogix, PLC-2, PLC-3, PLC-5, SynchLink, and TechConnect are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846