CompactLogix Controllers

(Cat. No. 1769-L20 and 1769-L30)

These release notes correspond to major revision 11, minor revision 27 of the CompactLogix controller firmware. Use this firmware release with:

Software Product:	Compatible Version:
RSLogix 5000 programming software	11.11
RSLinx software	2.40

This release note provides this information:

Publication 1769-RN003K-EN-P - April 2003

CompactLogix Enhancements

- You can now change the RPI for local I/O to be any rate from 2ms to 750ms. Make this selection on the properties for the CompactBus in the Controller Organizer.

- The controller now provides additional fault information for I/O modules. This information is stored in the Fault word of the tags created for the I/O module. The Fault word now includes these bits:

Bit	Description
Fault_Bit	This bit indicates that at least one bit in the fault word is set (1). If all the bits in the fault word are cleared (0), this bit is cleared (0).
Connection_Closed	This bit indicates whether the connection to the module is open (0) or closed (1). If the connection is closed (1), the Fault_Bit it set (1).

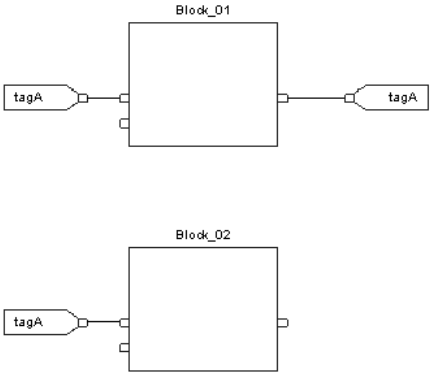
- You can use a 1769-SDN scanner in a DeviceNet configuration.

- If you connect a 1761-NET-ENI series B to the controller, you no longer need another 1761-NET-ENI module connected to your computer to be able to communicate with a CompactLogix controller over Ethernet. With the 1761-NET-ENI series B module connected to the controller, the computer can use its own Ethernet card.

Common Logix Enhancements

This revision of CompactLogix controllers contains these new features, which are common to all the Logix controllers:

Enhancement:	Description:
Sequential function chart programming language	A sequential function chart (SFC) is similar to a flowchart of your process. It defines the steps or states through which your system progresses. Use the SFC to: • organize the functional specification for your system • program and control your system as a series of steps and transitions A sequential function chart can contain these elements: • steps • transitions • actions • stops • text boxes
New instructions for use with a sequential function chart (SFC)	These new instructions are available in relay ladder or structured text: EOT set the state of a transition in an SFC to true or false SFP pause an executing SFC SFR reset the execution of an SFC to a different step or stop
Structured text programming language	Structured text is a textual programming language that uses statements to define what to execute. Structured text can contain these components: • assignments • expressions • instructions • constructs • comments You can either program structured text as a routine or embed the structured text within a sequential function chart.
Online editing of function block routines	This revision lets you edit function block routines (diagrams) while online with the controller. • Online edits include changes to logic, sheet names, pin visibility, block locations, etc. • You edit a function block routine the way you edit a ladder routine: start a pending edit, accept the edit, test the edit, and finally assemble the edit.
SIZE instruction lets you specify an array tag	The source for a SIZE instruction can now be an array tag. You no longer have to specify the first element in the array.

Enhancement:	Description:
Use the same tag in multiple IREFs and OREFs	You can use the same tag in multiple IREFs and an OREF in the same function block routine. Because the values of tags in IREFs are latched every scan through the routine, all IREFs will use the same value, even if an OREF obtains a different tag value during execution of the routine. In this example, if tagA has a value of 25.4 when the routine starts executing this scan, and Block_01 changes the value of tagA to 50.9, the second IREF wired into Block_02 will still use a value of 25.4 when Block_02 executes this scan. The new tagA value of 50.9 will not be used by any IREFs in this routine until the start of the next scan. 
REAL data type shows an extra digit of precision	The REAL data type now shows a 32-bit (4-byte) IEEE floating-point value with the following range: • $-3.40282347E^{38}$ to $-1.17549435E^{-38}$ (negative values) • 0 • $1.17549435E^{-38}$ to $3.40282347E^{38}$ (positive values) The REAL data type also stores ±infinite, ±NAN, and -IND, but the software display differs based on the display format.

Display Format:	Equivalent:	Software Display
Real	+infinite	1.$
	-infinite	-1.$
	+NAN	1.#QNAN
	-NAN	-1.#QNAN
	-indefinite	-1.#IND
Exponential	+infinite	1.#INF000e+000
	-infinite	-1.#INF000e+000
	+NAN	1.#QNAN000e+000
	-NAN	-1.#QNAN000e+000
	-indefinite	-1.#IND000e+000

The software also stores and displays the IEEE subnormal range:
- $-1.17549421E^{-38}$ to $-1.40129846E^{-45}$ (negative values)
- $1.40129846E^{-45}$ to $1.17549421E^{-38}$ (positive values)

Enhancement:	Description:	
PIDE_AUTOTUNE structure contains new status bits	When you use the PIDE Auto Tune feature, it is possible to set up a tuning environment in which the auto tune procedure successfully completes but the results are unusable. To provide an indication that this occurred, the PIDE_AUTOTUNE structure includes new members.	
	Member:	**Description:**
	PVChangeTooSmall	The PV change seen as a result of the CV step change was very small.
	StepSizeTooSmall	The CV step size that you configured for the autotuner was very small. Autotune the loop again using a larger step size.
	GainTooLarge	The autotuner identified a very large gain, which means that a small step change in CV output caused a very large change in PV.
	GainTooSmall	The autotuner identified a very small gain, which means that a step change in CV output caused only a very small change in PV.
	LongDeadTime	The autotuner identified a long deadtime, which means that it takes a long time between when the output of the loop changes and the PV starts to respond as a result of that change. This is often a result of having the sensor for your PV physically located far away from your actuator controlling the process. The autotuner will suggest a set of tuning constants, but standard PID control may have a difficult time controlling this process effectively.
PLC-5 typed read message errors if destination is too small	In a Message (MSG) instruction that is configured for *PLC5 Typed Read*, the instruction no longer executes if the Destination is too small for the Source data. If this occurs, the instruction sets the ER bit. If a MSG instruction is configured for *PLC5 Typed Read* and the data type of the Source does not match the data type of the Destination, the instruction converts the Source to the data type of the Destination. For example, if the data type of the Source is INTs and the data type of the Destination is DINTs, the instruction converts the INTs to DINTs. In this example, the Destination requires one DINT element for each INT of the Source data. In previous revisions, if a data conversion occurred but the Destination was too small, data beyond the Destination was overwritten. This may have caused the controller to fail during a download or online edit operation.	

Corrected Anomalies

This revision of the controller corrects the following anomalies:

<table>
<tr><th>Enhancement:</th><th>Description:</th></tr>
<tr><td>Product Service Advisory
Power disruptions cleared memory</td><td>If power to the controller turned on and then turned off again in less than a second, the controller might have cleared the project from its memory.
If the controller did not have enough time to complete a critical portion of the power-up sequence (less then 1 second), the controller typically cleared its memory.
This might have occurred during brownouts or other situations were power to the controller fluctuated for a short duration.</td></tr>
<tr><td>Load from nonvolatile memory produced faults</td><td>If a project automatically loaded from the nonvolatile memory of a controller, a fault might have occured.<table><tr><th>If the project:</th><th>Then:</th></tr><tr><td>contained motion axes</td><td>A fault was more likely to occur. The following faults might have occurred:
non-recoverable fault (solid red OK LED). This caused the controller to clear the project from its memory.
motion group fault. The controller failed to become the CST master. This caused the motion group to fault because there was no CST master in the chassis.</td></tr><tr><td>did not contain motion axes</td><td>A fault was still possible, though less likely. The following fault might have occurred:
non-recoverable fault (solid red OK LED). This caused the controller to clear the project from its memory.</td></tr></table>A project automatically loads from nonvolatile memory only if you configure it to do so. You can configure a project to automatically load under one of the following circumstances:
during power-up
when the memory of the controller is empty</td></tr>
<tr><td>Deleting a tag online caused slow communications or controller failure</td><td>If you deleted a tag while online with the controller either of the following might have occurred:
communications would slow down
the controller would fail (solid red OK LED)</td></tr>
<tr><td>Controller occasionally failed a parity check</td><td>This revision lets the controller recover from some parity errors without user intervention.</td></tr>
</table>

Enhancement:	Description:
Automatic reset of an SFC produced erroneous instruction execution	If you chose the *Automatic Reset* option for your SFCs, some instructions would execute incorrectly when executed directly or indirectly by an SFC: • JSR/RET instruction pair produced a major fault. • FFL, FFU, LFL, LFU instructions caused the controller to fail (solid red OK LED).
SQO instruction failed to execute a prescan	If you chose the *Restart at most recently executed step* option for your SFCs, an SQO instruction would not execute a prescan when executed directly or indirectly by an SFC.
BTD instruction caused controller failure	If an SFC directly or indirectly executed a BTD instruction and the Length was greater than 31, the controller would fail (solid red OK LED).
SFC alarming failed to stay enabled or disabled	In a step of an SFC, the *AlarmEnable* setting might not stay at the state you set it if you change it while online. For example, if you checked the *AlarmEnable* check box while online, it might revert to the cleared state later in the execution of your project.
SCRV instruction caused controller failure	Under certain combinations of input parameters, an S-Curve (SCRV) instruction might have divided a value by zero. This would have caused the controller to fail (solid red OK LED).

Restrictions

This revision of CompactLogix controllers contains this restriction, which is common to all the Logix controllers:

Restriction:	Description:
LDL2 instruction produces inaccurate coefficients or controller failure	A Second-Order Lead Lag (LDL2) instruction may produce the following: • inaccurate internal coefficients • controller failure (solid red OK LED) This may occur under the following *combination* of circumstances: • You initially set the WLead, WLag, or ZetaLag input parameters = 0: • While the controller is in Run mode, you change any of the WLead, WLag, ZetaLead, ZetaLag, or Order parameters. Instead of setting the input parameters = 0, set the input parameters as follows:

If:	Then set the input parameters as follows:
Order = 1	WLead = 0.0000001/DeltaTime WLead = 0.0000001/DeltaTime ZetaLag = 0.05
Order = 2	WLead = 0.00001/DeltaTime WLead = 0.00001/DeltaTime ZetaLag = 0.05

Verify I/O Layout by Adding Words of Backplane Memory Used

Each module in a CompactLogix system uses a set amount of backplane memory, in addition to the data that the module stores or transfers. Some modules require a considerable amount of backplane memory. Take this into account when designing your system because it affects how many modules a controller can support. Each CompactLogix controller supports 256, 16-bit words of backplane data. This table shows how many backplane words each module uses:

Catalog number:	Number of modules:	Number of words used:	Total number:
1769-IA8I		8	
1769-IA16		8	
1769-IM12		8	
1769-IQ16		8	
1769-IQ6XOW4		12	
1769-OA8		12	
1769-OA16		12	
1769-OB16		12	
1769-OB16P		12	
1769-OV16		12	
1769-OW8		12	
1769-OW8I		12	
1769-OW16		12	
1769-IF4		14	
1769-OF2		14	
1769-IF4XOF2		20	
1769-IR6		14	
1769-IT6		16	
1769-HSC		187 (35 words input, 34 words output, 118 words configuration)	
1769-SDN		76 plus total words in scanlist	
system overhead (per controller)		34	34
		Total Words Required:(1)	

(1) The total words required cannot exceed 256 words.

For more information about the I/O memory requirements and the scanlist for a 1769-SDN module, see the *Compact I/O 1769-SDN DeviceNet Scanner Module User Manual*, publication 1769-UM009.

Additional Memory Requirements

This revision *may* require more memory than previous revisions. To estimate the additional memory that your project *may* require, use the following table:

If you have this firmware revision (add *all* that apply):	Then add the following memory requirements to your project:				Which comes from this type of memory:	
	Component			Increase per instance	I/O (base)	Expansion
10.x or earlier	programs			12 bytes		
	routines			16 bytes		
9.x or earlier	tag that uses the MESSAGE data type			376 bytes		
7.x or earlier	project			1050 bytes		
	tags			0.55 bytes		
	messages that: • transfer more than 500 bytes of data *and* • target a controller in the same chassis This memory is allocated only when the MSG instruction is enabled. To estimate, count the number of these messages that are enabled and/or cached at one time.			2000 bytes		
6.x or earlier	base tags			24 bytes		
	alias tags			16 bytes		
	produced and consumed tags	Data type	Bytes per tag			
		DINT	4	12 bytes		
		REAL	4	12 bytes		
				3 x bytes per tag		
				3 x bytes per tag		
				3 x bytes per tag		
				3 x bytes per tag		
6.x	routines			68 bytes		
5.x or earlier	routines			116 bytes		

Hold Last State and User-Defined Safe State Not Supported

The CompactLogix5320 and CompactLogix5330 controllers do not support Hold Last State or User-Defined Safe State features.

- If an I/O module fails such that its communication to the controller is lost, or if any module is disconnected from the system bus while under power, the controller will go into the fault mode. All outputs turn off when the system bus or any module faults.

- RSLogix 5000 software creates tags for modules when you add them to the I/O configuration. These 1769 module tags define configuration (C) data type members which may include attributes for alternate outputs. CompactLogix does not enable the module to use the alternate outputs. Do not configure the attributes listed below:

For digital output modules:	For analog output modules:
• ProgToFaultEn • ProgMode • ProgValue • FaultMode • FaultValue	• CH*x*ProgToFaultEn • CH*x*ProgMode • CH*x*FaultMode where CHx = the channel number

DH-485 Communications

If the controller is connected to a DH-485 network, disconnect it from the DH-485 network *before* you update the firmware of the controller. If you update the firmware of a controller while it is connected to a DH-485 network, communication on the network may stop.

We recommend that you use DH-485 communications as follows:

- If you update the firmware of a controller while it is connected to a DH-485 network, communication on the network may stop. To prevent this, disconnect the controller from the DH-485 network *before* you update the firmware of the controller.
- Logix5000 controllers should be used on DH-485 networks only when you wish to add these controllers to an existing DH-485 network. For new applications with Logix5000 controllers, DeviceNet, Ethernet, and ControlNet are the recommended networks.
- While your system is running, use a DH-485 network to send messages between devices (e.g., controllers, PanelView terminals).
- To use RSLogix 5000 software over a DH-485 network, (upload, download, monitor, edit online), place all controllers in the program mode. Excessive traffic on the DH-485 network may make it impractical to connect and use RSLogix 5000 programming software while your system is running.

Notes:

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733

Publication 1769-RN003K-EN-P - April 2003 PN 957782-48

Supersedes Publication 1769-RN003J-EN-P - December 2002