



DriveLogix Controller

These release notes should be used with major revision 11, minor revision 16 of the DriveLogix Controller firmware. Use this firmware with:

Update this:	To this revision or later:
RSLinx™ software	2.31
RSLogix 5000™ software	11.11
RSNetWorx™ for ControlNet software	3.23
RSNetWorx for DeviceNet software	3.21

IMPORTANT

To use non-volatile storage on a DriveLogix5720 controller that uses the expanded memory option, you must use DriveLogix controller firmware revision 11.15 or higher.

What Is In These Release Notes

These release notes provide the following information:

For information about:	See this section:	On this page:
preliminary actions to take before you use this revision	Before You Update Your System	2
new features for DriveLogix controllers	DriveLogix Enhancements	2
new features for all Logix controllers	Common Enhancements	3
changes to existing features in DriveLogix controllers	DriveLogix Changes	4
changes to existing features in all Logix controllers	Common Changes	4
restrictions that no longer apply to DriveLogix controllers	Common Corrected Anomalies	7
restrictions that apply to DriveLogix controllers	DriveLogix Restrictions	8
restrictions on the use of DH-485	DH-485 Communications Recommendations	9
additional memory required to update to this revision	Additional Memory Requirements	10

Before You Update Your System

Before you update your controller or RSLogix 5000 software to this revision, do the following preliminary actions:

If:	Then:
Your controller is connected to a DH-485 network.	Disconnect it from the DH-485 network <i>before</i> 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.
Your controller is close to its limits of memory.	<p>This revision <i>may</i> require more memory than previous revisions. Before you upgrade to this revision, do the following:</p> <ol style="list-style-type: none"> 1. Check the amount of unused memory that you have in the controller. To determine your unused memory, see either of the following documents: <ul style="list-style-type: none"> • Knowledgebase document 13964. To access Rockwell Automation's Knowledgebase, go to www.ab.com. Select <i>Support</i>. • <i>Logix5000 Controllers Common Procedures</i>, publication 1756-PM001E or later 2. If your controller is close to its limits of memory, see "Additional Memory Requirements" on page 10 to determine how much additional memory you require. <p>To upgrade to this revision, you may have to add an expansion memory card to the controller or use a larger memory card.</p>

DriveLogix Enhancements

This revision of DriveLogix controllers contains these new features:

- the Requested Packet Interval (RPI) of the PowerFlex 700S Drive may be set as low as 3.0 ms. This is the minimum allowed by RSLogix 5000 programming software. The default value in the programming software is 4.0 ms.
- support for EtherNet/IP™ connectivity using the 1788-ENBT EtherNet/IP communication daughtercard
- support for Non-Volatile Memory on the DriveLogix controller. Previously this feature was not available. This gives DriveLogix the ability to restore the controller's last saved project without a battery.
- Flex I/O very high speed counter (1794-VHSC) is supported on the Local DIN rail
- support for a Memory Expansion option. This option provides additional SRAM and Flash, Non-Volatile memory.
- 256K bytes of user available memory without memory option
- 768K bytes of user available and flash memory with memory option

- support for DeviceNet connectivity using the 1788-DNBO Devicenet communication daughtercard
- support for 200V, 400V, and 600V PowerFlex 700S and 700Se drive interfaces
- Specific software support has been added to support the 1794-IB32 and OB32 Flex I/O modules.

Common Enhancements

This revision of DriveLogix controllers contains the following new features:

Enhancement:	Description:								
Sequential Function Chart Programming Language	<p>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:</p> <ul style="list-style-type: none"> • organize the functional specification for your system • program and control your system as a series of steps and transitions <p>A sequential function chart can contain these elements:</p> <ul style="list-style-type: none"> • steps • transitions • actions • stops • text boxes 								
New Instructions For Use with a Sequential Function Chart (SFC)	<table border="1"> <thead> <tr> <th data-bbox="586 1192 812 1241">This instruction:</th> <th data-bbox="812 1192 1464 1241">Lets you:</th> </tr> </thead> <tbody> <tr> <td data-bbox="586 1241 812 1283">EOT</td> <td data-bbox="812 1241 1464 1283">Set the state of a transition in an SFC to true or false</td> </tr> <tr> <td data-bbox="586 1283 812 1325">SFP</td> <td data-bbox="812 1283 1464 1325">Pause an executing SFC</td> </tr> <tr> <td data-bbox="586 1325 812 1371">SFR</td> <td data-bbox="812 1325 1464 1371">Reset the execution of an SFC to a different step or stop</td> </tr> </tbody> </table>	This instruction:	Lets you:	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
	This instruction:	Lets you:							
	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	<p>Structured text is a textual programming language that uses statements to define what to execute. Structured text can contain these components:</p> <ul style="list-style-type: none"> • assignments • expressions • instructions • constructs • comments <p>You can either program structured text as a routine or embed the structured text within a sequential function chart</p>								

Enhancement:	Description:
Online Editing of Function Block Routines	<p>This revision lets you edit function block routines (diagrams) while online with the controller.</p> <ul style="list-style-type: none"> • 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.

DriveLogix Changes

This revision of DriveLogix controllers contains the following changes:

Change:	Description:
700S no longer automatically placed in the I/O configuration	The PowerFlex 700S is no longer automatically placed in the I/O configuration of the DriveLogix controller. You must add the PowerFlex 700S drive to the configuration, in a manner similar to adding an I/O module. The Controller Organizer automatically places the drive in slot two.
Use non-volatile storage	To use non-volatile storage on a DriveLogix5720 controller that uses the expanded memory option, you must use DriveLogix controller firmware revision 11.15 or higher.

Common Changes

This revision of DriveLogix controllers contains the following changes:

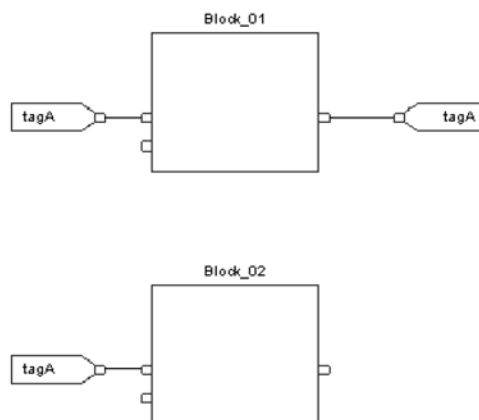
Change:	Description:
SIZE Instruction Lets You Specify an Array Tag	The source for a SIZE instruction can now be an array tag. You no longer have to specify the first element in the array.
PLC5 Typed Read Message Errors If Destination Is Too Small	<p>In a Message (MSG) instruction that is configured for <i>PLC5 Typed Read</i>, the instruction no longer executes if the Destination is too small for the Source data. If this occurs, the instruction sets the ER bit.</p> <p>If a MSG instruction is configured for <i>PLC5 Typed Read</i> 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.</p> <p>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.</p>

Change:	Description:																									
REAL Data Type Shows an Extra Digit of Precision	<p>The REAL data type now shows a 32-bit (4-byte) IEEE floating-point value with the following range:</p> <ul style="list-style-type: none"> • $-3.40282347E^{38}$ to $-1.17549435E^{38}$ (negative values) • 0 • $1.17549435E^{38}$ to $3.40282347E^{38}$ (positive values) <p>The REAL data type also stores \pminfinity, \pmNAN, and -IND, but the software display differs based on the display format.</p> <table border="1"> <thead> <tr> <th>Display Format:</th> <th>Equivalent:</th> <th>Software Display:</th> </tr> </thead> <tbody> <tr> <td rowspan="5">Real</td> <td>+infinite</td> <td>1.\$</td> </tr> <tr> <td>- infinite</td> <td>-1.\$</td> </tr> <tr> <td>+NAN</td> <td>1.#QNAN</td> </tr> <tr> <td>-NAN</td> <td>-1.#QNAN</td> </tr> <tr> <td>-indefinite</td> <td>-1.#IND</td> </tr> <tr> <td rowspan="5">Exponential</td> <td>+infinite</td> <td>1.#INF000e+000</td> </tr> <tr> <td>- infinite</td> <td>-1.#INF000e+000</td> </tr> <tr> <td>+NAN</td> <td>1.#QNAN00e+000</td> </tr> <tr> <td>-NAN</td> <td>-1.#QNAN00e+000</td> </tr> <tr> <td>-indefinite</td> <td>-1.#IND0000e+000</td> </tr> </tbody> </table> <p>The software also stores and displays the IEEE subnormal range:</p> <ul style="list-style-type: none"> • $-1.17549421E^{-38}$ to $-1.40129846E^{-45}$ (negative values) • $1.40129846E^{-45}$ to $1.17549421E^{-38}$ (positive values) 	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.#QNAN00e+000	-NAN	-1.#QNAN00e+000	-indefinite	-1.#IND0000e+000
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.#QNAN00e+000																								
	-NAN	-1.#QNAN00e+000																								
	-indefinite	-1.#IND0000e+000																								

Change:	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. You still have the option of accepting the auto tune values.	

To see if:	Examine this member of the PIDE_AUTOTUNE structure:	Explanation:												
Observed PV change was too small	PVChangeTooSmall	<p>The PV change seen as a result of the CV step change was very small.</p> <ul style="list-style-type: none"> Filter your PV to eliminate excessive noise, which could have caused the autotuner to mistake a noise spike for an actual PV response. Make sure that the PIDE instruction is executing at an appropriate rate for your process. For example, if your process is a slow temperature loop, run your PIDE instruction in a slow (0.5 sec to 2 sec) periodic task. An execution rate that is too fast can cause the autotuner to mistake a noise spike right after the autotuner starts for an actual PV response. 												
Step size is too small	StepSizeTooSmall	The CV step size that you configured for the autotuner was very small. You might get better results if you autotune the loop again using a larger step size.												
Process gain is too large	GainTooLarge	The autotuner identified your process as having a very large process gain. In other words, a small step change in CV output caused a very large change in PV. Make sure that your control actuator is properly sized for this application.												
Process gain is too small	GainTooSmall	<p>The autotuner identified your process as having a very small process gain. In other words, a step change in CV output caused only a very small change in PV. To get better results:</p> <ul style="list-style-type: none"> Filter your PV to eliminate excessive noise, which could have caused the autotuner to mistake a noise spike for an actual PV response. Make sure that the PIDE instruction is executing at an appropriate rate for your process. For example, if your process is a slow temperature loop, run your PIDE instruction in a slow (0.5 sec to 2 sec) periodic task. An execution rate that is too fast can cause the autotuner to mistake a noise spike right after the autotuner starts for an actual PV response. Make sure that your control actuator is properly sized for this application. 												
Dead time is	LongDeadTime	<p>The autotuner identified your process as having a long deadtime. In other words, it</p> <p>You can also examine the bits of the AtuneStatus member for the same information:</p> <table border="1"> <thead> <tr> <th>For this member:</th> <th>Examine this bit of the AtuneStatus member:</th> </tr> </thead> <tbody> <tr> <td>PVChangeTooSmall</td> <td>27</td> </tr> <tr> <td>StepSizeTooSmall</td> <td>28</td> </tr> <tr> <td>GainTooLarge</td> <td>29</td> </tr> <tr> <td>GainTooSmall</td> <td>30</td> </tr> <tr> <td>LongDeadTime</td> <td>31</td> </tr> </tbody> </table>	For this member:	Examine this bit of the AtuneStatus member:	PVChangeTooSmall	27	StepSizeTooSmall	28	GainTooLarge	29	GainTooSmall	30	LongDeadTime	31
For this member:	Examine this bit of the AtuneStatus member:													
PVChangeTooSmall	27													
StepSizeTooSmall	28													
GainTooLarge	29													
GainTooSmall	30													
LongDeadTime	31													

Change:	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 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.



700S no longer automatically placed in the I/O configuration	The PowerFlex 700S is no longer automatically placed in the I/O configuration of the DriveLogix controller. You must add the PowerFlex 700S drive to the configuration, in a manner similar to adding an I/O module. The Controller Organizer automatically places the drive in slot two.
Use non-volatile storage	To use non-volatile storage on a DriveLogix5720 controller that uses the expanded memory option, you must use DriveLogix controller firmware revision 11.15 or higher.

Common Corrected Anomalies

This revision of DriveLogix controllers corrects the following anomalies:

Corrected anomaly:	Description:
Online Edit of Tags Might Have Caused Communication Failure	If you deleted an unused tag while online, you might have lost communication with the controller. RSLinx showed a Red X over the controller and you were unable to communicate with the controller through either the serial port or another communication module. The communication failure could have occurred immediately after you deleted the tag or later on in the execution of the project. A power cycle would temporarily clear the problem.
Size of the ASCII Buffer <i>No Longer</i> Limited to 255 Characters	You can set the size of the ASCII buffer of the serial port to any number of characters up to 65,536 characters. In previous revisions, a setting larger than 255 characters caused ABL instructions to miss the termination character and set status bits to erroneous values.
Failure to Write to Tags Via OPC/DDE	In the R11 firmware revisions prior to this revision, you could not write to a tag that was an alias for the tag of an I/O module. Neither tag would accept the write but the controller would not report a failure. For example, the failure to write occurred when another software application attempted to write to an alias tag via OPC/DDE communications.

Corrected anomaly:	Description:
Deleting a Tag Online Caused Slow Communications or Controller Failure	<p>If you deleted a tag while online with the controller either of the following might have occurred:</p> <ul style="list-style-type: none"> • communications would slow down • the controller would fail (solid red OK LED)
Controller Occasionally Failed a Parity Check	This revision lets the controller recover from some parity errors without user intervention.
Automatic Reset of an SFC Produced Erroneous Instruction Execution	<p>If you chose the <i>Automatic Reset</i> option for your SFCs, some instructions would execute incorrectly when executed directly or indirectly by an SFC:</p> <ul style="list-style-type: none"> • 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 <i>Restart at most recently executed step</i> 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 <i>AlarmEnable</i> setting might not stay at the state you set it if you change it while online. For example, if you checked the <i>AlarmEnable</i> 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).

DriveLogix Restrictions

This firmware version has these restrictions:

- Forcing is not supported between the PowerFlex 700S and DriveLogix. The forcing values can be set for the controller inputs and outputs. However, these values will not be used by the Logix Program nor will they be transmitted to the PowerFlex 700S.
- The minimum recommended Requested Packet Interval (RPI) setting for the local Flex I/O rail is 30 ms.

DH-485 Communications Recommendations

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.
- Place a DriveLogix controller on a DH-485 network only when you need to add the controller to an existing system. For new systems, use a ControlNet network.
- 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 while online), place all controllers in the program mode. Excessive traffic may make it impractical to use RSLogix 5000 software over this network while your system is running.

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 <i>all</i> 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		Yes	
	routines	16 bytes		Yes	
9.x or earlier	tag that uses the MESSAGE data type	376 bytes		Yes	
8.x or 9.x	produced or consumed axis	(-21.6K bytes)	Yes		
	axis that <i>is not</i> produced or consumed	(-21.6K bytes)		Yes	
8.x or earlier	output cam execution targets	5,404 bytes		Yes	
	motion group	32 bytes		Yes	
7.x or earlier	project	1050 bytes	Yes		
	tags	0.55 bytes		Yes	
	messages that: <ul style="list-style-type: none"> transfer more than 500 bytes of data <i>and</i> 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	Yes		
6.x or earlier	base tags	24 bytes		Yes	
	alias tags	16 bytes		Yes	
	produced and consumed tags	Data type	Bytes per tag		
		DINT	4	12 bytes	Yes
		REAL	4	12 bytes	Yes
				3 x bytes per tag	Yes
				3 x bytes per tag	Yes
		3 x bytes per tag	Yes		
		3 x bytes per tag	Yes		
6.x	routines	68 bytes		Yes	
5.x or earlier	routines	116 bytes		Yes	

For additional information on how the controller organizes its memory, see Knowledgebase document 13964. To access Rockwell Automation's Knowledgebase, go to www.ab.com. Select *Support*.

Notes:

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