

Application Note #1190: Native Tag Drivers - Tips to Optimize Screen Performance

Introduction

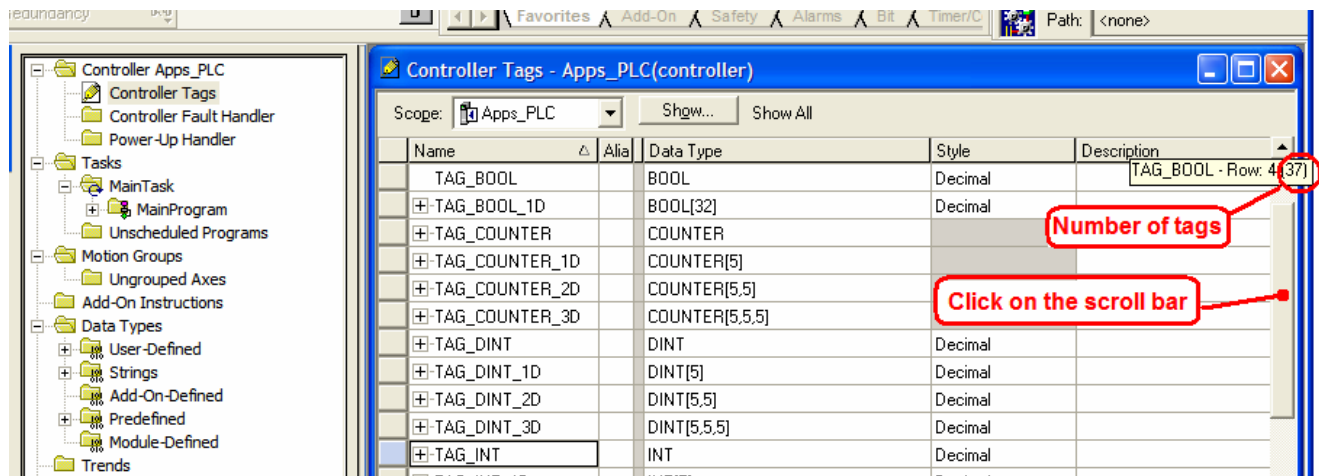
This document is intended for the users of GP-Pro EX Rockwell Native Tag Drivers. Much of this information also applies to other PLC/device drivers. HMI performance optimization begins with the initial design of the PLC project. This is especially true when the PLC uses a native tag protocol. This application note begins with a section for the PLC programmer. It then continues with a section for the GP-Pro EX programmer, including screen design tips.

Projects connected to Allen Bradley Control Logix PLCs using native tags are often very large and complex. For the most part the same techniques are applied as with any PLC. The native tag protocol uses un-located variables; however, and that presents some special challenges. Tips to meet those challenges are included in the guide.

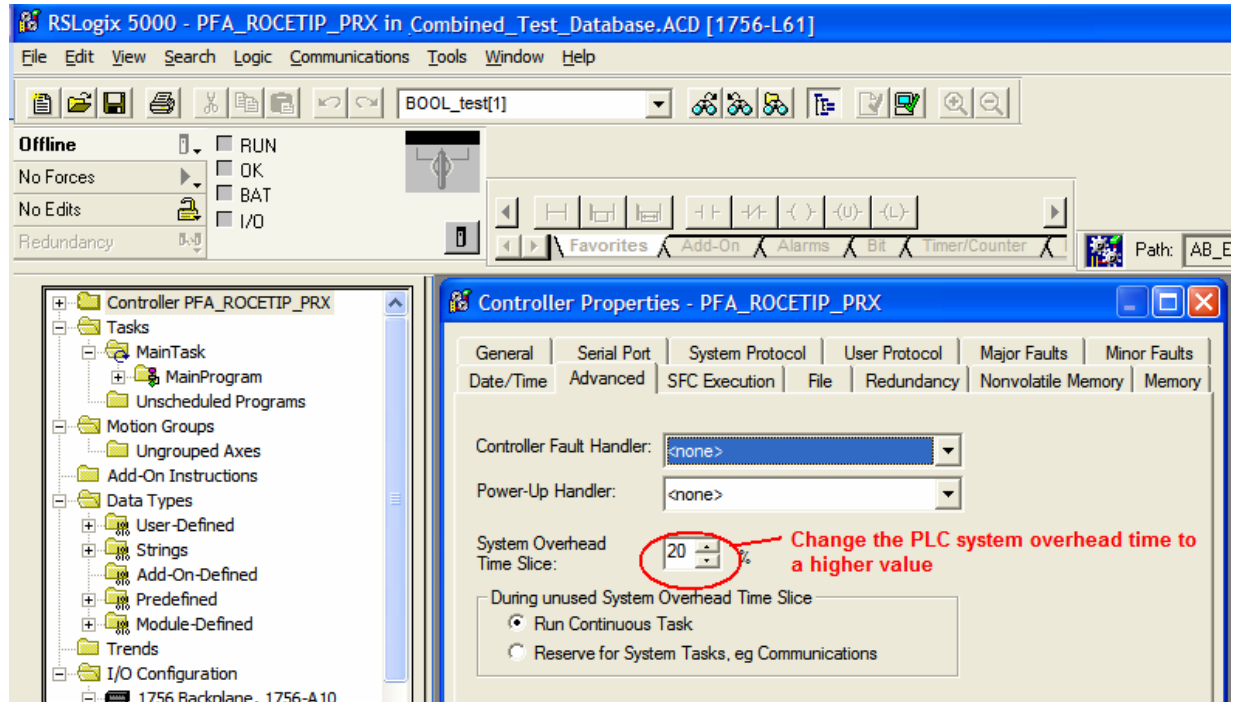
Be sure you have the GP-Pro EX Device/PLC driver Rockwell EtherNet/IP v1.13.11 or later. If you have an earlier version, obtain an update.

Developing the PLC Project:

1. Create an INT array of 20 elements for the HMI system area and watchdog timer. The array can be used by the PLC to monitor and control the HMI screens as well as synchronizing the clocks.
2. Try to keep the total number of tags in the PLC project below 4,000. The time to initial display of data on startup is slowed by large numbers of tags in the PLC regardless of how few are actually in the GP-Pro EX Tag Data database. Click on the Tag window scroll bar in RS-Logix to find the number of tags. Do this for controller scope tags and program scope tags. If the combined number of all tags exceeds 4000, reduce the number of tags by combining them into arrays and user defined structures.



3. Share PLC data with the HMI in arrays when possible. A native tag array is accessed by the HMI as a contiguous block of data. Combining alarm variables into BOOL or DINT arrays improves the overall performance of the entire HMI project. Screens that have large numbers of variables perform better when the variables are combined in arrays. Best performance is achieved when the number of discontinuous device tags on a screen plus alarms is less than 40 items (current screen + alarms). An array of variables counts as single item. In the case of an array consisting of structures each member counts as one item. Example: [PLC1]CartStatus.CartNo[2].Disabled) In the case where an array is a member of a Structure the array member counts as a single item. Example: [PLC1]CartStatus.Disabled.CartNo[2].
4. You can improve HMI performance by increasing the PLC "System Overhead Time Slice". In RS Logix go to the Controller Properties dialog box > Advanced tab to increase the System Overhead Time Slice. Be aware this change might also have an adverse performance impact on the control program being executed by the ControlLogix controller.



Developing the HMI Project:

1. Use an array of 20 INT variables for the System Area.
2. Words LS0000 – LS0019 or Bit memory LS000000 – LS001915 are designated as System Area. Refer to appendix A1.4 of the Reference Manual for information before addressing LS memory. Consider using USR memory instead.
3. Read Area Size is not a communications parameter. It should be 0 unless the Read Area feature is used to exchange data with the PLC.

4. The Watchdog Timer Setting should be 0 unless the PLC project is using this feature to monitor the HMI. Always use an INT variable for the Watchdog Write Address. For best performance choose an unused element in the same INT array as the System Area.
5. Use the header and footer feature rather than window screens when practical. A window screen may not appear until PLC data is displayed on the main screen.
6. Eliminate any unused global or extended scripts.
7. Only use the "Continuous Action" trigger for scripts that need to update every screen refresh. Trigger scripts only as often as needed. If a script needs to run multiple times per second consider using the internal HMI display scan counter register LS2038 for a trigger. Triggering a script with "Bit On" LS203803 will cause the script to run more than once a second. The script can be triggered twice as often by selecting the next lower bit (i.e. changing the trigger from LS203803 to LS203802).
8. Organize screen objects so static objects are drawn while waiting for data needed to draw dynamic objects. This also reduces the need to refresh static objects during screen updates. To do this move unanimated static objects such as bit map graphics, text, lines, and boxes to the "base layer" or background. Refer to the Screen Data List. Typically Draw/Parts objects above (behind) the first animation on the list are on the "base layer". See related APNT1175 for more information. <https://www.hmisource.com/otasuke/files/appnotes/APNT1175.pdf>

Draw/Parts	Information	Animation
DD_0000	[PLC1]TAG_DINT	
SL_0000	[PLC1]TAG_DINT.[PLC1]TAG_DINT	
SL_0001	[PLC1]TAG_DINT.[PLC1]TAG_DINT	
GR_0000	[PLC1]TAG_DINT	
Filled Polygo	(231, 137, 321, 199)	
Rectangle	(327, 62, 439, 163)	
Circle/Oval	(362, 186, 440, 210).[PLC1]TAG_DINT.00,	
Text	(201, 222, 456, 269)	
Rectangle	(231, 284, 316, 368),-----	
Circle/Oval	(363, 275, 440, 325)	

9. On screens with many dynamic objects choose screen parts rather than animation. Especially if the resulting screen appearance and functionality will be the same. For example, an indicator using Standard Parts object SW_SP221_OFF and SW_SP221_ON could be used to replace multi-state color/visibility animation of a rectangle or screen text.

For technical support email Pro-face America at support@profacamerica.com or call 734-944-0482.

© 2009 Pro-face America. Specifications may change without notice. Pro-face is a registered trademark of Digital Corporation. Other brand or product names are the property of their respective owners.