



## **XL200 Series PLC Interface**

**Copyright © 2023 – AMS Controls, Inc. All rights reserved.**

The information contained in this document is proprietary to AMS Controls, Inc., and is not to be reproduced in any form without the specific written authorization of AMS Controls, Inc.

AMS Controls, Inc. reserves the right to change the contents of this manual without customer notice or permission. The procedures described in this handbook must be performed only by trained, certified industrial electricians in accordance with local, state, federal, and NEC codes and regulations. While AMS Controls has taken all reasonable care to develop safe procedures, it accepts no responsibility for injuries, damages, death, or destruction to persons or equipment caused as a result of not closely following these instructions. For specific controller questions, contact AMS Controls Inc.

AMS Controls, Inc.  
12180 Prichard Farm Road  
Maryland Heights, MO 63043

314.344.3144 - [www.amscontrols.com](http://www.amscontrols.com)

---

# Table of Contents

Copyright © 2023 – AMS Controls, Inc. All rights reserved. .... ii

Table of Contents ..... i

Introduction ..... 4

Overview ..... 4

MODBUS Card Requirements ..... 4

Applications..... 4

    Standard Input/Output Mapping ..... 5

    External Gag Output Mapping ..... 5

    Y-Axis and Machine Setup Motion Driver ..... 5

    High Level Data Publishing ..... 5

    E-Stop Reasons and Other PLC Diagnostics Information ..... 5

    Delay and Scrap Code Publishing ..... 5

    Jog SERCOS PC Axes ..... 5

    PLC Multi-Axis Machine Setup and Tool Positioning ..... 5

    PLC Die Accelerator and Material Remote Control ..... 6

    PLC Remote Tool Select ..... 6

    Stick Fed Auxiliary Weld Detect Sensors ..... 6

    PLC Remote Setup Configuration Select ..... 6

    PLC Remote Run Velocity Pot ..... 6

    PLC Remote Press Speed Communication ..... 6

    PLC Closed Loop Material Feed Status ..... 6

    PLC Force Single Encoder Operation on CL Feeder ..... 6

    Second Line Encoder Selection ..... 6

    PLC Positioning of Multi-Axis Axes ..... 7

    Increase Quantity ..... 7

    Item Data Request ..... 7

    Display and Edit PLC Parameters in XL Setup Screen ..... 7

    PLC Remote Tool Configuration Select ..... 7

    Product Code Comparison Bypass ..... 7

    Feed to Stop Shear Handshake ..... 7

    XL200 Line Speed ..... 8

    Initiate Quality Audit (Custom List) Triggers ..... 8

Hardware Connections..... 8

XL200 Software Configuration..... 8

*Model Option (Bit Code)* ..... 8

*Setup Parameters* ..... 8

XL200 Controller Operation ..... 9

PLC Configuration Data Structure..... 11

PLC Coil Addressing..... 13

PLC Configuration Field Bits ..... 15

PLC Function Coils..... 17

PLC Configuration Bits and User Data Structures..... 19

    Version Level 100 ..... 19

*Map XL200 Standard Outputs to PLC* ..... 19

*Map XL200 Standard Inputs to PLC* ..... 19

*Strobe Watchdog Coil* ..... 20

*Report PLC Error Message*..... 20

*Report PLC Diagnostics Message* ..... 20

<i>Map XL200 External Gag Outputs to PLC</i> .....	21
Version Level 101 .....	22
<i>Receive Current Item Information</i> .....	22
<i>Receive Next Item Information</i> .....	24
<i>Receive Current Order Number</i> .....	24
<i>Receive Current Material Code</i> .....	24
<i>Receive Current Product Code</i> .....	25
<i>Receive Order User 1 Field</i> .....	25
<i>Receive Order User 2 Field</i> .....	25
<i>Receive Order User 3 Field</i> .....	26
<i>Receive Order User 4 Field</i> .....	26
<i>Receive Order User 5 Field</i> .....	26
<i>Receive Order User 6 Field</i> .....	27
<i>Receive Order User 7 Field</i> .....	27
<i>Receive Order User 8 Field</i> .....	27
<i>Receive Item User 1 Field</i> .....	27
<i>Receive Item User 2 Field</i> .....	28
<i>Report Delay Code</i> .....	28
<i>Report Scrap Code</i> .....	28
Version Level 102 .....	30
<i>Jog Axis Function</i> .....	30
Version Level 103 .....	31
<i>PLC Notification Event</i> .....	31
Version Level 104 .....	32
<i>PLC_Y-Axis and Machine Setup</i> .....	32
Version Level 105 .....	35
<i>Multi-Axis Material Change Event</i> .....	35
<i>Multi-Axis Product Code Change Event</i> .....	35
Version Level 106 .....	37
<i>Receive Current Item Information</i> .....	37
<i>Receive Next Item Information</i> .....	37
Version Level 107 .....	38
Version Level 108 .....	38
<i>Map XL200 Standard Outputs to PLC</i> .....	38
<i>Map XL200 Standard Inputs to PLC</i> .....	38
Version Level 109 .....	40
<i>PLC_Remote Control Functions</i> .....	40
Version Level 110 .....	42
<i>PLC_Remote Tool Select</i> .....	42
Version Level 111 .....	43
<i>Stick Fed Auxiliary Weld Detect Sensors</i> .....	43
Version Level 112 .....	44
<i>PLC_Remote Setup Configuration Select</i> .....	44
Version Level 113 .....	45
<i>PLC_Remote Run Velocity Pot</i> .....	45
Version Level 114 .....	46
<i>New Continuous Press Speed Reached Event</i> .....	46
Version Level 115 .....	47
Version Level 116 .....	48
<i>Enable Second Line Encoder Port</i> .....	48
Version Level 117 .....	49

<i>PLC Remote Control Functions</i> .....	49
Version Level 118 .....	50
<i>PLC Increase Quantity</i> .....	51
<i>PLC Increase Quantity Cancel</i> .....	51
Version Level 119 .....	52
<i>Item Search Information</i> .....	53
<i>Item Search User 1 Field</i> .....	53
<i>Item Search User 2 Field</i> .....	53
Version Level 120 .....	55
PLC Parameter Editing .....	55
Settings Array (v100) .....	56
<i>PLC Remote Tool Configuration Select</i> .....	60
Version Level 121 .....	61
Product Code Comparison Bypass .....	61
Version Level 122 .....	61
Feed to Stop Shear Handshake .....	61
Request Sequence .....	62
Notes: .....	62
Version Level 123 .....	62
<i>XL200 Line Speed</i> .....	62
Version Level 124 .....	63
<i>PLC Quality Audit (Custom List) Trigger</i> .....	63
Version Level 125 .....	63
<i>Receive Item User 3 Field</i> .....	64
<i>Receive Item User 4 Field</i> .....	64
<i>PLC Request Remote Setup Configuration Select</i> .....	64
Document Revisions .....	66

## Introduction

PLCs are often an integral part of a complete roll-forming control system. They can take on many of the non-time critical components of machine control. The benefits are obvious, but many of the data elements and triggers used by the XL200 were not available to the PLC. This created the need for the XL200 controllers to be able to communicate directly with PLCs. With the MODBUS protocol, AMS provides an industry-standard communications interface between an XL200 controller and a PLC. This interface allows the XL200 to support many features including:

- Mapping standard inputs & outputs to the PLC
- Mapping extended inputs & outputs to the PLC
- Mapping external gag outputs to the PLC
- Utilizing a PLC for Y-Axis positioning
- Publishing high-level data to the PLC
- Reporting E-Stop reasons and other PLC diagnostics messages
- Communicating Delay Codes and Scrap Codes between the XL200 and the PLC

## Overview

The XL200 Series controller communicates with a PLC via a MODBUS network. MODBUS is a serial communications specification supported by many PLC manufactures including Allen-Bradley, GE-Fanuc, Automation Direct and others. The MODBUS spec includes two modes of communication: ASCII and Binary (RTU). Only the RTU mode is supported by the XL200. The MODBUS network must be configured in a master-slave arrangement with the XL200 acting as the MODBUS master and a PLC acting as a MODBUS slave.

*Due to communication speed limitations (most PLCs do not support MODBUS baud rates above 38.4 kbps) the MODBUS interface is not well suited for activating time-critical I/O such as punch presses or boost cylinders. A user should have a thorough understanding of data transmission rates and PLC/XL200 scan times before using the MODBUS interface to activate inputs and/or outputs.*

Many higher-end PLC devices provide a MODBUS/TCP interface which is basically the MODBUS serial protocol wrapped in Ethernet packets. The XL200 controller can communicate with MODBUS/TCP devices on an Ethernet network by using an external “gateway” to convert the serial data into Ethernet packets. Communication rates between the XL200 and the “gateway” device can be much higher (115 kbps or higher). Communication rates between the “gateway” device and the PLC are comparable to typical Ethernet data communication rates.

## MODBUS Card Requirements

Some PLC's require add on cards in order to support the MODBUS protocol. **It is very important** that the card not use separate Read/Write address spaces. Many of the feature we support via MODBUS require handshaking techniques that require a shared Read/Write address space.

## Applications

The XL200 PLC Interface is intended for use (but not limited to) the following applications:

## Standard Input/Output Mapping

The state of all of the XL200's standard outputs can be transmitted to the PLC where they can be mapped to memory and monitored by the PLC's program. The PLC can also instruct the XL200 when to activate its standard inputs via the MODBUS interface. This feature can be used to simplify and reduce the amount of wiring needed for a typical installation by eliminating several discrete I/O connections. *Note that any input or output values transmitted serially cannot be activated in real-time and are subject to delays because of slow data transmission rates and scan times. This feature is intended only for inputs and outputs which are not time-critical.*

## External Gag Output Mapping

The XL200 PLC Interface can be used to eliminate the need for the existing "gag board" hardware. The state of the external gag outputs will be transmitted to the PLC where they can be mapped to memory and monitored by the PLC's program.

## Y-Axis and Machine Setup Motion Driver

The XL200 provides an interface with the PLC to allow the PLC to control the low-level movement functions of Y-Axis or Machine Setup Axis. The XL200 provides the high-level functions to determine where and when each axis should move. It monitors status information provided by the PLC to determine if and when the axis has moved to the correct location.

## High Level Data Publishing

The XL200 is capable of publishing several pieces of information that may be useful to a PLC program such as Current Order Number, Current Part Length, Quantity Remaining, etc.

## E-Stop Reasons and Other PLC Diagnostics Information

It is typical for a machine's emergency stop circuit to be wired through the PLC and for some PLCs to display status information regarding the reason for the e-stop condition. This and other status data can be reported to the XL200 controller through the MODBUS interface to provide a more-integrated, easier to use, centralized user interface.

## Delay and Scrap Code Publishing

The PLC may be able to categorize certain machine downtime conditions and transmit this information to the XL200 controller. The XL200 can use this data to simplify the operator's data entry when entering a delay or scrap reason.

## Jog SERCOS PC Axes

Allow Jog forward and reverse functions for each Axes defined on a SERCOS PC.

## PLC Multi-Axis Machine Setup and Tool Positioning

Allow the XL200 controller to send positions, jog and monitor the position of Machine and Tool setup axes.

## **PLC Die Accelerator and Material Remote Control**

Allow the PLC to take control of Die Accelerator positioning and material positioning, utilizing the control algorithms already contained within the XL200 controller.

## **PLC Remote Tool Select**

Allow the PLC to select Tools defined within the XL200 controller and use them for Manual Press operations.

## **Stick Fed Auxiliary Weld Detect Sensors**

Allow the PLC to detect the end of a bar on a stick fed machine using Auxiliary, MODBUS only inputs. The PLC notifies the XL of an edge (end) detect using a hardwired connection and provides a register for the XL to read which of the auxiliary inputs detected the end.

## **PLC Remote Setup Configuration Select**

Allow the PLC to select Setup Configurations defined within the XL200 controller and use them for run mode operations. It was originally defined in Version Level 112 and then extended in Version Level 125.

## **PLC Remote Run Velocity Pot**

Allow the PLC to act as a potentiometer against the Max Velocity setting on a Closed Loop Feed to Stop Controller. This allows the PLC to select almost any speed between Max Velocity and Zero Velocity.

## **PLC Remote Press Speed Communication**

The XL200CL Continuous Press control scheme requires that the XL know what speed the press is operating at. There is a setup parameter that make this possible. The MODBUS interface allows this to be configured remotely by the PLC.

## **PLC Closed Loop Material Feed Status**

When enabled, the MODBUS interface allows the PLC to be notified when a Closed Loop Feed to Stop controller is actively feeding material.

## **PLC Force Single Encoder Operation on CL Feeder**

A Closed Loop Two Encoder Feed to Stop controller closes its loop around the motor encoder when the material tails out. At least one customer needs to be able to force the controller to close the loop around the motor encoder at other times as well. This is generally going to be in conjunction with some other required PLC so it makes sense for this to be controllable via MODBUS.

## **Second Line Encoder Selection**

For Coil Thread-up and Tail-Out scrap optimization having two encoders on some machines is key. Traditionally selecting between encoders has been done with an external Encoder Switching device, but it required identical encoders and encoder wheels. This feature enables a second Line encoder port, unless the controller is a Single Encoder Closed Loop Feeder. When this feature is enabled, a second line encoder resolution setup parameter is also enabled to allow the encoders and encoder wheels to be different.

## PLC Positioning of Multi-Axis Axes

The AMS controller may have Multi-Axis Axes created specifically for use by the PLC. This feature allows the PLC to position those Axes.

## Increase Quantity

The AMS controller has an Increment (Increase) Quantity function that allows a part(s) that fails Quality Control to quickly be replaced with another part. If configured, a scrap code can be entered to indicate why the part or parts are being replaced. Increase Quantity requests can also be canceled. When enabled, the MODBUS interface allows the PLC to initiate the same functions via PLC logic.

## Item Data Request

The AMS controller, in the High-Level Data function, publishes information about the Current Item running and the Next Item to run. If the PLC needs information about the remaining items within the order (Job), this information can be requested by using the ItemId from the Current or Next Item as a Key. The key is used to ask for the Item that is prior or that follows the item whose id was provided. The ItemId of the item requested can then be used to traverse the chain of items until reaching either the first or last Item within the order (job).

## Display and Edit PLC Parameters in XL Setup Screen

Settings within a PLC can now be edited in the XL200 Setup Screen. They can be added to Setup Configurations, Backed Up to Eclipse and Locked with the Setup Lockout Key like any other setup.

## PLC Remote Tool Configuration Select

Allow the PLC to select Tool Configurations defined within the XL200 controller and use them for run mode operations.

## Product Code Comparison Bypass

This feature was added as a Band-Aid for a customer who had created an unmanageable number of PCode's for a Purlin line. The Product code field is intended to describe the Tooling and machine setup for a machine, not including properties of the material required to run the product. This customer had defined many duplicates.

Their machine has a recipe system that compares the properties of the Product Code definitions for machine setup to determine when machine setup is required. It is also a pre-cut line so their mistake does not create extra scrap like it would on many other lines.

Enabling this feature disables Setup and Tool Configuration enforcement by PCode. It also disables the features that stop the Queue on a PCode change.

## Feed to Stop Shear Handshake

This feature was added for Feed to Stop applications where the PLC wants to control when parts get cut. While in automatic, the AMS controller will request and wait for permission from the PLC before cutting the part.

## XL200 Line Speed

This feature was added to publish XL200 calculated Line Speed to the PLC.

## Initiate Quality Audit (Custom List) Triggers

The XL200 has the ability to collect information about Quality or any other sort of information a customer is interested in tracking. The system can be configured to collect the information based on a number of Trigger criteria, for example on a Material Change. This feature allows the PLC to trigger collection of information based on whatever criteria the PLC programmer deems necessary.

## Hardware Connections

The communication protocol specifies that a MODBUS serial system may use any of several different physical interfaces (RS232, RS485, RS422) with Two-Wire RS485 being the most common. The MODBUS implementation on the XL200 will utilize Communications Port D on the F connector and will require a two-wire RS485 cable. An approved, shielded, twisted-pair cable is recommended. The physical connections on the XL200 should be made as follows:

<u>Pin</u>	<u>Description</u>	<u>Details</u>
F11	Shield	Always connect the shield at only one end of the cable
F12	GND	Typically not connected
F13	RX+	Connect one signal wire from RS485 cable (B), also jumper this pin to F16
F14	RX-	Connect one signal wire from RS485 cable (A), also jumper this pin to F15
F15	TX-	Jumper this pin to F14
F16	TX+	Jumper this pin to F13

For wiring details on the PLC, refer to the manufacturer's documentation for the PLC being used. Applications with PLCs that do support an RS485 interface will require an external RS485 to RS232 converter. Applications with PLCs that support MODBUS/TCP will require a MODBUS Serial to MODBUS/TCP gateway device. Several commercial gateway devices are available. Refer to the manufacturer's documentation for wiring details.

## XL200 Software Configuration

### Model Option (Bit Code)

The XL200 controller must have the "1" option in its model number before MODBUS communication is possible.

### Setup Parameters

A new node will be visible in the Setup Menu tree whenever the MODBUS model option is enabled. The new node is titled "PLC Communication" and is located as a submenu under "Controller Settings" directly above the "Operator Preferences" node.

The following new setup parameters will be visible in the PLC Communication menu:

- PLC Unit ID  
This parameter specifies the unit ID for the MODBUS slave device (PLC) that the XL200 communicates with. The valid range for unit Ids is from 1 to 247. The default value will be 0 which disables any and all

MODBUS communication. Note: The PLC must be configured to use the same unit number that is specified by the XL200.

- **PLC Baud Rate**  
This parameter specifies the baud rate used for communicating with the MODBUS slave device (PLC). The following list of baud rates can be selected from the drop-down list: 4800, 9600, 19200, 38400, 57600, 76800, 115200, 230400, and 460800. The default baud rate will be 19200. Note: The PLC must be configured to use the same baud rate as the XL200.
- **PLC Parity**  
This parameter specifies the parity bit setting for communicating with the MODBUS slave device (PLC). The following list of parity settings can be selected from the drop-down list: None, Even, and Odd. According to the MODBUS RTU specification, when Even or Odd parity is selected, 1 stop bit will be used. When No parity is selected, 2 stop bits will be used. *However, most PLC do not conform to the spec in this area so we have added the ability to configure this independent of stop bits.* Note: The PLC must be configured to use the same parity as the XL200
- **PLC Stop Bits**  
This parameter specifies the stop bit setting for communicating with the MODBUS slave device (PLC). According to the MODBUS RTU specification, when Even or Odd parity is selected, 1 stop bit will be used. When No parity is selected, 2 stop bits will be used. *However, most PLC do not conform to the spec in this area so we have added the ability to configure this independent of parity.* Note: The PLC must be configured to use the same stop bit setting as the XL200
- **Configuration Register Address**  
This parameter specifies the starting address for the PLC's holding register that contains the configuration data describing the structure of the data transfer between the XL200 and the PLC. The address must be entered in decimal and must match the address in the PLC where the configuration data structure begins. The configuration data structure will be described later in this document.

## ***XL200 Controller Operation***

In some applications using an AMS controller that is equipped with the PLC Interface feature, it may be critical that the PLC is properly installed and configured before the machine can operate correctly. We will do our best to ensure that machine safety is not sacrificed when using this feature by implementing some "safeguards" into the XL200 controller software.

Following a memory clear, if the PLC Interface has not been configured (i.e., the PLC Unit ID is equal to zero) and the operator tries to enter the run mode, the controller will prompt him with a pop-up window explaining that this controller has the PLC Interface feature but that no PLC is installed. If he chooses to accept this condition, he will not be warned again. If he does not accept, then the operator must enter a valid PLC Unit ID before he will be allowed to enter the run mode.

Once a valid PLC Unit ID has been entered, the controller will test that valid PLC configuration data has been read and that the PLC is currently online before allowing any change to the controller's run mode. This includes manual jog functions, manual punch and shear functions, as well as entering the run mode. If this test fails, the controller will display an appropriate error message.

Once the PLC has been installed and valid configuration data has been read, the controller will consider the PLC to be online. Any communication failure between the two devices will cause the controller to display a “PLC Communication Failure” error message and exit the run mode (if it was currently running). The PLC will then be considered offline until the operator goes to the Diagnostics screen, Printer Communication menu, and presses a function key to restore communications and bring the PLC back online.

## PLC Configuration Data Structure

The following data structure must be provided by the PLC and located in holding registers at the address specified by the Configuration Register Address parameter in the XL200 controller. This data will be read by the XL200 during its power up sequence and will determine which pieces of data will be transferred between the XL200 and the PLC on a regular basis. Within the structure there are various Configuration word values. Each bit within the Configuration word values represents a feature or function that is selectively enabled by the bit being ON.

As features are added, new fields are added to the end that are required to support those features. In order for the XL200 controller to know which version (Level) of the structure the PLC is using there is a word value in the beginning that allows the XL200 to know how much more memory it needs to read to get the remainder of the Configuration.

The following table represents the fields within the table and has a handy reference of each fields offset from the beginning of the structure in decimal, octal and hex formats. The table also has the version level when the field was introduced or when the field is used.

With the exception of the first field, which is a 32bit Double Word (DWORD), the remaining fields are all 16 bit Word Values (WORD).

Data Name	Version Level	Address Offset (Decimal, Octal and Hex respectively)
cAMSIdentifier	100	0, 00, 0x0
wVersionLevel	100	2, 02, 0x2
wConfigurationBits00	100	3, 03, 0x3
wXLStandardOutputsAddress	100	4, 04, 0x4
wXLStandardInputsAddress	100	5, 05, 0x5
wStartOfCoilsAddress	100	6, 06, 0x6
wPLCFunctionCoilsIndex	100	7, 07, 0x7
wPLCErrorMsgAddress	100	8, 010, 0x8
wPLCDiagnosticsMsgAddress	100	9, 011, 0x9
wCurrentItemInfoAddress	101	10, 012, 0xA
wNextItemInfoAddress	101	11, 013, 0xB
wXLExternalGagOutputsAddress	100	12, 014, 0xC
wConfigurationBits01	101	13, 015, 0xD
wCurOrderNumAddress	101	14, 016, 0xE
wCurMaterialCodeAddress	101	15, 017, 0xF
wCurPCCodeAddress	101	16, 020, 0x10
wCurOrdUser1Address	101	17, 021, 0x11
wCurOrdUser2Address	101	18, 022, 0x12
wCurOrdUser3Address	101	19, 023, 0x13
wCurOrdUser4Address	101	20, 024, 0x14
wCurOrdUser5Address	101	21, 025, 0x15
wCurOrdUser6Address	101	22, 026, 0x16
wCurOrdUser7Address	101	23, 027, 0x17
wCurOrdUser8Address	101	24, 030, 0x18
wCurItmUser1Address	101	25, 031, 0x19
wCurItmUser2Address	101	26, 032, 0x1A
wPLCDelayCodeAddress	101	27, 033, 0x1B

wPLCScrapCodeAddress	101	28, 034, 0x1C
wJogAxisForwardBitsAdd	102	29, 035, 0x1D
wJogAxisReverseBitsAdd	102	30, 036, 0x1E
wJogAxisCount	102	31, 037, 0x1F
wConfigurationBits03	103	32, 040, 0x20
wPLCNotificationEventAdd	103	33, 041, 0x21
wConfigurationBits04	104	34, 042, 0x22
wPLCAxisStructuresAdd	104	35, 043, 0x23
wPLCAxisStructuresCount	104	36, 044, 0x24
wPLCAxisStructureVersion	104	37, 045, 0x25
wConfigurationBits05	105	38, 046, 0x26
wPLCAxisMaterialChangeAdd	105	39, 047, 0x27
wPLCAxisPCodeChangeAdd	105	40, 050, 0x28
wNumberOfExtrInputWords	108	41, 051, 0x29
wNumberOfExtraOutputWords	108	42, 052, 0x2A
wRemoteControlStructAdd	109	43, 053, 0x2B
wRemoteControlStructVer	109	44, 054, 0x2C
wRemoteToolSelectAdd	110	45, 055, 0x2D
wAuxWeldDetects	111	46, 056, 0x2E
wAuxWeldDetectRegAdd	111	47, 057, 0x2F
wRemoteSetupConfigSelAdd	112	48, 060, 0x30
wRemoteRunVelocityPotAdd	113	49, 061, 0x31
wRemotePressSpeedAdd	114	50, 062, 0x32
wMAPositioningStructAdd	117	51, 063, 0x33
wMAPositioningStructVer	117	52, 064, 0x34
wIncQuantityStructAdd	118	53, 065, 0x35
wIncQuantityStructVer	118	54, 066, 0x36
wSearchItemAdd	119	55, 067, 0x37
wSearchItemUser1Add	119	56, 070, 0x38
wSearchItemUser2Add	119	57, 071, 0x39
wIdSearchAdd	119	58, 072, 0x3A
wOrderFieldSize	119	59, 073, 0x3B
wMaterialFieldSize	119	60, 074, 0x3C
wPCodeSize	119	61, 075, 0x3D
wUserFieldSize	119	62, 076, 0x3E
wConfigurationBits120	120	63, 077, 0x3F
wSettingsArrayVersion	120	64, 0100, 0x40
wNumberOfSettings	120	65, 0101, 0x41
wSettingsArrayAdd	120	66, 0102, 0x42
wAMSsettingsUpdateCoilsIndex	120	67, 0103, 0x43
wPLCsettingsUpdateCoilsIndex	120	68, 0104, 0x44
wRemoteToolConfigSelAdd	120	69, 0105, 0x45
wLineSpeedUpdateRate	123	70, 0106, 0x46
wLineSpeedInchPerSecAdd	123	71, 0107, 0x47
wQATriggerSelectAdd	124	72, 0110, 0x48
wCurItmUser3Address	125	73, 0111, 0x49
wCurItmUser4Address	125	74, 0112, 0x50

**IMPORTANT NOTE: Every member of the configuration structure should be initialized on the first scan of the PLC program in order to prevent unintended operation and as good programming practice.**

The first member of the structure `cAMSIdentifier[4]` is an array of four characters and must contain the following four bytes of data in this order (all values in hex): 0x03, 0x41, 0x4D, and 0x53. These are the ASCII values for the text string “AMS” preceded by the number of characters in the string. For troubleshooting purposes, depending on the age of the AMS Controller software, these four bytes are displayed in hex as a 32bit numeric value on the MODBUS Diagnostics screen. The AMS controller uses Little Endian format. A valid AMS ID will be displayed as 534D4103.

The next member of the structure `wVersionLevel` determines the remaining structure of the configuration data and the structures for all other user data. Once the XL200 has read this version level, it knows the structure of all the remaining user data which can be transferred back and forth to the PLC. Having a version level allows us to expand the configuration data structure and the other user data structures in the future so that as new features and capabilities are added, the version level will be increased. Therefore, newer releases of XL200 software will always be backwards compatible with older PLC programs because the XL200 can identify the version level in the PLC and use the older data structures. The data structure for the initial version level and subsequent version levels will be described later in this document.

The first member of the version level structure `wConfigurationBits00` represents a configuration register where each individual bit enables or disables a particular structure of data to be transferred between the XL200 and the PLC. For example, one bit will be used to enable the transfer of the XL200’s standard 24 output values to the PLC. When this bit is set, the current values for each of the XL200’s 24 outputs will be transferred to the PLC during each “scan” of the communication loop. If this bit is not set, this particular data transfer will be disabled. Using individual configuration bits allows the PLC user to select only the data he wants to transfer during each scan, thus keeping the communication scan time to a minimum. The definition for all configuration bits is given in the section below.

The next few words in the version level structure define the starting addresses in the PLC’s memory where the transferred data will be stored. The address for each particular structure of data must be specified separately. This gives the PLC user the flexibility of locating specific pieces of data at various places in the PLC’s memory. He can, for example, specify that the XL200’s standard outputs be mapped directly to coil memory, while PLC diagnostics messages are stored in holding register memory. **IMPORTANT NOTE: The user is responsible for ensuring that none of the data structures overlap one another, and that they do not conflict with any other memory locations being used by the PLC logic program.** The definitions and sizes for all data structures are given in the section below.

`WStartOfCoilsAddress` is the address of where coils begin in the PLC’s memory map. **See the PLC Coil Addressing Section.**

The first word of the Version Level 101 structure, `wConfigurationBits01` represents a configuration register where each individual bit enables or disables a particular structure of data to be transferred between the XL200 and the PLC.

All other members of the configuration structure are described below.

## **PLC Coil Addressing**

`WStartOfCoilsAddress` is the address of where coils begin in the PLC’s memory map. **This is very important to understand.** Set this setting to the address that defines where in memory MODBUS commands are able to access the very first coil in the PLC’s memory.

**Important Note: All other coil index settings within this document are relative to this address.**

**For example:**

If the first coil in memory according to the memory map for MODBUS for your PLC is at 1001 decimal, set WStartOfCoilsAddress to 1001. Then, if you want to use the 32'd coil in memory for one of the coil index configuration settings in this document you will set that setting to 32.

## PLC Configuration Field Bits

Within the PLC Configuration Data Structure are configuration words. Bits within these words enable or configure various features available in the XL200/PLC interface. Below is a reference of each configuration word, each configuration bit and which Version Level it was introduced in:

Enumeration	Bit Value (hex)	Configuration Word	Level
CONFIG_0_XL_STANDARD_OUTPUTS	0x0001	wConfigurationBits00	100
CONFIG_0_XL_STANDARD_INPUTS	0x0002	wConfigurationBits00	100
CONFIG_0_STROBE_WATCHDOG_COIL	0x0004	wConfigurationBits00	100
CONFIG_0_PLC_ERROR_MSG	0x0008	wConfigurationBits00	100
CONFIG_0_PLC_DIAGNOSTICS_MSG	0x0010	wConfigurationBits00	100
CONFIG_0_CURRENT_ITEM_INFO	0x0020	wConfigurationBits00	101
CONFIG_0_NEXT_ITEM_INFO	0x0040	wConfigurationBits00	101
Reserved	0x0080	wConfigurationBits00	100
CONFIG_0_EXTERNAL_GAG_1_16	0x01XX	wConfigurationBits00	100
CONFIG_0_EXTERNAL_GAG_1_32	0x02XX	wConfigurationBits00	100
CONFIG_0_EXTERNAL_GAG_1_48	0x03XX	wConfigurationBits00	100
CONFIG_0_EXTERNAL_GAG_1_64	0x04XX	wConfigurationBits00	100
CONFIG_1_CURRENT_ORDER_NUMBER	0x0001	wConfigurationBits01	101
CONFIG_1_CURRENT_MATERIAL_CODE	0x0002	wConfigurationBits01	101
CONFIG_1_CURRENT_PRODUCT_CODE	0x0004	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER1	0x0008	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER2	0x0010	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER3	0x0020	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER4	0x0040	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USERS5	0x0080	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER6	0x0100	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER7	0x0200	wConfigurationBits01	101
CONFIG_1_CURRENT_ORDER_USER8	0x0400	wConfigurationBits01	101
CONFIG_1_CURRENT_ITEM_USER1	0x0800	wConfigurationBits01	101
CONFIG_1_CURRENT_ITEM_USER2	0x1000	wConfigurationBits01	101
CONFIG_1_DELAY_CODE_NOTIFICATION	0x2000	wConfigurationBits01	101
CONFIG_1_SCRAP_CODE_NOTIFICATION	0x4000	wConfigurationBits01	101
CONFIG_1_JOG_AXIS_FUNCTIONS	0x8000	wConfigurationBits01	102
CONFIG_3_PLC_NOTIFICATION_EVENT	0x0001	wConfigurationBits03	103
Reserved	0xFFFF	wConfigurationBits03	103
CONFIG_4_ENABLE_PLC_AXIS_STRUCTURES	0x0001	wConfigurationBits04	104
Reserved	0xFFFF	wConfigurationBits04	104
CONFIG_5_MATERIAL_CHANGE_EVENTS	0x0001	wConfigurationBits05	105
CONFIG_5_PCODE_CHANGE_EVENTS	0x0002	wConfigurationBits05	105
CONFIG_5_CUSTOM_FUNCTION	0x0004	wConfigurationBits05	107
CONFIG_5_REMOTE_CONTROL_FUNCTIONS	0x0008	wConfigurationBits05	109
CONFIG_5_REMOTE_TOOL_SELECT	0x0010	wConfigurationBits05	110
CONFIG_5_AUX_WELD_DETECTS_ENABLED	0x0020	wConfigurationBits05	111
CONFIG_5_REMOTE_SETUP_CONFIG_SELECT	0x0040	wConfigurationBits05	112
CONFIG_5_REMOTE_RUN_VELOCITY_POT	0x0080	wConfigurationBits05	113
CONFIG_5_REMOTE_CONTINUOUS_PRESS_SPEED	0x0100	wConfigurationBits05	114

CONFIG_5_ENABLE_ENCODER_SELECTION	0x0200	wConfigurationBits05	114
CONFIG_5_MATERIAL_FEED_MOTION	0x0400	wConfigurationBits05	115
CONFIG_5_ENABLE_SECOND_LINE_ENCODER	0x0800	wConfigurationBits05	116
CONFIG_5_MULTI_AXIS_POSITION_FUNCTIONS	0x1000	wConfigurationBits05	117
CONFIG_5_INCREASE_QTY_FUNCTIONS	0x2000	wConfigurationBits05	118
CONFIG_5_ITEM_DATA_REQUEST	0x4000	wConfigurationBits05	119
CONFIG_5_SPECIFY_STRING_FIELD_SIZE	0x8000	wConfigurationBits05	119
CONFIG_120_PLC_PARAMETER_SUPPORT	0x0001	wConfigurationBits120	120
CONFIG_120_XL_UPDATE_COILS	0x0002	wConfigurationBits120	120
CONFIG_120_PLC_UPDATE_COILS	0x0004	wConfigurationBits120	120
CONFIG_120_PLC_REMOTE_TOOL_CONFIGURATION_SELECT	0x0008	wConfigurationBits120	120
CONFIG_120_PCODE_COMPARE_BYPASS	0x0010	wConfigurationBits120	121
CONFIG_120_FTS_SHEAR_HANDSHAKE	0x0020	wConfigurationBits120	122
CONFIG_120_XL_LINE_SPEED	0x0040	wConfigurationBits120	123
CONFIG_120_QA_TRIGGER	0x0080	wConfigurationBits120	124
CONFIG_120_CURRENT_ITEM_USER3	0x0100	wConfigurationBits120	125
CONFIG_120_CURRENT_ITEM_USER4	0x0200	wConfigurationBits120	125
CONFIG_120_REQUEST_SETUP_CONFIG_SELECT	0x0400	wConfigurationBits120	125

## PLC Function Coils

Function Coils are special coil bits defined in the PLC that allow the XL200 to quickly determine if the PLC is requesting it to do something or for the XL200 to quickly tell the PLC something.

All of the coils are sequential and contiguous in memory which allows the XL200 to monitor all of them with one request.

wStartOfCoilsAddress in the PLC Configuration Structure specifies where coils begin in the PLC's memory map. See the PLC Coil Addressing section of the manual.

wPLCFunctionCoilsIndex is the index of the first of PLC function request coils. For example, If you wish the first function coil, FUNC\_0\_STROBE\_WATCHDOG\_COIL, to reside at the 32'd coil within the PLC's memory, set this to 32.

The first function that will be fully described later is a Watch Doc Coil. If the PLC's program desires the watch dog coil to be the second coil (C1) in memory then the wPLCFunctionCoilsIndex should be set to 1 since the index of the first coil in memory has an index of zero(C0).

This is the complete table of all defined function coils:

Name	Coil	Version Level
FUNC_0_STROBE_WATCHDOG_COIL	0	100
FUNC_0_PLC_ERROR_MSG	1	100
FUNC_0_PLC_DIAGNOSTICS_MSG	2	100
FUNC_1_PLC_DELAY_CODE	3	101
FUNC_1_PLC_SCRAP_CODE	4	101
FUNC_1_SENDING_CURRENT_ORDER_ITEM_INFO	5	101
FUNC_1_SENDING_NEXT_ORDER_ITEM_INFO	6	101
FUNC_2_JOG_AXIS_BITS_CHANGED	7	102
FUNC_3_PLC_EVENT_NOTIFICATION	8	103
FUNC_5_PCODE_AXIS_CHANGE_EVENT	9	105
FUNC_5_MATERIAL_AXIS_CHANGE_EVENT	10	105
FUNC_5_RESERVED_12	11	107
FUNC_5_TOOL_SELECT_CHANGED	12	110
FUNC_5_TEST_FOR_BAR_END	13	111
FUNC_5_REMOTE_RUN_VEL_POT_CHANGE	14	113
FUNC_5_REQUEST_PRESS_SPEED_CHANGE	15	114
FUNC_14_OK_TO_CHANGE_PRESS_SPEED	16	114
FUNC_14_NEW_PRESS_SPEED_REACHED	17	114
FUNC_14_SELECT_MOTOR_ENCODER	18	114
FUNC_14_MATERIAL_FEED_MOTION	19	115
FUNC_14_MULTI_AXIS_POSITION_FUNC_REQUEST	20	117
FUNC_18_INCREASE_QUANTITY_REQUEST	21	118
FUNC_18_INCREASE_QUANTITY_CANCEL	22	118
FUNC_19_ITEM_NEXT_REQUEST	23	119
FUNC_19_ITEM_PREVIOUS_REQUEST	24	119
FUNC_120_PLC_PARAMETER_UPDATE	25	120
FUNC_122_XL_REQUEST_SHEAR	26	122
FUNC_122_PLC_OKS_SHEAR	27	122

FUNC_122_XL_SHEAR_ACTIVE	28	122
FUNC_122_PLC_SEES_SHEAR	29	122
FUNC_124_QA_TRIGGER_REQUEST	30	124
FUNC_125_REQUEST_SETUP_CONFIGURATION_SELECT	31	125

## PLC Configuration Bits and User Data Structures

### Version Level 100

Configuration bits for wConfiguration00 Bits are defined below:

Enumeration	Bit Value (hex)
CONFIG_0_XL_STANDARD_OUTPUTS	0x0001
CONFIG_0_XL_STANDARD_INPUTS	0x0002
CONFIG_0_STROBE_WATCHDOG_COIL	0x0004
CONFIG_0_PLC_ERROR_MSG	0x0008
CONFIG_0_PLC_DIAGNOSTICS_MSG	0x0010
!!!Reserved Bits for future use	0xE0
Configuration for Number of Gag Words	Byte Value (hex) of most significant byte in the configuration Word (XX is least significant byte)
CONFIG_0_EXTERNAL_GAG_1_16	0x01XX
CONFIG_0_EXTERNAL_GAG_1_32	0x02XX
CONFIG_0_EXTERNAL_GAG_1_48	0x03XX
CONFIG_0_EXTERNAL_GAG_1_64	0x04XX

#### PLC Function Request Coils

User Data Size 3 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils for Version Level 100 are:

Enumeration	Coil
FUNC_0_STROBE_WATCHDOG_COIL	0
FUNC_0_PLC_ERROR_MSG	1
FUNC_0_PLC_DIAGNOSTICS_MSG	2

#### Map XL200 Standard Outputs to PLC

Configuration Bit: CONFIG\_0\_XL\_STANDARD\_OUTPUTS (0x0001)

Starting address in PLC memory specified by: wXLStandardOutputsAddress

User Data Size: 2 words

Command Used: MODBUS (16) Write Holding Registers

Enabling this feature causes the XL200 to transfer the current state of all 24 standard outputs to the PLC during each communication scan. The data will be transferred as two data words with one bit representing the state of each output (logic 1 = ON, logic 0 = OFF). Output 1 corresponds to the least significant bit in the structure and only the lower 24 bits will be used. For example, a value of 0x0091 indicates that outputs 1, 5, and 8 are ON and all other outputs are OFF. *IMPORTANT NOTE: To help keep scan times to a minimum, the output values will only be transferred when one or more of the outputs has changed state since the previous scan.*

#### Map XL200 Standard Inputs to PLC

Configuration Bit: CONFIG\_0\_XL\_STANDARD\_INPUTS (0x0002)

Starting address in PLC memory specified by: wXLStandardInputsAddress

User Data Size: 2 words

Command Used: MODBUS (3) Read Holding Registers

Enabling this feature causes the XL200 to read a value from the PLC's memory during each communication scan. This value will be logically OR'ed with the XL200 standard inputs to effectively allow the PLC to activate a controller input without being hard wired to the terminal strip. The data will be transferred as two data words with one bit representing the state of each input (logic 1 = ON, logic 0 = OFF). Input 1 corresponds to the least significant bit in the structure and only the lower 24 bits will be used. For example, a value of 0x0008 indicates that the PLC wishes to activate input 4 on the XL200. *IMPORTANT NOTE: Input values are read during each communication scan when this feature is enabled. Each input will be active on the XL200 controller for the duration of time that the corresponding bit is set in the PLC's memory.*

Strobe Watchdog Coil

Configuration Bit: CONFIG\_0\_STROBE\_WATCHDOG\_COIL (0x0004)  
 Coil address in PLC memory specified by: wPLCFunctionCoilsIndex  
 User Data Size: 1 Coil  
 Command Used: MODBUS (5) Write Single Coil

Enabling this feature causes the XL200 to turn on the specified coil at least once every 5 seconds. This safety feature is intended to be used by the PLC to monitor whether or not the XL200 controller is actively communicating with it. If the PLC detects that the specified coil has not turned on for more than 5 seconds, it can assume that it has lost communication with the XL200 and can take the appropriate action (i.e. turn off critical outputs, etc.). The PLC program is responsible for turning this coil off each time it detects that the coil has been turned on.

Report PLC Error Message

Configuration Bit: CONFIG\_0\_PLC\_ERROR\_MSG (0x0008)  
 Starting address in PLC memory specified by: wPLCErrorMsgAddress  
 User Data Size: 41 words  
 Command Used: MODBUS (3) Read Holding Registers

Enabling this feature allows the XL200 to display an error message reported by the PLC that indicates a machine stoppage condition. Approximately once every second, the XL200 checks to see if the PLC has an error message to display. If so, the XL200 exits the run mode, reads the error message from the PLC and displays the text as an error message on the XL200 display. The message text may be up to 80 characters long. The first word of this user data structure should contain the number of characters in the text message. The remaining 40 words should contain the ASCII values for the message (two characters per word). The XL200 reads the PLC Function coils (FUNC\_0\_PLC\_ERROR\_MSG) to determine if there is a text message from the PLC. If a message is detected it reads the message and then immediately turns off the FUNC\_0\_PLC\_ERROR\_MSG function coil to inform the PLC that the message has been read. When creating a new message to display, the PLC user should ensure that the complete message text is filled in before turning on the PLC function coil to inform the XL of the message.

Report PLC Diagnostics Message

Configuration Bit: CONFIG\_0\_PLC\_DIAGNOSTICS\_MSG (0x0010)  
 Starting address in PLC memory specified by: wPLCDiagnosticsMsgAddress  
 User Data Size: 41 words  
 Command Used: MODBUS (3) Read Holding Registers

Enabling this feature allows the XL200 to display a diagnostics message reported by the PLC but does not necessarily indicate a machine stoppage condition. This feature works exactly the same as the PLC Error Message feature but does not cause the XL200 Controller to exit the run mode when the message is displayed and it uses the FUNC\_0\_PLC\_DIAGNOSTICS\_MSG function coil.



## Version Level 101

This Version Level adds additional functionality and configuration bits to the Version Level 100 configuration structure. It uses some of the configuration bits and addresses that were reserved for future use in the Version Level 100 configuration structure

Enumeration	Bit Value (hex)	Configuration Word
!!!Defined in Version Level 101	0xFF1F	wConfiguration00
CONFIG_0_CURRENT_ITEM_INFO	0x0020	wConfiguration00
CONFIG_0_NEXT_ITEM_INFO	0x0040	wConfiguration00
!!!Reserved for future use	0x0080	wConfiguration00
CONFIG_1_CURRENT_ORDER_NUMBER	0x0001	wConfiguration01
CONFIG_1_CURRENT_MATERIAL_CODE	0x0002	wConfiguration01
CONFIG_1_CURRENT_PRODUCT_CODE	0x0004	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER1	0x0008	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER2	0x0010	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER3	0x0020	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER4	0x0040	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER5	0x0080	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER6	0x0100	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER7	0x0200	wConfiguration01
CONFIG_1_CURRENT_ORDER_USER8	0x0400	wConfiguration01
CONFIG_1_CURRENT_ITEM_USER1	0x0800	wConfiguration01
CONFIG_1_CURRENT_ITEM_USER2	0x1000	wConfiguration01
CONFIG_1_DELAY_CODE_NOTIFICATION	0x2000	wConfiguration01
CONFIG_1_SCRAP_CODE_NOTIFICATION	0x4000	wConfiguration01
!!!Reserved for future use	0x8000	wConfiguration01

### PLC Function Request Coils

User Data Size 7 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils added for Version Level 101 are:

Enumeration	Coil (0 based index)
FUNC_1_PLC_DELAY_CODE	3
FUNC_1_PLC_SCRAP_CODE	4
FUNC_1_SENDING_CURRENT_ORDER_ITEM_INFO	5
FUNC_1_SENDING_NEXT_ORDER_ITEM_INFO	6

### Receive Current Item Information

Configuration Bit: CONFIG\_0\_CURRENT\_ITEM\_INFO (0x0020)  
 Starting address in PLC memory specified by: wCurrentItemInfoAddress  
 User Data Size: 6 or 17 words  
 Command Used: MODBUS (16) Write Holding Registers

There are two different structures depending on what Control System is connected to the PLC.

User Data Structure for XL200 Series Roll-Forming Controls:

Data Size	Field	Range	Level
WORD	Bundle	0-999	101

WORD	QuantityProgrammed	0-32767	101
WORD	QuantityRemaining	0-32767	101
DWORD	PartLength	0-3500000 (0.001")	101
WORD	Pattern	0-999	101
DWORD	ItemID	Not present until Level 106	106
DWORD	ParentID	Not present until Level 106	106

Enabling this feature allows tells the XL200 to send information about the current item, running or about to run. The information will be sent when the XL powers up or if any of the information within the data structure changes. *Note: The Part Length field is sent in units of one thousandths of an inch.*

User Data Structure for XL220 HVAC controller:

Data Size	Field	Range	Level
WORD	Bundle	0-999	101
WORD	QuantityProgrammed	0-32767	101
WORD	QuantityRemaining	0-32767	101
DWORD	Height	0-999000 (0.001")	101
DWORD	Width	0-999000 (0.001")	101
DWORD	Offset	0-999000 (0.001")	101
DWORD	DHPosition	0-3996000 (.001")	101
DWORD	BendAllowance	0- 999000 (.001")	101
DWORD	CornerRotation	0-999000 (.001")	101
WORD	PartType 0 = Four Piece 1 = L Shaped 2 = U Shaped 3 = Full Wrapper(4 legs, 3 V-Notches) 4 = Shear Only 5 = Split Wrapper(5 legs, 4 V-notches. Leg 1,5 are ½ leg3)	0-5	101
WORD	Velocity 0 = No Insulation 1 = High 2 = Low 3 = Special	0-3	
DWORD	ItemID	Not present until Level 106	106
DWORD	ParentID	Not present until Level 106	106

Enabling this feature allows tells the XL220 to send information about the current item, running or about to run. The information will be sent when the XL powers up or if any of the information within the data structure changes. *Note: All fields followed by the text (.001") are sent in units of one thousandths of an inch.*

**IMPORTANT NOTE:** To help keep scan times to a minimum, the structure will only be transferred when one or more of the fields within the structure has changed state since the previous scan.

**IMPORTANT NOTE2:** The `CONFIG_1_SENDING_CURRENT_ORDER_ITEM_INFO` function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.

#### Receive Next Item Information

Configuration Bit: CONFIG\_0\_NEXT\_ITEM\_INFO (0x0040)  
 Starting address in PLC memory specified by: wNextItemInfoAddress  
 User Data Size: 6 or 17 words  
 Command Used: MODBUS (16) Write Holding Registers

There are two different structures depending on what Control System is connected to the PLC. They are defined identically to the structures referenced in the Receive Current Item Information function.

Enabling this feature allows tells the XL200 to send information about the next item to run, after the running or about to run item. The information will be sent when the XL powers up or if any of the information within the data structure changes. *Note: The Part Length field is sent in units of one thousandths of an inch.*

Enabling this feature allows tells the XL220 to send information about the next item to run, after the running or about to run item. The information will be sent when the XL powers up or if any of the information within the data structure changes. *Note: All fields followed by the text (.001") are sent in units of one thousandths of an inch.*

**IMPORTANT NOTE:** To help keep scan times to a minimum, the structure will only be transferred when one or more of the fields within the structure has changed state since the previous scan.

**IMPORTANT NOTE2:** The `FUNC_1_SENDING_NEXT_ORDER_ITEM_INFO` function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.

#### Receive Current Order Number

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_NUMBER (0x0001)  
 Starting address in PLC memory specified by: wCurOrderNumAddress  
 User Data Size: 11 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order number. The first word in the data indicates the length of the string.

**IMPORTANT NOTE:** To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.

**IMPORTANT NOTE2:** The `FUNC_1_SENDING_CURRENT_ORDER_ITEM_INFO` function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.

#### Receive Current Material Code

Configuration Bit: CONFIG\_1\_CURRENT\_MATERIAL\_CODE (0x0002)  
 Starting address in PLC memory specified by: wCurMaterialCodeAddress  
 User Data Size: 11 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current Material Code. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

#### Receive Current Product Code

Configuration Bit: CONFIG\_1\_CURRENT\_PRODUCT\_CODE (0x0004)  
Starting address in PLC memory specified by: wCurPCodeAddress  
User Data Size: 11 words  
Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current Product Code. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

#### Receive Order User 1 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USER1 (0x0008)  
Starting address in PLC memory specified by: wCurOrdUser1Address  
User Data Size: 16 words  
Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 1. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

#### Receive Order User 2 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USER2 (0x0010)  
Starting address in PLC memory specified by: wCurOrdUser2Address  
User Data Size: 16 words  
Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 2. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Order User 3 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USERS3 (0x0020)  
 Starting address in PLC memory specified by: wCurOrdUser3Address  
 User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 3. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Order User 4 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USER4 (0x0040)  
 Starting address in PLC memory specified by: wCurOrdUser4Address  
 User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 4. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Order User 5 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USERS5 (0x0080)  
 Starting address in PLC memory specified by: wCurOrdUser5Address  
 User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 5. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

Receive Order User 6 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USER6 (0x0100)  
Starting address in PLC memory specified by: wCurOrdUser6Address  
User Data Size: 16 words  
Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 6. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Order User 7 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USER7 (0x0200)  
Starting address in PLC memory specified by: wCurOrdUser7Address  
User Data Size: 16 words  
Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 7. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Order User 8 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ORDER\_USER8 (0x0400)  
Starting address in PLC memory specified by: wCurOrdUser8Address  
User Data Size: 16 words  
Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current order User Field 8. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Item User 1 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ITEM\_USER1 (0x0800)  
Starting address in PLC memory specified by: wCurltmUser1Address

User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current item User Field 1. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

#### Receive Item User 2 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ITEM\_USER2 (0x1000)  
 Starting address in PLC memory specified by: wCurltmUser2Address  
 User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current item User Field 2. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

#### Report Delay Code

Configuration Bit: CONFIG\_1\_DELAY\_CODE\_NOTIFICATION (0x2000)  
 Starting address in PLC memory specified by: wPLCDelayCodeAddress  
 User Data Size: 1 word  
 Command Used: MODBUS (3) Read Holding Registers

Enabling this feature allows the XL200 to pre-populate the delay code in the delay code data entry window. Approximately once every second, the XL200 checks the PLC Function request coils to see if the PLC has a delay code to pre-populate. If so, the XL200 reads the delay code from the PLC and then immediately turns off the PLC Delay Code function coil to inform the PLC that the delay code has been read. When reporting a new delay code, the PLC user should ensure that the delay code is filled in before turning on the PLC function coil to inform the XL of the delay code.

#### Report Scrap Code

Configuration Bit: CONFIG\_1\_SCRAP\_CODE\_NOTIFICATION (0x4000)  
 Starting address in PLC memory specified by: wPLCScrapCodeAddress  
 User Data Size: 1 word  
 Command Used: MODBUS (3) Read Holding Registers

Enabling this feature allows the XL200 to pre-populate the scrap code in the scrap code data entry window. Approximately once every second, the XL200 checks the PLC Function request coils to see if the PLC has a scrap

code to pre-populate. If so, the XL200 reads the scrap code from the PLC and then immediately turns off the PLC Scrap Code function coil to inform the PLC that the scrap code has been read. When reporting a new scrap code, the PLC user should ensure that the scrap code is filled in before turning on the PLC function coil to inform the XL of the scrap code.

## Version Level 102

This Version Level adds one configuration bit to the Version Level 101 configuration structure. The Additional configuration bits for wConfiguration01 Bits is defined below:

Enumeration	Bit Value (hex)	Configuration Word
!!!Previously defined in Version Level 101	0x7FFF	wConfiguration01
CONFIG_1_JOG_AXIS_FUNCTIONS	0x8000	wConfiguration01

### PLC Function Request Coils

User Data Size 8 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils added for Version Level 102 are:

Enumeration	Coil (0 based index)
FUNC_2_JOG_AXIS_BITS_CHANGED	7

### Jog Axis Function

Configuration Bit: CONFIG\_1\_JOG\_AXIS\_FUNCTIONS

Data address in PLC memory specified by:

wJogAxisForwardBitsAdd, wJogAxisForwardBitsAdd

User Data Size: (wJogAxisCount/16)+1 Words

Command Used: MODBUS (3) Read Holding Registers

Enabling this feature allows the PLC to inform the XL200 that an Axis is requesting to Jog forward or reverse.

wJogAxisForwardBitsAdd specifies what word in the PLC’s memory the Jog Axis Forward request bits start.

wJogAxisReverseBitsAdd specifies what word in the PLC’s memory the Jog Axis Reverse request bits start.

wJogAxisCount specifies how many axes will be requesting jog functions and, using the above formula, how many words will be read from the PLC’s memory by the XL200 controller. wJogAxisCount must not be larger than 255.

To keep MODBUS scan times to a minimum the following procedure must be followed when an Axis wants to jog. The Axis should set the FUNC\_2\_JOG\_AXIS\_BITS\_CHANGED function coil. This will inform the XL200 controller to start reading the array of words containing the Jog Axis Forward and Jog Axis Reverse bits. These words will continue to be read until all bits turn off and then the XL200 will turn off the FUNC\_2\_JOG\_AXIS\_BITS\_CHANGED function coil.

Here are several examples of how the Jog Forward Bits are used. If the first word of the Jog Forward Bits is has the value 0x0001 then Axis Id 1 will be requested to jog forward because it is the 1st bit in the word. If the first word of the Jog Forward Bits has the value 0x8000 then Axis Id 16 will be requested to jog forward because it is the 16th bit in the word. If the second word of the Jog Forward Bits has the value 0x0001 then Axis Id 17 will be requested to jog forward because it is the 17’t h bit in the array of words. The same examples apply to the Jog Reverse Bits.

## Version Level 103

This Version Level adds wConfiguration03. The configuration bits for wConfiguration03 in Version Level 103 are defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_3_PLC_NOTIFICATION_EVENT	0x0001	wConfiguration03
!!!Reserved for future use	0xFFFE	wConfiguration03

### PLC Function Request Coils

User Data Size 9 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils added for Version Level 103 are:

Enumeration	Coil (0 based index)
FUNC_3_PLC_EVENT_NOTIFICATION	8

### PLC Notification Event

Configuration Bit: CONFIG\_3\_PLC\_NOTIFICATION\_EVENT (0x0001)

Starting address in PLC memory specified by: wPLCNotificationEventAdd

User Data Size: 1 word

Command Used: MODBUS (3) Read Holding Registers

A PLC Notification Event allows the PLC Programmer to move Error and Diagnostics String definitions outside of the PLC Program. It also allows the PLC Programmer to enable one feature that can selectively cause a message to be displayed on the XL Controller and pre-populate a delay or scrap code. It will be expanded to include other options in the future. What occurs in the XL controller when a PLC Notification event is generated is completely configured outside of the PLC which makes the PLC Programmers life easier.

Approximately once every second, the XL200 reads the PLC Function coils (FUNC\_3\_PLC\_EVENT\_NOTIFICATION) to see if the PLC has a Notification Event to report. If so, the XL200 reads the register indicated by wPLCNotificationEventAdd. Then the XL200 turns off the FUNC\_3\_PLC\_EVENT\_NOTIFICATION function coil to inform the PLC that the event has been read. Valid values for the event id range from 0 through 65,535. The XL will look this number up in a database to determine if it should display an error, diagnostic message, pre-populate a scrap or delay code.

The PLC programmer should ensure that the event id is filled in before turning on the PLC function coil to inform the XL of the event.

**IMPORTANT NOTE:** The CONFIG\_3\_PLC\_NOTIFICATION\_EVENT configuration bit may conflict with the CONFIG\_0\_PLC\_ERROR\_MSG, CONFIG\_0\_PLC\_DIAGNOSTICS\_MSG, FUNC\_1\_PLC\_DELAY\_CODE and FUNC\_1\_PLC\_SCRAP\_CODE configuration bits if enabled at the same time.

## Version Level 104

This Version Level adds wConfiguration03. The configuration bits for wConfiguration03 in Version Level 103 are defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_4_ENABLE_PLC_AXIS_STRUCTURES	0x0001	wConfiguration04
!!!Reserved for future use	0xFFFE	wConfiguration04

### PLC Y-Axis and Machine Setup

Configuration Bit: CONFIG\_4\_ENABLE\_PLC\_AXIS\_STRUCTURES (0x0001)

Starting address in PLC memory specified by: wPLCAxisStructuresAdd

User Data Size: wPLCAxisStructuresCount \* SizeInWords(AxisStructure) words

Command Used: MODBUS (16) Write Holding Registers

MODBUS (3) Read Holding Registers

The wPLCAxisStructureVersion field is used to determine the definition of the Axis Structure and allow the structure to be modified to add new features. It should be set to 100 or 101 depending on which version of the Axis Structure is used.

The wPLCAxisStructuresAdd should be set to a value indicating which register the array of AxisStructures begins.

Axis Structure Version 100:

Size	Field
WORD	wConfig
DWORD	dNewProgPosition
WORD	wNewCommand
DWORD	dCurrentPosition
WORD	wComStatus
WORD	wState

Axis Structure Version 101:

```
//This structure is identical to Axis Structure Version 100 except it adds two extra fields
//at the end. The two extra fields were added to allow an Axis that is positioned to
//discrete locations based on limit switches to pass the allowed positions to the XL200.
//The ONLY positions that should be in this array are positions that the XL200's shape
//programming can use.
```

Size	Field
WORD	wConfig
DWORD	dNewProgPosition
WORD	wNewCommand
DWORD	dCurrentPosition
WORD	wComStatus
WORD	wState
WORD	wNumberOfDiscretePositions
WORD	wDiscretePosArrayAddress

The wConfig field has 16 bits used to let the XL200 know which of the supported functions the PLC will be using. There are four functions, positioning, jogging, referencing, and enabling. All Axis' will support positioning. Jogging,

referencing and enabling are optional. These are the bits to be used to enable an optional function for each individual axis:

Function	Bit Value (hex)
PLC_AXIS_STRUCTURE_JOGGING	0x0001
PLC_AXIS_STRUCTURE_REFERENCE_TO_SWITCH	0x0002
PLC_AXIS_STRUCTURE_REFERENCE_TO_POSITION	0x0004
PLC_AXIS_ENABLING	0x0008
!!Reserved	0xFFFF

The dNewProgPosition field is where the XL200 will send new positions that the Axis should position itself to. The field will be sent the position in thousands of an inch. For example, the position 3.5" will be sent as 3500. The PLC should monitor the wNewCommand to determine when a new position has been sent and when to use the new position. If the PLC\_AXIS\_STRUCTURE\_REFERENCE\_TO\_POSITION function is enabled then the PLC should use this position as a reference position when the reference command is received. This position will tell the PLC where the axis is currently at.

The wNewCommand field is used to notify the PLC of new commands. The PLC should clear out the field after it has recognized the new command. Commands are defined as follows:

Command	Value(decimal)
PLC_MOVE_TO_NEW_POSITION	1
PLC_START_JOGGING_FORWARD	2
PLC_START_JOGGING_REVERSE	3
PLC_STOP_JOGGING	4
PLC_BEGIN_REFERENCE_FUNCTION	5
PLC_ENABLE_AXIS	6
PLC_DISABLE_AXIS	7
PLC_STOP_ALL_COMMANDS	8

Any new command should terminate the previous, if any, command. For instance, the PLC\_STOP\_ALL\_COMMANDS will be used to indicate the PLC should stop jogging an axis forward or reverse or stop positioning.

The dCurrentPosition field should be set by the PLC to the current position of the Axis in thousandths of an inch. For example, the position 3.5" inches should be indicated with the value 3500.

The wComStatus field should be used by the PLC to indicate a status of the Axis. The following values indicate the current status:

Status	Value(decimal)
PLC_NO_COMMAND	0
PLC_MOVING_TO_NEW_POSITION	1
PLC_NEW_POSITION_REACHED	2
PLC_POSITION_OUT_TOLERANCE	3
PLC_JOGGING_FORWARD	4
PLC_JOGGING_REVERSE	5
PLC_JOGGING_STOPPED	6
PLC_REFERENCING	7
PLC_REFERENCING_FINISHED	8
PLC_ENABLEING	9
PLC_ENABLE_FINISHED	10

PLC_DISABLEING	11
PLC_DISABLE_FINISHED	12
PLC_COMMAND_ERROR	13

The PLC should initialize wComStatus to PLC\_NO\_COMMAND on power up. PLC\_MOVING\_TO\_NEW\_POSITION should be used after a new position is recognized and the axis is being positioned. PLC\_NEW\_POSITION\_REACHED should be used when the axis has stopped after a position command and has stopped within tolerance. PLC\_POSITION\_OUT\_TOLERANCE should be used to indicate that the axis is stopped after a position command but has stopped out of tolerance. PLC\_JOGGING\_FORWARD should be used while the axis is jogging forward. PLC\_JOGGING\_REVERSE should be used when the axis is jogging in reverse. PLC\_JOGGING\_STOPPED should be used when the axis is stopped after jogging. PLC\_REFERENCING should be used when the axis is referencing. PLC\_REFERENCING\_FINISHED should be used when the axis is finished referencing. PLC\_ENABLE\_FINISHED should be used when the axis finishes enabling without error. PLC\_DISABLE\_FINISHED should be used when the axis finishes disabling without error. PLC\_COMMAND\_ERROR should be used if a jog, reference, enable, disable or position commands cannot be executed. The PLC should use the error or diagnostic function, or some other means, to indicate to the XL or operator why the command cannot be executed.

The wState field should be used by the PLC to indicate things about the state of the Axis. Each bit in the field is used to indicate a particular state. The following states are defined:

State	Bit Value(hex)
PLC_ENABLED	0x0001
!!!Reserved for future use	0xFFFE

If the PLC\_AXIS\_ENABLING function is enabled the PLC should set the PLC\_ENABLED bit when the drive is enabled and clear it when it is not enabled.

The wNumberOfDiscretePositions field is only available beginning in the version level 101 axis structure. It has a maximum value of 10. If wNumberOfDiscretePositions is ZERO then the XL200's shape programming function can use any position within the travel range of the axis. If wNumberOfDiscretePositions is NON-zero, wDiscretePosArrayAddress must point to an array of DWORD positions that have the same numeric format as the dCurrentPosition field. The array must have the number of elements specified in the wNumberOfDiscretePositions field.

*IMPORTANT NOTE: The extra fields in the version level 101 axis structure were added to let the XL200's shape programming algorithm know if the axis only supports two or more discrete positions within the travel range rather than any position within the travel range. If the shape programming routine is not allowed to use a position it should NOT be included in the array.*

## Version Level 105

This Version Level adds wConfiguration05. The configuration bits for wConfiguration05 in Version Level 105 are defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_MATERIAL_CHANGE_EVENTS	0x0001	wConfiguration05
CONFIG_5_PCODE_CHANGE_EVENTS	0x0002	wConfiguration05
!!!Reserved for future use	0xFFFD	wConfiguration05

### PLC Function Request Coils

User Data Size 16 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils added for Version Level 105 are:

Enumeration	Coil (0 based index)
FUNC_5_PCODE_AXIS_CHANGE_EVENT	9
FUNC_5_MATERIAL_AXIS_CHANGE_EVENT	10
!!Reserved	11 - 15

### Multi-Axis Material Change Event

Configuration Bit: CONFIG\_5\_MATERIAL\_CHANGE\_EVENTS (0x1)  
 Starting address in PLC memory specified by: wPLCAxisMaterialChangeAdd  
 User Data Size: 11 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the Material code that the PLC should set the machine up for and notify the PLC when the material code has been sent. The Material code string will be written to the address stored in wPLCAxisMaterialChangeAdd. The first word in the data indicates the length of the string.

When the user selects that the machine should be setup as a result of a material change the XL200 will send the material code string to the PLC and then set the FUNC\_5\_MATERIAL\_AXIS\_CHANGE\_EVENT function coil. When this function coil is set the PLC should read and buffer the material code string, if required. The material code string should be considered valid **only** on the first scan that the FUNC\_5\_MATERIAL\_AXIS\_CHANGE\_EVENT function coil is set. The PLC should use the material code string and then clear out the FUNC\_5\_MATERIAL\_AXIS\_CHANGE\_EVENT function coil.

*IMPORTANT NOTE: CONFIG\_5\_MATERIAL\_CHANGE\_EVENTS is only valid when CONFIG\_4\_ENABLE\_PLC\_AXIS\_STRUCTURES is also set.*

*IMPORTANT NOTE2: The PLC must be able to set the machine up for the Material change in a matter of a few seconds and notify the operator of problems via HMI or other PLC diagnostic message methods described within this document.*

### Multi-Axis Product Code Change Event

Configuration Bit: CONFIG\_5\_PCODE\_CHANGE\_EVENTS (0x2)  
 Starting address in PLC memory specified by: wPLCAxisPCodeChangeAdd  
 User Data Size: 11 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the Product code that the PLC should set the machine up for and notify the PLC when the product code has been sent. The Product code string will be written to the address stored in wPLCAxisPCoDeChangeAdd. The first word in the data indicates the length of the string.

When the user selects that the machine should be setup as a result of a product change the XL200 will send the material code string to the PLC and then set the FUNC\_5\_PCODE\_AXIS\_CHANGE\_EVENT function coil. When this function coil is set the PLC should read and buffer the product code string, if required. The product code string should be considered valid **only** on the first scan that the FUNC\_5\_PCODE\_AXIS\_CHANGE\_EVENT function coil is set. The PLC should use the product code string and then clear out the FUNC\_5\_PCODE\_AXIS\_CHANGE\_EVENT function coil.

*IMPORTANT NOTE: CONFIG\_5\_PCODE\_CHANGE\_EVENTS is only valid when CONFIG\_4\_ENABLE\_PLC\_AXIS\_STRUCTURES is also set.*

*IMPORTANT NOTE2: The PLC must be able to set the machine up for the Product change in a matter of a few seconds and notify the operator of problems via HMI or other PLC diagnostic message methods described within this document.*

## Version Level 106

This version adds two new fields to the Current and Next Item Information structures that are defined in Version Level 101. The new fields are ItemID and ParentID. The ItemID field is unique for every item in the controller's memory. It can be used to determine when the Current or Next Item Information fields belong to a new item. The ParentID fields can be used to determine when the Current and Next Item fields belong to different orders.

### Receive Current Item Information

Configuration Bit: CONFIG\_0\_CURRENT\_ITEM\_INFO (0x0020)  
Starting address in PLC memory specified by: wCurrentItemInfoAddress  
User Data Size: 10 or 21 words  
Command Used: MODBUS (16) Write Holding Registers

There are two different structures depending on what Control System is connected to the PLC. For Version Level 106 these structures are called the str106ItemData structure.

All other information about these structures are documented in version level 101.

### Receive Next Item Information

Configuration Bit: CONFIG\_0\_NEXT\_ITEM\_INFO (0x0040)  
Starting address in PLC memory specified by: wNextItemInfoAddress  
User Data Size: 10 or 21 words  
Command Used: MODBUS (16) Write Holding Registers

There are two different structures depending on what Control System is connected to the PLC.

There are two different structures depending on what Control System is connected to the PLC. For Version Level 106 these structures are called the str106ItemData structure.

All other information about these structures are documented in version level 101.



Enabling this feature causes the XL200 to read a value from the PLC's memory during each communication scan. This value will be logically OR'ed with the XL200 standard inputs to effectively allow the PLC to activate a controller input without being hard wired to the terminal strip. The data will be transferred as data words with one bit representing the state of each input (logic 1 = ON, logic 0 = OFF). Input 1 corresponds to the least significant bit in the array. For example, a value of 0x0008 indicates that the PLC wishes to activate input 4 on the XL200.

*IMPORTANT NOTE: Input values are read during each communication scan when this feature is enabled. Each input will be active on the XL200 controller for the duration of time that the corresponding bit is set in the PLC's memory.*

## Version Level 109

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 109 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_REMOTE_CONTROL_FUNCTIONS	0x0008	wConfiguration05

### PLC Remote Control Functions

Configuration Bit: CONFIG\_5\_REMOTE\_CONTROL\_FUNCTIONS (0x0008)

Starting address in PLC memory specified by: wRemoteControlStructAdd

User Data Size: SizeInWords(RemoteControlStructure) words

Command Used: MODBUS (16) Write Holding Registers  
 MODBUS (3) Read Holding Registers

The wRemoteControlStructVer field is used to determine the definition of the Axis Structure and allow the structure to be modified to add new features. It should be set to 100 for now until future modifications occur.

The wRemoteControlStructAdd should be set to a value indicating which register the RemoteControlStructure begins.

RemoteControlStructure Version 100:

Size	Field
WORD	wComand
WORD	wResult
DWORD	dCurrentPosition
WORD	wDie
DWORD	dNewProgPosition

*IMPORTANT NOTE: This feature requires and relies on the extended MODBUS only IO capabilities added in version level 108. Each XL200 model will have an extended input that places the XL200 in a PLC remote control mode and an output that indicates the XL200 is in the PLC remote control mode.*

Each XL200 model will have a PLC remote control input and a PLC remote control output. The XL200 will read the RemoteControlStructure only when the PLC remote control input is ON and its PLC remote control output is on. See the XL200 documentation for the model you intend to control for a definition of which input and output to use.

The wComand field is used by the PLC to let the XL200 know which of the supported functions the PLC would like the XL200 to perform. The field will be cleared out once the XL200 has read out the command. Valid commands are:

Command	Value (decimal)
REMOTE_CONTROL_REQUEST_DIE_POSITION	1
REMOTE_CONTROL_MOVE_DIE_ABSOLUTE	2
REMOTE_CONTROL_MOVE_DIE_RELATIVE	3

The wDie field is used to specify which die accelerator should be positioned. Dies are defined as follows:

Die	Value (decimal)
DIE_1	1
DIE_2	2

The dNewProgPosition field is where the XL200 will read the relative or absolute position that the XL should position its' Die to. The field will be interpreted as thousands of an inch. For example, the position 3.5" should be sent as 3500.

The dCurrentPosition field will be set by the XL200 to the current position of the specified die in thousandths of an inch. For example, the position 3.5" inches will be indicated with the value 3500. This field can be requested with the REMOTE\_CONTROL\_REQUEST\_DIE\_POSITION command and it will be updated by the XL200 while executing any of the other move commands until the PLC remote input to the controller is turned OFF.

The wResult field will be used by the XL200 to indicate the status of the command. The following values indicate the current status:

Result	Value (decimal)
UNDEFINED	0
WORKING	1
SUCCESS	2
FAILED	3

The PLC should initialize wResult to UNDEFINED prior to setting the wCommand field. WORKING indicates that the command has been recognized and is currently being executed.

SUCCESS indicates the command has been recognized and has been processed successfully.

FAILED indicates the command has been recognized but there was an error while processing it.

## Version Level 110

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 110 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_REMOTE_TOOL_SELECT	0x0010	wConfiguration05

### PLC Function Request Coils

User Data Size 16 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 110 are:

Enumeration	Coil (0 based index)
FUNC_5_TOOL_SELECT_CHANGED	12

### PLC Remote Tool Select

Configuration Bit: CONFIG\_5\_REMOTE\_TOOL\_SELECT (0x0010)  
 Starting address in PLC memory specified by: wRemoteToolSelectAdd  
 User Data Size: 1 word  
 Command Used: MODBUS (16) Write Holding Registers  
 MODBUS (3) Read Holding Registers

The wRemoteToolSelectAdd field should be set to a value indicating which word register contains the Remote Tool Select value. The Remote Tool Select value shall be a value 0-999.

If the CONFIG\_5\_REMOTE\_TOOL\_SELECT bit is enabled every Manual Press Operation should conform to the following sequence of events:

1. Set the register indicated by wRemoteToolSelectAdd to the desired tool.
2. Turn on the FUNC\_5\_TOOL\_SELECT\_CHANGED function coil.
3. Wait for the FUNC\_5\_TOOL\_SELECT\_CHANGED to be cleared by the XL.
4. Turn on the XL Manual Press input to initiate the press cycle
5. Wait for the press cycle to complete before allowing a new sequence to start.

**IMPORTANT NOTE:** Several XL models have multiple Manual Press inputs. Only the FIRST Manual Press input will monitor this value. All other presses will use the Tool Select parameter or tool number specified for that input.

## Version Level 111

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 111 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_AUX_WELD_DETECTS_ENABLED	0x0020	wConfiguration05

### PLC Function Request Coils

User Data Size 16 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 111 are:

Enumeration	Coil (0 based index)
FUNC_5_TEST_FOR_BAR_END	13

### Stick Fed Auxiliary Weld Detect Sensors

Configuration Bit: CONFIG\_5\_AUX\_WELD\_DETECTS\_ENABLED (0x0020)  
 Starting address in PLC memory specified by: wAuxWeldDetects  
 User Data Size: 2 word  
 Command Used: MODBUS (3) Read Holding Registers

The wAuxWeldDetects field should be set to a value indicating the number of Auxiliary Weld Detects configured. The XL currently supports up to 4 Auxiliary Weld Detect sensors for a total of five, including the main Weld Detect sensor. The main Weld Detect sensor must be hard wired. Except in a special case described below, the state of the main Weld Detect sensor is simply passed through the PLC.

The wAuxWeldDetectRegAdd field should be set to a value indicating which word register will contain an index 1-5 indicating which Weld Detect sensed the end of the stick. We will call this the Weld Detect Index Register. The PLC should initialize this index to ZERO on powerup.

Index 1 represents the main Weld detect. Indexes 2-5 represent auxiliary detectors that are progressively further away from the shear.

If the function coil FUNC\_5\_TEST\_FOR\_BAR\_END is set, the PLC should monitor all of the Weld Detect Inputs for a STICK\_PRESENT to STICK-NOT-PRESENT sensor transition which indicates that the end of a stick has passed by. The index of the input that sensed the transition must be written to the Weld Detect Index Register and the FUNC\_5\_TEST\_FOR\_BAR\_END function coil must be cleared. If it was index 2-5 that found the end of the stick, then the main Weld Detect input on the XL must be held in the WELD-DETECTED state for 0.5 seconds and then it can go back to the state driven by its sensor.

If the function coil FUNC\_5\_TEST\_FOR\_BAR\_END is NOT set, the state of main Weld Detect sensor is still forwarded to the hard wired, main Weld Detect input.

If Extended IO is enabled, the status of all configured Weld Detect sensors must be updated by the PLC in the Extended IO registers.

*IMPORTANT NOTE: The PLC should make use of high-speed inputs, interrupts or any other available feature to improve the response time of triggering the hardwired weld detect input on the XL any time an edge is detected.*

## Version Level 112

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 112 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_REMOTE_SETUP_CONFIG_SELECT	0x0040	wConfiguration05

### PLC Remote Setup Configuration Select

Configuration Bit CONFIG\_5\_REMOTE\_SETUP\_CONFIG\_SELECT (0x0040)  
 Starting address in PLC memory specified by: wRemoteSetupConfigSelAdd  
 User Data Size: 1 word  
 Command Used: MODBUS (3) Read Holding Registers

The wRemoteSetupConfigSelAdd field should be set to a value indicating which word register contains the Remote Setup Configuration Select value. The Remote Setup Configuration value shall be a value 1-999.

If the CONFIG\_5\_REMOTE\_SETUP\_CONFIG\_SELECT bit is enabled Run Mode Operation should conform to the following sequence of events:

1. Set the register indicated by wRemoteSetupConfigSelAdd to the desired setup configuration.
2. Initiate the desire run mode operation.

*IMPORTANT NOTE: If setups are Locked out by eclipse or if the configuration ID is not defined, the desired run mode will not execute, and the controller will report an error.*

## Version Level 113

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 113 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_REMOTE_RUN_VELOCITY_POT	0x0080	wConfiguration05

### PLC Function Request Coils

User Data Size 16 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 113 are:

Enumeration	Coil (0 based index)
FUNC_5_REMOTE_RUN_VEL_POT_CHANGE	14

### PLC Remote Run Velocity Pot

Configuration Bit: CONFIG\_5\_REMOTE\_RUN\_VELOCITY\_POT (0x0080)  
 Starting address in PLC memory specified by: wRemoteRunVelocityPotAdd  
 User Data Size: 1 word  
 Command Used: MODBUS (16) Write Holding Registers  
 MODBUS (3) Read Holding Registers

The wRemoteRunVelocityPotAdd field should be set to a value indicating which word register contains the Remote Run Velocity Potentiometer value. The Remote Run Velocity Potentiometer value shall be a value 0-1000. A value of 1 represents a velocity percentage of 0.1%. A value of 999 represents a percentage of 99.9%. A value of 1000 is a request for full speed.

This feature is only supported for Closed Loop Feed to Stop controllers while in the Run Mode when the Slow Run input is OFF. It allows the PLC to select a percentage of the Maximum Velocity setting. It has no effect on the Slow Run Velocity setting because it is intended to be used instead of this setting.

If the CONFIG\_5\_REMOTE\_RUN\_VELOCITY\_POT bit is enabled every Velocity Pot change Operation should conform to the following sequence of events:

1. Change a Local Velocity Pot Holding Register (Local Val). (This can be done any time)
2. Wait for FUNC\_5\_REMOTE\_RUN\_VEL\_POT\_CHANGE function coil to be OFF
3. Set the register indicated by wRemoteRunVelocityPotAdd to the Local Val.
4. Turn on the FUNC\_5\_REMOTE\_RUN\_VEL\_POT\_CHANGE function coil.

*IMPORTANT NOTE: For Real Time Performance issues the FUNC\_5\_REMOTE\_VEL\_POT\_CHANGE function coil should only be set when the Local Value has changed from the value residing at wRemoteRunVelocityPotAdd. The XL will clear FUNC\_5\_REMOTE\_VEL\_POT\_CHANGE when it has read the new value.*





## Version Level 116

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 116 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_ENABLE_SECOND_LINE_ENCODER	0x0800	wConfiguration05

### Enable Second Line Encoder Port

Configuration Bit: CONFIG\_5\_ENABLE\_SECOND\_LINE\_ENCODER (0x0800)

This feature also requires that a MODBUS input be used to select which of the two-line encoder ports are active.

The second line encoder port will not be enabled on a Single Encoder Closed Loop Feed to Stop line.

## Version Level 117

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 117 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_MULTI_AXIS_POSITION_FUNCTIONS	0x1000	wConfiguration05

### PLC Function Request Coils

User Data Size 32 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 117 are:

Enumeration	Coil (0 based index)
FUNC_14_MULTI_AXIS_POSITION_FUNC_REQUEST	20

### PLC Remote Control Functions

Configuration Bit: CONFIG\_5\_MULTI\_AXIS\_POSITION\_FUNCTIONS (0x1000)

Starting address in PLC memory specified by: wMAPositioningStructAdd

User Data Size: SizeInWords(MAPositioningStructure) words

Command Used: MODBUS (16) Write Holding Registers  
MODBUS (3) Read Holding Registers

The wMAPositioningStructVer field is used to determine the definition of the Axis Structure and allow the structure to be modified to add new features. It should be set to 100 for now until future modifications occur.

The wMAPositioningStructAdd should be set to a value indicating which register the MAPositioningStructure begins.

MAPositioningStructure Version 100:

Size	Field
WORD	wComand
WORD	wAxisId
DWORD	dNewProgPosition
WORD	wResult
WORD	wPosStatus
DWORD	dCurrentPosition

The AMS Controller will read the MAPositioningStructure only when the PLC sets the function coil FUNC\_14\_MULTI\_AXIS\_POSITION\_FUNC\_REQUEST ON. The PLC must have initialized the MAPositioningStructure fields prior to setting the function coil. Once the AMS controller has recognized the command it will clear the function coil.

The wComand field is used by the PLC to let the AMS controller know which of the supported functions the PLC would like it to perform. The field will be cleared out once the AMS Controller has read out the command. Valid commands are:

Command	Value (decimal)
MULTI_AXIS_POSITION_REQUEST_MOVE	1
MULTI_AXIS_POSITION_REQUEST_STATUS	2

The wAxisId field is used to specify which Axis the command should be executed on. The AXIS must not be of type Tool or one of the Machine Setup Axis Types. It must be of Type PLC or the command will Fail.

The dNewProgPosition field is where the AMS Controller will read the position that it should position the Axis to. The field will be interpreted as thousands of an inch. For example, the position 3.5” should be sent as 3500.

The dCurrentPosition field will be set by the AMS Controller to the current position of the specified Axis in thousandths of an inch. For example, the position 3.5” inches will be indicated with the value 3500. This field can be requested with the MULTI\_AXIS\_POSITION\_REQUEST\_STATUS command and it will be updated by the AMS Controller while positioning during a MULTI\_AXIS\_POSITION\_REQUEST\_MOVE.

The wPosStatus field is updated by the AMS controller at the same time as the dCurrentPosition field. The following values indicate the current Status:

Status	Value (decimal)
UNDEFINED	0
STOPPED_IN_POSITION	1
STOPPED_OUT_OF_POSITION	2
MOVING	3

The wResult field will be used by the AMS Controller to indicate the status of the command. The following values indicate the current status:

Result	Value (decimal)
UNDEFINED	0
WORKING	1
SUCCESS	2
FAILED	3

The PLC should initialize wResult to UNDEFINED prior to setting the wCommand field. WORKING indicates that the command has been recognized and is currently being executed.

SUCCESS indicates the command has been recognized and has been processed successfully.

FAILED indicates the command has been recognized but there was an error while processing it.

## Version Level 118

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 118 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_INCREASE_QTY_FUNCTIONS	0x2000	wConfiguration05

### PLC Function Request Coils

User Data Size 32 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 118 are:

Enumeration	Coil (0 based index)
FUNC_18_INCREASE_QUANTITY_REQUEST	21
FUNC_18_INCREASE_QUANTITY_CANCEL	22

PLC Increase Quantity

Configuration Bit: CONFIG\_5\_INCREASE\_QTY\_FUNCTIONS (0x2000)  
 Starting address in PLC memory specified by: wIncQuantityStructVer  
 User Data Size: SizeInWords(IncQuantityStructure) words  
 Command Used: MODBUS (1) Read Coils  
                   MODBUS (5) Write Single Coil  
 MODBUS (16) Write Holding Registers  
                   MODBUS (3) Read Holding Registers

The wIncQuantityStructVer field is used to determine the definition of the Increase Quantity Structure and allow the structure to be modified to add new features. It should be set to 100 for now until future modifications occur.

The wIncQuantityStructAdd should be set to a value indicating which register the IncQuantityStructure begins.

IncQuantityStructure Version 100:

Size	Field
WORD	wAddQuantity
WORD	wScrapCode
WORD	wTotalQuantity

The AMS Controller will read the IncQuantityStructure only when the PLC sets the function coil FUNC\_18\_INCREASE\_QUANTITY\_REQUEST ON. The PLC must have initialized the IncQuantityStructure fields prior to setting the function coil. Once the AMS controller has recognized the command it will clear the function coil.

The wAddQuantity field should be used to request (additive) a quantity of parts to be incremented. The Scrap Code (reason the parts being added are required) can be set and will be used if the controller is configured to use Scrap Codes.

To allow the Total Increase Quantity request value to be displayed on an HMI and countdown to zero, just like on the AMS screen, the wTotalQuantity field will be updated by the AMS controller to show the total of Increase Quantity parts that have been requested but not completed. This will be done on a periodic basis as the controller updates its displayed value.

PLC Increase Quantity Cancel

Configuration Bit: CONFIG\_5\_INCREASE\_QTY\_FUNCTIONS (0x2000)  
 Command Used: MODBUS (1) Read Coils  
                   MODBUS (5) Write Single Coil

The AMS Controller will monitor the function coil FUNC\_18\_INCREASE\_QUANTITY\_CANCEL. When the controller sees the function ON it will cancel the Increase Quantity and then clear then clear the function coil in acknowledgment.

## Version Level 119

This Version Level uses a reserved bit from wConfiguration05. The bit for wConfiguration05 in Version Level 119 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_5_ITEM_DATA_REQUEST	0x4000	wConfiguration05
CONFIG_5_SPECIFY_STRING_FIELD_SIZE	0x8000	wConfiguration05

### PLC Function Request Coils

User Data Size 32 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 119 are:

Enumeration	Coil (0 based index)
FUNC_19_ITEM_NEXT_REQUEST	23
FUNC_19_ITEM_PREVIOUS_REQUEST	24

This functionality is built on top of the Current or Next Item functionality of Version Level 101 and 106. If CONFIG\_5\_ITEM\_DATA\_REQUEST is enabled, CONFIG\_0\_CURRENT\_ITEM\_INFO or CONFIG\_0\_NEXT\_ITEM\_INFO must also be enabled and all related Version Level 106 and 101 functionality configured appropriately.

CONFIG\_1\_CURRENT\_ITEM\_USER1 and CONFIG\_1\_CURRENT\_ITEM\_USER2 bits in wConfiguration01 will be used to determine if the User1 and User2 Item User field Data will also be requested.

In the Version Level 119 portion of the configuration the following fields must be configured:

- wSearchItemAdd specifies where in memory the str106ItemData structure is reserved to store the requested item data.
- wSearchItemUser1Add specifies where in memory the User1 field data is stored for the requested item data, if enabled.
- wSearchItemUser2Add specifies where in memory the User2 field data is stored for the requested item data, if enabled.
- wldSearchAdd specifies where to read the ID to use for the Next or Previous item search.

In order to perform the Item Data Request, the DWORD ItemID of a known item must be written to the address specified by wldSearchAdd. The PLC must set either the FUNC\_19\_ITEM\_NEXT\_REQUEST or the FUNC\_19\_ITEM\_PREVIOUS\_REQUEST function coil and wait for the AMS controller to clear it. When the AMS controller sees the request, if the ItemID is found, the AMS controller will use the Next or Previous Item to fill Item Data into the memory addresses specified in the Version Level 119 configuration. The returned ItemID will reflect the result of the search. If found and configured, the User field data will also be filled for the item and then the function coil will be cleared, indicating that all item data fields have been fully populated with the result of the search. Repetitive searches can be executed to retrieve data for every item within an order.

Returned ItemID Value	Result
0	Searched Item ID was valid but there is No Next or Previous Item.
1	An Invalid ItemID was provided for the search

Any other value	Searched Item ID was valid, this is the ItemID of the new item that was found in the search
-----------------	---

Item Search Information

Configuration Bit: CONFIG\_5\_ITEM\_DATA\_REQUEST (0x4000)  
 Starting address in PLC memory specified by: wSearchItemAdd  
 User Data Size: 10 or 21 words  
 Command Used: MODBUS (16) Write Holding Registers

There are two different structures depending on what Control System is connected to the PLC. All information about this structure is documented in version level 106.

*IMPORTANT NOTE: The FUNC\_19\_ITEM\_NEXT\_REQUEST or the FUNC\_19\_ITEM\_PREVIOUS\_REQUEST function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Item Search User 1 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ITEM\_USER1 (0x0800)  
 Starting address in PLC memory specified by: wSearchItemUser1Add  
 User Data Size: 11 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current item User Field 1. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: The FUNC\_19\_ITEM\_NEXT\_REQUEST or the FUNC\_19\_ITEM\_PREVIOUS\_REQUEST function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Item Search User 2 Field

Configuration Bit: CONFIG\_1\_CURRENT\_ITEM\_USER2 (0x1000)  
 Starting address in PLC memory specified by: wSearchItemUser2Add  
 User Data Size: 11 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current item User Field 2. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: The FUNC\_19\_ITEM\_NEXT\_REQUEST or the FUNC\_19\_ITEM\_PREVIOUS\_REQUEST function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Configure String Field Sizes

In prior Version Levels a vulnerability existed. The PLC program had no way to protect itself from newer releases of AMS Controller software that may support larger field sizes in the future. During the life of the MODBUS integration feature, the field sizes have never increased but this vulnerability will still be addressed to plan for future enhancements.

If CONFIG\_5\_SPECIFY\_STRING\_FIELD\_SIZE is set in wConfiguration05 the following fields in Version Level 119 will be used by the PLC to indicate how much memory has been allocated for each string type. Otherwise, the default field size for each string type will be used.

To avoid future surprises, it is highly suggested to allocate more memory than the default or the capabilities of the latest AMS controller firmware for each string type. At the time of this writing, the AMS Controller capabilities match the default values.

The memory allocation requires one WORD to hold the length of the string data. The remaining WORDS each hold up to two characters.

String Type	Variable	Default Value in WORDS
Order Number	wOrderFieldSize	11 (20 characters)
Material Code	wMaterialFieldSize	11 (20 characters)
Product Code	wPCoDeSize	11 (20 characters)
User Field (Order and Item)	wUserFieldSize	16 (30 characters)



	the first Setup in the Settings Array. Each subsequent coil is for the subsequent Setup in the Settings Array.	
--	--	--

If enabled by the CONFIG\_120\_XL\_UPDATE\_COILS configuration bit, wAMSsettingsUpdateCoilsIndex must be initialized and reference a range of contiguous coils, one for each setting in the Settings Array. The parameters position within the array is used as an Index to determine which coil belongs to each parameter. When a parameter is modified, XL will turn ON its coil to inform the PLC that the parameter has been edited. It is the PLC's responsibility to turn the coil back OFF once it has read the new value.

If enabled by the CONFIG\_120\_PLC\_UPDATE\_COILS configuration bit, wPLCsettingsUpdateCoilsIndex must be initialized and reference a range of contiguous coils, called Auto Update Coils, one for each setting in the Settings Array. The parameters position within the array is used as an Index to determine which coil belongs to each parameter. When a parameter that has been configured as [Auto Update](#) has been modified by the PLC, the parameters Auto Update coil must be turned on by the PLC and then the FUNC\_120\_PLC\_PARAMETER\_UPDATE function coil should be turned ON. To preserve data coherency and prevent race conditions, the Auto Update Coils should only be turned ON by the PLC while the FUNC\_120\_PLC\_PARAMETER\_UPDATE coil is OFF. The XL will turn OFF the Auto Update Coils and then turn OFF the FUNC\_120\_PLC\_PARAMETER\_UPDATE Function Coil after it has read all of the parameter values that had their Auto Update Coil in the ON state.

**Settings Array (v100)**

Each Setup included in the wNumberOfSettings configuration requires an entry in the Settings array. Each entry in the array must have the following properties in order to be displayable and editable in the XL user interface.

Property	Description	Size
<a href="#">Display Name Address</a>	Address where NULL terminated display name resides. Max name length must be less than or equal to 29 characters.	WORD
<a href="#">Data Format</a>	Type(Integer, Short Integer, Decimal Fix Point, IEEE Float, List, String, IP(ipv4) Address), Width, Fraction, Units	DWORD
<a href="#">Data Address</a>	Address to read and write data value	WORD
<a href="#">Data Flags</a>	Defines properties like Read-Only, Auto-Update...etc about the setup	WORD
<a href="#">Data Upper Limit</a>	Maximum allowed value. Present for all but only used for numeric datatypes.	DWORD
<a href="#">Data Lower Limit</a>	Minimum allowed value. Present for all but only used for numeric datatypes.	DWORD

**Display Name Address**

Address where display name resides contained within an Array of 16 bit WORDs. The name must be unique.

The format of the string is expected to be as follows. The first WORD must contain the length of the string. Subsequent 16 bit WORDs will each contain two 8 bit ascii characters except the last WORD if the string has an odd number of characters. The length of the name must be less than or equal to 29 characters.

For Example, [N][CC][CC] represents a string in memory where N represents the number of characters in the Display Name, C represents a character in the string, [] represents a 16 bit WORD in memory. In the example N would be 4 because there are 4 characters in the string.

**List Strings**

One type of setup, the List type, presents an enumerated List of strings to select from. The selected string then represents a numeric value selection. If the setup is a List type, the list of strings must immediately follow the Display name in Memory. The length of each string within the list of strings must be 32 characters (Bytes) or less

The format of the List in memory is described as follows:

Width, from the [Data Format](#) represents the number of Strings in the list. Fraction, from the Data Format, represents the number of 16 bit WORDs required to read the entire list of strings. Just like the Display Name, the first WORD of each string in the list contains the length of the string. Each subsequent WORD contains up to two 8 bit ascii characters of the string.

Here is an example. If Width is 2 and Fraction is 7 there are two strings in the list and 7 words represent the length of the data to read in the list. The data of subsequent strings immediately follow in the next WORD after the WORD containing the last character of the preceding string. In this example [L][CC][CC][CC][L][CC][CC] represents two list strings where [ ] represents a WORD, L represents the length of the string and CC represents space for two ascii characters. A string with an odd number of characters still requires a full 16 bit WORD to contain the last character. Combining the Example for the [Display Name Address](#) and the above List String example produces data in the format [N=4][CC][CC][L=6][CC][CC][CC][L=4][CC][CC].

**Data Format**

The Data Format field contains four 8 bit subfields listed below in order of Most Significant to Least Significant within the two WORDs that make up the 32 bit DWORD. It is up to the PLC programmer to populate the 8 bit bytes in the correct order to be returned to the XL correctly.

Field	Description
<a href="#">Type</a>	Describes Data Type of Setup
<a href="#">Width</a>	Depending on Data Type, Limits or specifies Size
<a href="#">Fraction</a>	Depending on Data Type, Limits or specifies Size
<a href="#">Units</a>	Numeric Fields may have units that represent Length, Velocity...etc.

**Type**

See [Data Address](#) for explanation of data size for each Type and explanation of numeric ranges and data format for each Type.

Enumeration	Data Type
0	Undefined
1	Integer
2	Short Integer
3	Decimal Fixed Point
4	IEEE Floating Point
5	List

**Width**

Type	Use
Integer	Sets the number of characters used to display the Value on the XL, including the units string. See the Note: below.
Short Integer	Sets the number of characters used to display the Value on the XL, including the units string. See the Note: below.
Decimal Fixed Point	Sets the number of characters used to display the Value on the XL, including the units string. See the Note: below.
IEEE Float	Sets the number of characters used to display the Value on the XL, including the units string.
List	See <a href="#">List Strings</a> for an explanation.

**Note:** For the numeric data types it is best to set the Width to a value larger than is expected to be required, including the units string. If the value is using Length units, it is best to just set this to 14 as it will allow enough characters to edit and display the value in all of the operator selectable Numeric Display Formats as described in the Units section.

**Fraction**

Type	Use
Integer	NA
Short Integer	NA
Decimal Fixed Point	Sets how many decimal places the number will have to the right of the decimal when displayed or edited. In addition, it sets where the implied decimal place is in the PLC value which determines how the number is converted for display. For example, if the Data Value in the PLC is 32767 and the Fraction is 2 the controller will convert and display the value as 327.67. The same Data Value in the PLC and a Fraction of 3 results in three decimal places and the number is converted and displayed as 32.767
IEEE Float	Sets how many decimal places the number will have to the right of the decimal when displayed or edited.
List	Used as the size of the List in 16 bit WORDs. See <a href="#">List Strings</a> for an explanation.

**Units**

The XL controller has an Operator Preference setting for Numeric Display Format. This setting allows the operator to select between the English and Metric forms of the units below. The actual unit strings or symbols used are shown in parenthesis with the exception noted in the Length unit Type.

Enumeration	Units	Description
0	No units	
1	Length	Displays in any of the XL200 Length formats depending on the operator preference. Inches ("), Feet Decimal Inch(0' 0.00"), Feet Fractional Inch (0' 0 0/0"), Millimeters(mm), Centimeters(cm) or Meters(M). <b>Note:</b> The compound Display Formats of Feet Decimal and Feet Fractional show an example of how the Feet and Inches units will be represented.
2	Velocity	Feet per Minute (FPM) or Meters per Minute (MPM)
3	Feet	Feet (Ft) or Meters(M)
4	Inches	Inches (") or Millimeters(mm)
5	Acceleration	Inches per Second Squared (In/S <sup>2</sup> ) or Centimeters per Second Squared (Cm/S <sup>2</sup> )
6	Seconds	Seconds (sec)
7	Percentage	Percentage (%)
8	Hertz	Hertz (HZ)
9	Volts	Voltage (Volts)
10	Days	Days (Days)
11	Minutes	Minutes (Min)
12	Pounds(Lbs)	Pounds (Lbs) or Kilograms (Kg)
13	Degrees(Angle)	Degrees (°)
14	Percentage per Second	Percentage per Second(%/Sec)



The wRemoteToolConfigSelAdd field should be set to a value indicating which word register contains the Remote Setup Configuration Select value. The Remote Setup Configuration value shall be a value 1-999.

If the CONFIG\_5\_REMOTE\_SETUP\_CONFIG\_SELECT bit is enabled Run Mode Operation should conform to the following sequence of events:

1. Set the register indicated by wRemoteToolConfigSelAdd to the desired setup configuration.
2. Initiate the desire run mode operation.

*IMPORTANT NOTE: If tools are Locked out by eclipse or if the configuration ID is not defined, the desired run mode will not execute and the controller will report an error.*

## Version Level 121

This Version Level adds one configuration bit to the Version Level 120 configuration structure. The Additional configuration bit for wConfigurationBits120 Bits is defined below:

Enumeration	Bit Value (hex)	Configuration Word
!!!Previously defined in Version Level 120	0x000F	wConfigurationBits120
CONFIG_120_PCODE_COMPARE_BYPASS	0x0010	wConfigurationBits120

### Product Code Comparison Bypass

Configuration Bit: CONFIG\_120\_PCODE\_COMPARE\_BYPASS

Enabling this mode disables the Product code comparisons that halt the queue or enforce setup or tool configurations. The settings controlling those features get forced to disabled and the settings get hidden from the user.

## Version Level 122

This Version Level adds one configuration bit to the Version Level 120 configuration structure. The Additional configuration bit for wConfigurationBits120 Bits is defined below:

Enumeration	Bit Value (hex)	Configuration Word
!!!Previously defined in Version Level 120 and 121	0x001F	wConfigurationBits120
CONFIG_120_FTS_SHEAR_HANDSHAKE	0x0020	wConfigurationBits120

The function coils used for Version Level 122 are:

Enumeration	Nickname	Coil (0 based index)
FUNC_122_XL_REQUEST_SHEAR	XL_RS	26
FUNC_122_PLC_OKS_SHEAR	PLC_OKS	27
FUNC_122_XL_SHEAR_ACTIVE	XL_SA	28
FUNC_122_PLC_SEES_SHEAR	PLC_SEES	29

### Feed to Stop Shear Handshake

Configuration Bit: CONFIG\_120\_FTS\_SHEAR\_HANDSHAKE (0x0020)

Enabling this feature requires the AMS controller to ask permission to fire the shear press. The PLC and the AMS controller will execute a handshake sequence defined below. This permission is only required for shear operations that are executed while in the Run mode (Automatic). Manual Shears do not require permission or the handshake process.

**Request Sequence**

Step	XL_RS	PLC_OKS	XL_SA	PLC_SEES	Description
1	ON	OFF	OFF	OFF	AMS Controller requests an automatic shear and waits for PLC_OKS
2	ON	ON	OFF	OFF	PLC is ready for the shear and turned on PLC_OKS
3	ON	ON	ON		AMS Controller started the Shear Process, firing the shear press.
4	ON	ON	ON	ON	After the Shear cycle is complete the AMS Controller looks for the PLC_SEES coil that indicates the PLC has seen the XL_SA coil.
5	OFF	OFF	OFF	OFF	The AMS Controller turns OFF all of the coils.

**Notes:**

- When entering and exiting the Run mode (Automatic) the AMS Controller will turn OFF all of the handshake coils.
- The AMS Controller will wait forever for the PLC\_OKS function coil as long as the Run mode is active.
- The AMS Controller will wait for the PLC\_SEES function coil for 500msec. An error will be displayed if the PLC does not respond in time.

**Version Level 123**

This Version Level uses a reserved bit from wConfigurationBits120. The bit for wConfigurationBits120 in Version Level 120 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_120_XL_LINE_SPEED	0x0040	wConfigurationBits120

*XL200 Line Speed*

Configuration Bit: CONFIG\_120\_XL\_LINE\_SPEED (0x0080)  
 Starting address in PLC memory specified by: wLineSpeedInchPerSecAdd  
 User Data Size: 1 word  
 Command Used: MODBUS (16) Write Holding Registers  
 MODBUS (3) Read Holding Registers

The wLineSpeedInchPerSecAdd field should be set to a value indicating which word register the XL200 Line Speed value should be written to. The line speed will be written as an integer in Inches/(100 Seconds). For example: 12.5 inches/second will be written as an integer as 1250.

The wLineSpeedUpdateRate field must also be configured with a value to indicate how often the XL200 should send the Line Speed to the PLC. This field defines how many times per second the Line Speed should be sent.

Note: Care should be taken not to use too large of a number in wLineSpeedUpdateRate. Using a larger number will leave less time to send and receive data for the remaining enabled features.

## Version Level 124

This Version Level uses a reserved bit from wConfigurationBits120. The bit for wConfigurationBits120 in Version Level 120 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_120_QA_TRIGGER	0x0080	wConfigurationBits120

### PLC Function Request Coils

User Data Size 32 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 124 are:

Enumeration	Coil (0 based index)
FUNC_124_QA_TRIGGER_REQUEST	30

### PLC Quality Audit (Custom List) Trigger

Configuration Bit: CONFIG\_120\_QA\_TRIGGER (0x0080)  
 Starting address in PLC memory specified by: wQATriggerSelectAdd  
 User Data Size: 1 word  
 Command Used: MODBUS (3) Read Holding Registers

The wQATriggerSelectAdd field should be set to a value indicating which Word register address contains the DWORD value for the List ID of the QA Record (Custom List) that the PLC wishes to trigger.

If the CONFIG\_5\_REMOTE\_TOOL\_SELECT bit is enabled, every desired Trigger should conform to the following sequence of events:

1. Set the DWORD value at the address indicated by wQATriggerSelectAdd to the desired List ID value.
2. Turn on the FUNC\_124\_QA\_TRIGGER\_REQUEST function coil.
3. The XL will search for the QA Record containing that List ID and Trigger it to be displayed on the XL operator screen.
4. Wait for the FUNC\_124\_QA\_TRIGGER\_REQUEST to be cleared by the XL before any further requests.

## Version Level 125

This version level adds two additional user Item User fields to the fields described in version level 101. Version level 101 was defined prior to the existence of two additional Item User fields, 3 and 4. It also allows Setup Configurations to be loaded prior to entering the run mode for cases where settings affect things like coil threading.

### PLC Function Request Coils

User Data Size 32 Coils

See the PLC Function Coils section for a description of Function Coils.

The function coils used for Version Level 124 are:

Enumeration	Coil (0 based index)
FUNC_125_REQUEST_SETUP_CONFIGURATION_SELECT	31

This Version Level uses reserved bits from wConfigurationBits120. The bits for wConfigurationBits120 in Version Level 120 is defined below:

Enumeration	Bit Value (hex)	Configuration Word
CONFIG_120_CURRENT_ITEM_USER3	0x0100	wConfigurationBits120
CONFIG_120_CURRENT_ITEM_USER4	0x0200	wConfigurationBits120
CONFIG_120_REQUEST_SETUP_CONFIG_SELECT	0x0400	wConfigurationBits120

Receive Item User 3 Field

Configuration Bit: CONFIG\_120\_CURRENT\_ITEM\_USER3 (0x0100)  
 Starting address in PLC memory specified by: wCurltmUser3Address  
 User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current item User Field 3. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

Receive Item User 4 Field

Configuration Bit: CONFIG\_120\_CURRENT\_ITEM\_USER4 (0x0200)  
 Starting address in PLC memory specified by: wCurltmUser4Address  
 User Data Size: 16 words  
 Command Used: MODBUS (16) Write Holding Registers

Enabling this feature tells the XL200 to send the current item User Field 4. The first word in the data indicates the length of the string.

*IMPORTANT NOTE: To help keep scan times to a minimum, the data will only be transferred when the data has changed state since the previous scan.*

*IMPORTANT NOTE2: The FUNC\_1\_SENDING\_CURRENT\_ORDER\_ITEM\_INFO function coil should be monitored when using this data. The function coil indicates that the data for the current order and item is in flux. You should wait until this bit is clear and/or double buffer the data to prevent using incomplete information.*

PLC Request Remote Setup Configuration Select

Configuration Bit CONFIG\_5\_REMOTE\_SETUP\_CONFIG\_SELECT (0x0040)  
 CONFIG\_120\_REQUEST\_SETUP\_CONFIG\_SELECT (0x0400)  
 Starting address in PLC memory specified by: wRemoteSetupConfigSelAdd  
 User Data Size: 1 word  
 Command Used: MODBUS (3) Read Holding Registers

The wRemoteSetupConfigSelAdd field should be set to a value indicating which word register contains the Remote Setup Configuration Select value. The Remote Setup Configuration value shall be a value 1-999.

This feature is different from the PLC Remote Setup Configuration Select in that the Setup Configuration can be requested to be loaded any time the Run Mode is Idle. The controller will attempt to load it right away rather than waiting until a run mode is selected.

If the CONFIG\_5\_REMOTE\_SETUP\_CONFIG\_SELECT bit is enabled Run Mode Operation should conform to the following sequence of events:

1. Set the register indicated by wRemoteSetupConfigSelAdd to the desired setup configuration.
2. Set the Function Coil FUNC\_125\_REQUEST\_SETUP\_CONFIGURATION\_SELECT.
3. The XL will Clear the Function Coil FUNC\_125\_REQUEST\_SETUP\_CONFIGURATION\_SELECT once it has read the wRemoteSetupConfigSelAdd register and has attempted to load the specified configuration.

*IMPORTANT NOTE: If setups are Locked out by eclipse or if the configuration ID is not defined, the desired configuration will not load and the controller will report an error.*

## Appendix A

**Document Revisions**

October 13, 2005	DT	Added stop bit setup parameter
October 19, 2005	DT	Added the Version Level 101 data structure.
October 25, 2005	DT	Attempted to clarify the differences and shared portions of the version level 100 and 101 structures. Added function coils CONFIG_1_SENDING_CURRENT_ORDER_ITEM_INFO and CONFIG_1_SENDING_NEXT_ORDER_ITEM_INFO to help keep the item and order fields in sync and prevent the PLC from using data that is in flux.
January 23, 2006	DT	Added the Version Level 102 data structure.
March 8, 2006	DT	Added comment to the external gag outputs option indicating that it will be ignored by the XL200 controller if the "E" expanded gag bit code option is enabled
April 12, 2006	DT	Begin design and documentation of Y-Axis and Machine Setup Movement
May 18, 2006	DT	Expand the design of the Y-Axis and Machine Setup Movement function
August 16, 2006	DT	Remove the unfinished design for Y-Axis and Machine Setup from the document. Added a new method for reporting Error and Diagnostics messages in Version Level 103.
August 21, 2006	DT	Add PLC Notification Event function.
September 15, 2006	DT	Added the design for Y-Axis and Machine Setup back into the document.
January 5, 2007	DT	Added a note to the PLC Configuration Data Structure section warning the user to initialize the entire structure on the first pass of the PLC program.
December 21, 2007	DT	Added Version Level 105 to allow for more Function Coils.
January 2, 2008	DT	Version Level 105 changed and finalized.
April 16, 2008	DT	Added support for XL220 HVAC controllers by defining the Item structure for an XL220 in Version Level 101
Nov 20, 2009	DT	Added Item and Parent ID's to Current and Next Item data structures in Version Level 106.
Feb 23, 2010	DT	Added Version Level 107 for a special custom Function
Feb 26, 2010	DT	Added Version Level 108 to specify how many extra words of Input and Output bits to send or receive from the controller
Aug 2, 2010	DT	Removed the PMP style language from the document.
April 7, 2011	DT	Added new fields to the Axis Structure to support shape programming. If an axis has flag switches as its feedback and shape programming is being used then these new fields should be used to let the shape programming function know which positions it can send the Axis to.
Nov 26, 2011	DT	Added Version Level 109 to allow remote control of Die Accelerators by the PLC.
Dec 17, 2011	DT	Fixed error in document concerning CONFIG_5_REMOTE_CONTROL_FUNCTIONS. Changed the value from 0x8000 to 0x8.
Dec 2, 2012	DT	Added Version Level 110 to allow Remote Tool Select.
Nov10, 2014	DT	Added Remote Setup Configuration Select.
Nov 16, 2014	DT	Added Remote Run Velocity Pot.
Jan 6, 2016	DT	Added Version Level 115 to allow PLC's to know when a CL FTS Controller is feeding material.
Apr 11, 2016	DT	Added Version Level 116 to enable a second line encoder port
Jan 6, 2017	DT	Added Version Level 117 to allow PLC to Position new PLC axis types.
Nov 13, 2017	DT	Fixed Typo in Version Level 117 configuration bit value.

Jan 5, 2018	DT	Added Split wrapper documentation to HVAC Part Type. No Version Level change was/is required.
June 11, 2018	DT	Added Increase Quantity and Increase Quantity Cancel Functions
Sept 19, 2018	DT	Added Version Level 119 to allow data for additional items within an order to be requested by the PLC.
Sept 24, 2018	DT	Fixed Document Typo on user field size. User fields have always been 16 words in size.
Sept 24, 2018	DT	Added table in Appendix A with decimal, octal and hex offsets for every entry in the Configuration Structure.
Sept 25, 2018	DT	Added String Field Size configuration to Version Level 119 prior to release.
Feb 4, 2019	DT	Fixed Typos in Remote Setup Configuration feature of Version Level 112.
Feb 12, 2019	DT	Added MODBUS Card Requirements
Feb 12, 2019	DT	Added Version Level 120 to allow editing of parameters in a PLC on the XL200 user interface. Removed and replaced all of the C language structure formatting and definitions with tables. Added a comprehensive table of Function Coils, each with their Version Levels. Also added a comprehensive table of Configuration Bits with their Version Levels and Bit fields.
Nov 18, 2019	DT	Added Version Level 121 for the Product Code Comparison Bypass Feature.
Nov 12, 2020	DT	Added Version Level 122 for the Feed to Stop Shear Handshake Feature
April 12, 2021	DT	Added Version Level 123 for the XL200 Line Speed feature.
Oct 7, 2021	DT	Added Version Level 124 for the Quality Audit (Custom List) Trigger feature.
April 11, 2022	DT	Added information about the AMS ID display in the AMS Controller MODBUS Diagnostic screen.
Nov 28, 2022	DT	Added Version Level 125. The additional Item User Fields, 3 and 4, were added to the high-level data that can be sent.
Dec 12, 2022	DT	Added the PLC Request Remote Setup Configuration enhancement to Version Level 125.