



## Release Notes for MP2000iec Series

### Release 1.1.2 Build 5

### Cumulative for changes from 1.1.1 Build 4

Yaskawa Electric America, Inc.

1	Supported Function Blocks.....	2
1.1	Unsupported Function Block Inputs and Outputs.....	3
2	Important changes from 1.1.1 Release.....	4
2.1	Function Blocks .....	4
2.2	EtherNet/IP .....	8
2.3	Modbus/TCP.....	9
2.4	Hardware.....	9
2.5	Mechtrolink.....	9
2.6	Web Interface.....	10
3	Known issues .....	10
3.1	Function Blocks .....	10
3.2	Modbus/TCP .....	13
3.3	MECHATROLINK.....	13
3.4	SGDV Servo Drive .....	13
3.5	Outputs.....	13
3.6	Communication Overhead for Modbus/TCP and Ethernet/IP.....	14
3.7	Typical PLC Scan Times .....	14

## Important!

Firmware 1.1.1 is required to upgrade to Firmware 1.1.2. Upgrading from any prior firmware version requires first upgrading to Firmware 1.1.1 and then upgrading to Firmware 1.1.2.

## 1 Supported Function Blocks

The following list contains the function blocks supported in this release:

- MC\_AbortTrigger
- MC\_FinishHoming
- MC\_GearIn\*
- MC\_GearInPos\*
- MC\_GearOut\*
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveSuperimposed
- MC\_MoveVelocity
- MC\_Power\*
- MC\_ReadActualPosition
- MC\_ReadActualTorque
- MC\_ReadActualVelocity
- MC\_ReadAxisError\*
- MC\_ReadParameter\*
- MC\_ReadBoolParameter
- MC\_ReadStatus\*
- MC\_Reset\*
- MC\_SetPosition\*
- MC\_StepRefPulse\*
- MC\_StepLimitSwitch\*
- MC\_Stop
- MC\_TorqueControl
- MC\_TouchProbe
- MC\_WriteBoolParameter
- MC\_WriteParameter
- Y\_CamFileSelect
- Y\_CamIn\*
- Y\_CamOut\*
- Y\_CamScale
- Y\_CamShift
- Y\_CamStructSelect
- Y\_ClearAlarms
- Y\_HoldPosition\*\*
- Y\_ReadAlarm

- Y\_ReadCamTable\*
- Y\_ReadDriveParameter
- Y\_ResetAbsoluteEncoder\*
- Y\_ResetMechatrolink\*
- Y\_SlaveOffset
- Y\_VerifyParameters\*
- Y\_WriteCamTable
- Y\_WriteDriveParamter
- Y\_WriteParameters\*

\* Indicates that this function block has a known issue.

\*\* Indicates that the function block has been deprecated and will be removed in a future release.

### 1.1 **Unsupported Function Block Inputs and Outputs**

The following function block inputs/outputs are not supported and are reserved for future use:

- MC\_MoveAbsolute.Jerk
- MC\_MoveRelative.Jerk
- MC\_MoveAdditive.Jerk
- MC\_MoveSuperImposed.Jerk
- MC\_MoveVelocity.Jerk
- MC\_Stop.Jerk
- MC\_Stop.BufferMode (assumed BufferMode is *aborting*)
- MC\_Power.BufferMode
- MC\_Power.Enable\_Positive
- MC\_Power.Enable\_Negative
- MC\_ReadStatus.Busy (always FALSE)
- MC\_ReadAxisError.Busy (always FALSE)
- MC\_Read[Bool]Parameter.Busy (always FALSE)
- MC\_TorqueControl.Direction
- MC\_TorqueControl.TorqueRamp
- MC\_TorqueControl.Acceleration
- MC\_TorqueControl.Deceleration
- MC\_TorqueControl.Jerk
- MC\_Write[Bool]Parameter.Busy (always FALSE)
- MC\_ReadActualPosition (always FALSE)
- MC\_GearIn.Jerk
- MC\_TouchProbe.WindowOnly
- MC\_TouchProbe.FirstPosition
- MC\_TouchProbe.LastPosition
- MC\_SetPosition.Busy (always FALSE)
- MC\_ReadActualVelocity.Busy (always FALSE)
- MC\_ReadActualTorque.Busy (always FALSE)
- MC\_GearInPos.Jerk

## 2 Important changes from 1.1.1 Release

### 2.1 Function Blocks

#### 2.1.1 Bug Fixes

- MC\_SetPosition does not set the unwrapped (non-cyclic) position (SCR 3371)
  - Fix: MC\_SetPosition sets the unwrapped (non-cyclic) position
  - Details: Because MC\_SetPosition does not set the unwrapped (non-cyclic) position it can be impossible to set the cam table's master position to the beginning of the table.
- MC\_Power does not clear cam state to zero (SCR3353)
  - Fix: MC\_Power clears cam state to zero when axis is disabled.
  - Details: If MC\_Power.Enable=FALSE while the cam state is engaged (3), the cam state remains at engaged (3), but it should be not engaged (0).
- MC\_Power fails if axis warning is set (SCR 3436)
  - Fix: MC\_Power succeeds even if axis warning is set.
  - Details: If MC\_Power.Enable=TRUE and there is a warning on the axis, MC\_Power.Status is FALSE.
- MC\_SetPosition incorrectly set the command velocity. (SCR 3469)
  - Fix: MC\_SetPosition does not affect command velocity.
  - Details: MC\_SetPosition incorrectly set the command velocity, which could cause erroneous motion if a move immediately followed.
- Y\_CamFileSelect uses case sensitive comparison in file extension check (SCR 3311)
  - Fix: File extension is case insensitive.
  - Details: The file extension determines the type of cam table, and this mapping uses a case sensitive comparison.
- Y\_VerifyParameters: matching output always on (SCR 3457)
  - Fix: Y\_VerifyParameters.Matches is only set to true after all parameters have been checked.
  - Details: Errors were not clearing the Matches output. Matches output is only TRUE after all parameters have been verified.
- Need to expose exact PLC alarm code (SCR 3458)
  - Fix: Added more PLC alarm codes to better detail the error.
  - Details: Multiple PLC error codes were mapped to PLC Critical Error. Now each error maps to a unique alarm code.
- MC\_StepRefPulse had internal motion kernel error (SCR 3433)
  - Fix: MC\_StepRefPulse now generates the a MotionProhibited error code.
  - Details: If axis was disabled or had an alarm, the homing could not be initiated and the ErrorID output was set to internal motion kernel error.
- Y\_ReadCamTable causes address bus error if Data has wrong data type (SCR 3440)
  - Fix: The raw memory address is checked prior to accessing structure members.
  - Details: Firmware function block have to use an "any" data type for any non-primitive input. This "any" data type corresponds to a data structure including some data type checks, but when a BOOL type is passed in, trying to use those data types checks to check if the input is valid creates an address bus error.

- MC\_StepRefPulse issue when triggered multiple times (SCR 3334)
  - Fix: After detecting the C pulse, MC\_StepRefPulse moves back to the C pulse using an absolute move “shortest way”.
  - Details: When at the C pulse with a rotary axis, executing MC\_StepRefPulse sometimes some times causes the motor to move more than one revolution. Because a rotary move wraps, the move absolute back to C-pulse would sometimes take the long way back the C pulse.
- CamState might be zero for one cycle if Y\_CamIn is aborted by Y\_CamIn (SCR 3372)
  - Fix: When the first Y\_CamIn is cancelled by the aborting Y\_CamIn segment, the second Y\_CamIn directly sets the cam state.
  - Details: If the aborting Y\_CamIn executed first, then it was possible for the CamState to be zero for one cycle.
- AlarmTask takes 1ms when alarm occurs (SCR 3151)
  - Fix: Lower priority alarm handling, such as maintaining the alarm history, is delegated to a lower priority task. This duration has been reduced to 240 usec.
  - Details: The alarm task could take up to 1ms when an alarm occurs, which would preempt the PLC scan.
- MC\_StepRefPulse against a torque limit causes strange motion (SCR 3422)
  - Fix: After the torque limited velocity move hits the torque limit, it immediately switches to hold position control mode to resist changing positions.
  - Details: When MC\_StepLimitSwitch FB moves towards the c-pulse and encounters a resistance, it gives an overtorque error (3) on the FB. This is correct behavior. However, afterwards, the axis jumps back in the opposite direction, moving as much as 4 revs on a rotary axis. On a linear axis against a hard stop, the axis jumped back 1 cm.
- MC\_StepRefPulse conflicts with MC\_TouchProbe (SCR3170)
  - Fix: For the second function block to execute, Error will be TRUE and ErrorID will be 0x112c (AxisLatchFunctionInUse).
  - Details: MC\_StepRefPulse, MC\_StepLimitSwitch and MC\_TouchProbe all use drive side latching, but there are no checks to see if this feature is already being used for another function block.
- CommandedPositionNonCyclic (Parameter 1016) ignores axis offset (SCR 3423)
  - Fix: MC\_SetPosition affects commandedPositionNonCyclic (Parameter 1016).
  - Details: commandedPositionNonCyclic (Parameter 1016) is not affected by MC\_SetPosition
- Y\_CamOut.ErrorID=4375 when executed while waiting to engage (SCR 3402)
  - Fix: Y\_CamOut correctly cancels waiting to engage state.
  - Details: This error should occur when there cam state is 0, not engaged, but should not occur when the came state is 1, waiting to engage.
- Increase Global Memory size (SCR 3347)
  - Fix: Global Memory size increased to 1.5 MB; retain memory increased to 1.5 MB on MP2310iec.
  - Details: Previously, the global memory was 400k.

- In aborting mode, Y\_CamShift, Y\_CamScale and Y\_SlaveOffset “cancel Y\_CamIn (SCR 3484)
  - Fix: Y\_CamShift, Y\_CamScale and Y\_SlaveOffset only abort other Y\_CamShift, Y\_CamScale and Y\_SlaveOffset function blocks.
  - Details: If Y\_CamIn was waiting to engage, the CamShift, CamScale and SlaveOffset could cancel Y\_CamIn.
- The slave axis will jump when changing the master’s position MC\_StepRefPulse (SCR 3237)
  - Fix: MC\_StepRefPulse is prohibited on a master that is involved in coming or gearing.
  - Details: The slave axis will jump, often causing position overflow drive alarms, when changing the master’s position with MC\_StepRefPulse. This was found while trying to home an X-X configuration.
- MC\_TouchProbe causes watchdog with a virtual axis (SCR 3489)
  - Fix: MC\_TouchProbe with a virtual axis will set Error=TRUE and ErrorCode=0x1127 (VirtualAxisNotAllowed).
  - Details: Using MC\_TouchProbe with a virtual axis caused an exception in the PLC task, resulting in a PLC watchdog error.
- Y\_ReadCamTable and Y\_WriteCamTable do not set any outputs if StartIndex & EndIndex are same (SCR 3488)
  - Fix: If StartIndex>=EndIndex, then Error=TRUE and ErrorID=0x121d (InvalidEndIndex).
  - Details: Y\_ReadCamTable and Y\_WriteCamTable did not correctly handle the error condition, StartIndex >= EndIndex and as a result did not set any outputs.
- Internal Motion Kernel Error with MC\_StepLimitSwitch (SCR 3497)
  - Fix: MC\_StepLimitSwitch now ensures that the limit switch is off for 10 Mechatrolink cycles before stopping, and reports FailedToMoveAwayFromOT (0x112d) if the limit switch is unexpectedly on.
  - Details: If the axis is already at a limit, then the MC\_StepLimitSwitch moves away from the limit, and stops once the limit is off. However, if after stopping the limit was still on, perhaps because of a noisy switch or if switched back on by hand with a manual tests case, then the MC\_StepLimitSwitch would post an internal motion kernel error.
- Axis state from Stopping to error stop seems broken (SCR 3500)
  - Fix: The axis switches from Stopping to ErrorStop when the controller detects Mechatrolink communication alarms.
  - Details: When the Mechatrolink cable was disconnected while stopping, the axis remained in the stopping state. The problem was that the axis was not listening for communication alarms.

- MC\_StepLimitSwitch (neg direction) from within a POT limit gives internal motion kernel error (SCR 3501)
  - Fix: MC\_StepLimitSwitch only checks the limit switch in the direction it supposed to move. The opposite limit switch is not checked.
  - Details: If the axis is already at a limit, then the MC\_StepLimitSwitch moves away from the limit, and stops once the limit is off. If the limit is still on, then the axis posts internal motion kernel error. However, in this case, MC\_StepLimitSwitch was not differentiating between positive and negative over travels.
- MC\_StepRefPulse had internal motion kernel error (SCR 3502)
  - Fix: Removed implicit hold in MC\_StepLimitSwitch, and added interlock between MC\_StepRefPulse and MC\_StepLimitSwitch.
  - Details: After setting the Done output, MC\_StepLimitSwitch performed an implicit hold position to switch back into position control mode. MC\_StepRefPulse started executing when MC\_StepLimitSwitch.Done was TRUE, checked if any moves were executing, and then posted an internal motion kernel error when it detected the hold position from MC\_StepLimitSwitch.

### 2.1.2 New issues

- Do not use CamScale with one way cam (SCR 3461)

**Details:** Using Y\_CamScale with a one way cam can cause large changes in position since the cumulative cyclic offset is scaled too. This is the documented behavior.

**Mitigation:** Do not use Y\_CamScale with one way cams.
- Downloading application while executing Y\_CamFileSelect crashes controller (SCR 3490)

**Details:** When Y\_CamFileSelect executes it spawns a new thread to load the cam file in the back ground. When a new program is downloaded, the original function block is destroyed, and the background thread will be executing deleted or partially deleted objects.

**Mitigation:** Do not download a new program while Y\_CamFileSelect is executing.
- MC\_Power shows Status High even after Mechatrolink is down (SCR 3493)

**Details:** When the Mechatrolink cable is disconnected, the controller can not communicate with the drive, and it generates a controller watch dog alarm (0x23010001). The drives response to this same event it to generate an AE50 alarm and disable. Since communication has been lost, the controller does not detect that the drive has disabled, so MC\_Power.Status is still high.

**Mitigation:** The axis is in the ErrorStop state, as it should be, so motion is prohibited, and MC\_ReadAxisError shows the communication error. If the communication cable is inserted again, the MC\_Power.Status will be FALSE.
- Downloading application while executing MC\_StepRefPulse and MC\_StepRefLimit crashes controller (SCR 3499)

**Details:** When MC\_StepRefPulse and MC\_StepRefLimit execute, they spawns a new thread to perform the homing sequence. When a new program is downloaded, the original function block is destroyed, and the background thread will be executing deleted or partially deleted objects.

**Mitigation:** Do not download a new program while executing MC\_StepRefPulse and MC\_StepRefLimit.



- Output Exclusivity of Y\_CamStructSelect (SCR 3503)  
**Details:** If Y\_CamStructSelect is busy when it's retriggered, both the Busy and Error outputs will be on. In this case, Y\_CamStructSelect is still loading the original structure, and when finished, then the Done output will be TRUE and both the Error and Busy will be off. The motivation for this implementation is to prevent the resource leak associated with losing track of a CamID.  
**Mitigation:** Retriggering Y\_CamStructSelect while busy is a programming error, and Error output will come on.
- Y\_VerifyParameters has InvalidParameter set when Matches is TRUE (SCR 3504)  
**Details:** When Y\_VerifyParameters.Matches=TRUE, Y\_VerifyParameters.InvalidParameter is set to the last parameter.  
**Mitigation:** Error and Error ID outputs are not set, and Expected and Actual are both zero.

## 2.2 EtherNet/IP

### 2.2.1 Bug fixes

- Investigate ways to push Ethernet IP performance (SCR 2923)
  - Fix: Ethernet/IP now runs at the resolution of the Mechatrolink update rate and is event driven, firing off only when an internal timer expires.
- EIP and Modbus tasks run even when not configured (SCR 3451)
  - Fix: Communication tasks for Ethernet/IP and Modbus/TCP are only created if configured.
  - Details: Even when not configured, Ethernet/IP and Modbus/TCP consumed CPU.
- EIP listen only status word issue (SCR 3478)
  - Fix: The controller will try to reconnect when it encounters the following errors:
    - 0x100: duplicate connection/connection in use
    - 0x106: ownership conflict
    - 0x113: too many connections
    - 0x119: no owning connection
    - 0x11A: too many connections (application)
    - 0x204: Unconnected Send timeout
    - 0x302: Not enough bandwidth
    - 0x31E: No more consumer resourcesIn these cases, the Ethernet/IP status word will contain a '2' in the top nibble and the Ethernet/IP error code in the lower three nibbles.
  - Details: The scanner is repeatedly spamming forward open requests even though the adaptor replies with a 119 error, indicating the reconnect timeout isn't working properly. Additionally, error 119 is currently a fatal error, so recovering is not possible.



- E/IP 'exclusive owner' detection may be wrong (SCR #3236)
  - Fix: The E/IP 'exclusive owner' check only verifies that the O2T instances don't overlap-- the T2O can overlap OK
  - Details: Error 106 should only be issued if an \*output\* (o2t) conflict occurs. Input mappings can be shared amongst exclusive connections. See CIP spec 3-5.6.2.4 and 3-6.3.

## **2.3 Modbus/TCP**

### **2.3.1 Bug fixes**

- Investigate ways to improve modbus performance (SCR 2923)
  - Fix: Modbus/TCP now run at the resolution of the Mechatrolink update rate and is event driven, firing off only when an internal timer expires.
- EIP and Modbus tasks run even when not configured (SCR 3451)
  - Fix: Communication tasks for Ethernet/IP and Modbus/TCP are only created if configured.
  - Details: Even when not configured, Ethernet/IP and Modbus/TCP consumed CPU.

## **2.4 Hardware**

### **2.4.1 Bug fixes**

- ALM light goes on for warnings (SCR 3346)
  - Fix: ALM light ignores warnings

### **2.4.2 New hardware support**

- MP2310

## **2.5 Mechtrolink**

### **2.5.1 Bug fixes**

- S-curve moving average filter causes jerk in acceleration phase (SCR 3431)
  - Fix: Control mode passed through filter.
  - Details: For the first filter time constant period, the control mode is being set to NoControl, so the first part of the trajectory is not being sent to the drives.
- Ignore one bad Mechatrolink packet (SCR 3463)
  - Fix: Controller will ride through one bad Mechtrolink packet.
  - Details: In a noisy environment, a bad packet causes a controller watch dog alarm.
- Implement Mechatrolink retries (SCR 3385)
  - Fix: Implemented Mechatrolink retries (SCR 3385)
  - Details: In noisy environments, Mechatrolink retries can improve robustness.

## 2.6 Web Interface

- Web applet does not support setting negative parameter values (SCR 3441)
  - Fix: Can display of negative values.
  - Details: Due to a display bug it appeared as if it was impossible to set drive parameters to negative values. Values were being set correctly.
- Large applet logging download disconnects IDE (SCR 2717)
  - Fix: Priority of RMI communication task is now lower than IDE communication.
  - Details: Previously, applet logging pre-empted IDE communication.
- Web Interface for MP2300siec disrupts operation of MotionWorks IDE (SCR 3428)
  - Fix: Priority of web interface is now lower than IDE communication (SCR 3428)
  - Details: Web Interface for MP2300siec disrupts operation of MotionWorks IDE

## 3 Known issues

### 3.1 Function Blocks

#### 3.1.1 Bugs

- Y\_CamFileSelect
  - Downloading application while executing Y\_CamFileSelect crashes controller (SCR 3490)  
**Details:** When Y\_CamFileSelect executes it spawns a new thread to load the cam file in the back ground. When a new program is downloaded, the original function block is destroyed, and the background thread will be executing deleted or partially deleted objects.  
**Mitigation:** Do not download a new program while Y\_CamFileSelect is executing.
- MC\_StepRefPulse and MC\_StepRefLimit
  - Downloading application while executing MC\_StepRefPulse and MC\_StepRefLimit crashes controller (SCR 3499)  
**Details:** When MC\_StepRefPulse and MC\_StepRefLimit execute, they spawns a new thread to perform the homing sequence. When a new program is downloaded, the original function block is destroyed, and the background thread will be executing deleted or partially deleted objects.  
**Mitigation:** Do not download a new program while executing MC\_StepRefPulse and MC\_StepRefLimit.

#### 3.1.2 Usage Notes

- MC\_GearOut
  - MC\_GearOut holds current velocity even if not gearing. (SCR 2808)  
**Details:** For example, executing MC\_GearOut while a MC\_MoveAbsolute function block is active will abort the MC\_MoveAbsolute function and hold the current velocity.  
**Mitigation:** Only call MC\_GearOut when gearing. If holding the current velocity is not desired, then use MC\_Stop.

- MC\_Power
  - A 95 being issued when MC\_Power disabled (SCR 2810, 3065)  
**Mitigation:** User programs can clear this alarm.
  - MC\_Power shows Status High even after Mechatrolink is down (SCR 3493)  
**Details:** When the Mechatrolink cable is disconnected, the controller can not communicate with the drive, and it generates a controller watch dog alarm (0x23010001). The drives response to this same event it to generate an AE50 alarm and disable. Since communication has been lost, the controller does not detect that the drive has disabled, so MC\_Power.Status is still high.  
**Mitigation:** The axis is in the ErrorStop state as it should be, so motion is prohibited, and MC\_ReadAxisError shows the communication error. If the communication cable is inserted again, the MC\_Power.Status will be FALSE..
- MC\_ReadAxisError
  - Alarm does not match alarm shown on drive (SCR 2792)  
**Mitigation:** The drive may have multiple alarms, and one of these is returned by MC\_ReadAxisError
- MC\_ReadStatus (Axis State Machine):
  - No transition from ErrorStop to Disabled when MC\_Power.Enable=False. (SCR 2822)  
**Mitigation:** Technically this is not part of the PLCopen specification; the specification does not indicate any transitions to Disabled state.
  - No transition from Disabled to ErrorStop when MC\_Power.Enable=True and there is an error on the axis. (SCR 3450)  
**Mitigation:** Customers should use MC\_ReadAxisError to determine when the axis has an error.
- MC\_Reset
  - MC\_Reset does not clear A.ED on Sigma II (SCR 2729)  
**Details:** A.ED alarm requires the servo network to be reset.
- MC\_StepRefPulse & MC\_StepLimitSwitch
  - MC\_StepRefPulse behaves incorrectly at high command velocity (SCR 2879)  
**Details:** When the velocity is set at 50 rev/s the motor spins for several seconds before the Done output is TRUE. The issue is that the torque limited velocity move is immediately followed by a hold position, and the hold position takes several revolutions to stop the axis. Suggested maximum speed is 1 rev/s.  
**Mitigation:** This issue does not occur with slower velocities (less than 1 rev/s) which are more typical.
  - MC\_StepLimitSwitch only supports one LimitSwitchMode: MC\_EdgeOn (3131)  
**Details:** MC\_StepLimitSwitch only works when detecting the rising edge of an input.  
**Mitigation:** Application use MC\_EdgeOn only.
- MC\_TorqueControl
  - MC\_TorqueControl requires MC\_Stop before using any other motion function block. (SCR 3051)  
**Details:** MC\_TorqueControl can not be aborted by a 'position mode' motion block such as MC\_MoveAbsolute.  
**Mitigation:** Changing control modes while moving has not been a requirement for a customer

- Y\_CamFileSelect
  - Y\_CamFileSelect becomes unresponsive (SCR 3393)  
**Details:** If the Execute input is toggled off and on while the function block is busy loading a file, then the function blocks output will never turn on.  
**Mitigation:** The Execute input on the Y\_CamFileSelect block should be interlocked with the busy output so that the Execute input will not “see” a rising edge while the busy output is set.
- Y\_CamIn
  - Y\_CamIn does not detect if the engage window is too small (SCR 3356)  
**Details:** If the window is too small, Y\_CamIn will not engage.
  - Y\_CamIn engages after a cycle of engage position (SCR 3413)  
**Details:** When the axis crosses the engage position, the controller prepares the axis for cam, but the actual cam processing does not occur until one Mechatrolink cycle later, when all the axes are processed at the same time.  
**Mitigation:** To ensure a smooth transition into a coming, the user can use a “pick-up” cam table where the axis is idle with in the position window.
- Y\_CamOut
  - No automatic adjustment if command position does not match disengage position (SCR 3358)  
**Details:** The slave is commanded to the disengage position with in one Mechatrolink cycle, perhaps causing too large a control effort for the drives.  
**Mitigation:** Applications should disengage in a portion of the profile while the slave is in a stopping portion of the profile or use a “drop off” cam table.
- Y\_CamScale
  - CamScale with one way cam is broken (SCR 3461)  
**Details:** Using Y\_CamScale with a one way cam can cause large changes in position since the cumulative cyclic offset is scaled too.  
**Mitigation:** Do not use Y\_CamScale with one way cams.
- Y\_CamStructSelect
  - Output Exclusivity of Y\_CamStructSelect (SCR 3503)  
**Details:** If Y\_CamStructSelect is busy when it’s retriggered, both the Busy and Error outputs will be on. In this case, Y\_CamStructSelect is still loading the original structure, and when finished, then the Done output will be TRUE and both the Error and Busy will be off. The motivation for this implementation is to prevent the resource leak associated with loosing track of a CamID.  
**Mitigation:** Retriggering Y\_CamStructSelect while busy is a programming error, and Error output will come on.
- Y\_ResetMechatrolink
  - Y\_ResetMechatrolink anomilies (SCR 3420)  
**Details:** MC\_Power will get kernel error 61713 if enabled while Y\_ResetMechatrolink is executed.  
**Mitigation:** Use Y\_ClearAlarms to clear the alarm after the reset is complete.

- Y\_VerifyParameters
  - Y\_VerifyParameters has InvalidParameter set when Matches is TRUE (SCR 3504)  
**Details:** When Y\_VerifyParameters.Matches=TRUE, Y\_VerifyParameters.InvalidParameter is set to the last parameter.  
**Mitigation:** Error and Error ID outputs are not set, and Expected and Actual are both zero.

### 3.2 Modbus/TCP

- Function code 15: write multiple coils is not supported (SCR 2739)  
**Details:** Write multiple coils is not supported, so each coil has to be written using a separate transaction. As a result, writing multiple coils is not recommended. Use registers instead.

### 3.3 MECHATROLINK

- Controller reboots if gearing 16 axes with a 2ms Mechatrolink update rate (SCR 2919)  
**Details:** MC\_GearIn.Execute=TRUE for all axes, the controller reboots.  
**Mitigation:** Increase the Mechatrolink update rate.

### 3.4 SGD Servo Drive

- A94B alarm generated after Relative, Absolute, or Geared move. (SCR 3083)  
**Details:** With SGD drives previously tuned with Sigma Win +, executing MC\_MoveRelative, MC\_MoveAbsolute or MC\_GearIn(Pos) at roughly half the rated speed causes an A94B warning. This is caused by “model following control” being enable in Pn140.  
**Mitigation:** Set SGD drive Pn140 to the default value of 0x0100.

### 3.5 Outputs

- Outputs can not be retained (SCR 3475)  
**Details:** In the IDE, the Global Variables table contains a column marked “Retain.” Selecting a check box within this column causes the corresponding variable to be allocated in SRAM. However, for outputs, this feature is not working. At this point, it is not clear if the problem is with the IDE or the firmware.  
**Mitigation:** This issue is fairly easy to workaround by first storing the output value in a global variable that is retained and then copying the global variable to the output.

### 3.6 Communication Overhead for Modbus/TCP and Ethernet/IP

Release 1.1.2 improved Modbus/TCP and Ethernet/IP performance. First, the Modbus/TCP and Ethernet/IP communication drivers are optional—if they are not configured, there is no performance overhead. Second, the maximum scan time of both communication drivers was reduced from 10ms to the Mechatrolink update rate, nominally 2ms. Consequently, the minimum RPI has been reduced to 2ms for Ethernet I/P. Third, the communication driver’s scan only consumes CPU if there is a request to service.

Additionally, MotionWorksIEC Pro allows the Ethernet/IP and Modbus/TCP to be assigned to a task, allowing communication to run at a slower rate than motion. (When using MotionWorksIEC Express to configure and program the MP2300 series controller, there is only one user task so Ethernet/IP and Modbus/TCP must be attached to it.)

With the default configurations for Ethernet/IP and Modbus/TCP, the communication drives consume roughly 400 usec each every time they run. Additionally, Ethernet/IP and Modbus/TCP add 300usec each of overhead to the scan that they are attached to.

### 3.7 Typical PLC Scan Times

Based on performance measurements, the follow PLC scan times are typical:

	Case 1	Case 2	Case 3	Case 4
Number of servo drives	4	4	8	8
MECHATROLINK period (ms)	1	2	2	3
PLCopen Function blocks	40	40	40	40
Number of axes moving at once.	4	4	8	8
Ethernet/IP connections	1	1	1	1
Scan time average (ms)	7	4	7	5
Scan time worst case (ms)	17	9	26	17