



Release Notes for MP2300Siec

Release 1.1.1 Build 4

Cumulative for changes from 1.0.7 Build 4

Yaskawa Electric America, Inc.

1	Important Function Block Issues	2
2	Supported Function Blocks	3
2.1	Unsupported Function Block Inputs and Outputs	4
3	Important changes from 1.0.7 Release	5
3.1	Function Blocks	5
3.2	EtherNet/IP	8
3.3	Modbus/TCP	8
3.4	Hardware	9
4	Known issues	10
4.1	Function Blocks	10
4.2	Modbus/TCP	13
4.3	MECHATROLINK	13
4.4	SGDV Servo Drive	14

1 Important Function Block Issues

Release testing revealed two issues with the MC_StepRefPulse function block:

- MC_StepRefPulse issue when triggered multiple times (SCR 3334)
Details: When at the C pulse, executing MC_StepRefPulse sometimes will cause the motor to move several revolutions. This issue is not easily repeatable, and triggering MC_StepRefPulse multiple times while at the C pulse is not a typical use case.
Preventative action: The distance and time limits should be used to prevent the axis from travelling too far.
- MC_StepRefPulse against a torque limit causes strange motion (SCR 3422)
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 moves in the opposite direction.
Preventative action: The TorqueLimit input is not supported, and its use is prohibited until this issue is fixed.

2 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_WriteDriveParameter
- 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.

2.1 Unsupported Function Block Inputs and Outputs

The following function block inputs and 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_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

3 Important changes from 1.0.7 Release

3.1 Function Blocks

3.1.1 New Function Blocks

The following function blocks were added in the 1.1.0 release:

- Y_CamFileSelect
- Y_CamIn
- Y_CamOut
- Y_CamScale
- Y_CamShift
- Y_CamStructSelect
- Y_ClearAlarms
- Y_ReadCamTable
- Y_ReadDriveParameter
- Y_ResetAbsoluteEncoder
- Y_SlaveOffset
- Y_VerifyParameters
- Y_WriteCamTable
- Y_WriteDriveParameter
- Y_WriteParameters

3.1.2 Bug Fixes

- MC_TorqueControl ignores direction input (SCR 3075)
 - Fix: direction input is now used.
 - Problem details: Direction input was previously ignored. Instead, the direction was determined based on the sign of the Torque input. For backwards compatibility, negative Torque input will still result in torque in the reverse direction.
- MC_MoveSuperImposed does not abort when another motion block executes (SCR 3128)
 - Fix: MC_MoveSuperImposed now aborts.
 - Problem details: If MC_MoveSuperImposed is executing and another motion block executes with BufferMode set to aborting, the MC_MoveSuperImposed does not abort.
- MC_MoveVelocity with zero velocity input causes an error. (SCR 3142)
 - Fix: Zero velocity no longer causes an error.
 - Problem details: If the Velocity=0, then the Error=TRUE and ErrorID=4658.
- MC_GearIn and MC_GearInPos with zero numerator causes an error. (SCR 3143)
 - Fix: Zero numerator no longer causes an error.
 - Problem details: If the Numerator=0 for MC_GearIn and MC_GearInPos, then the Error=TRUE and ErrorID=4665.
- RTC_S in the PROCONOS firmware library now gives a valid time and date string. (SCR 3101)

- Fix: Valid time string is now returned.
- MC_GearIn and MC_GearInPos require correct axis order (SCR 3053)
 - Fix: Master axis ID no longer needs to be less than the slave axis ID.
 - Problem details: The master axis ID had to be less than the slave axis ID or a velocity dependent position error would occur.
- "InTorque" output not correct (SCR 3169)
 - Fix: InTorque output will turn on when moving in the negative direction.
 - Problem details: TorqueControl.InTorque output would never equal true if moving in the negative direction.
- Reverse order of alarms coming from amplifier (SCR 3179)
 - Fix: MC_ReadAxisError shows drive alarms.
 - Problem details: When an encoder cable was disconnected, MC_ReadAxisError would show AxisErrorID=16#0002 (and ErrorClass=16#2301) for the axis disabled unexpectedly alarm rather than showing AxisErrorID=16#C90 for the encoder cable disconnected alarm on the drive.
- The slave axis will jump when changing the master's position MC_SetPosition (SCR 3226)
 - Fix: MC_SetPosition checks if the axis is being used as a master and generates an error if it is.
 - Problem details: Changing the master's position with MC_SetPosition causes a jump in the slave's position.
- Discrete Motion block does not abort MC_MoveSuperImposed block (SCR 3260)
 - Fix: When a motion function block executes with BufferMode=MC_BufferMode#Aborting, MC_MoveSuperImposed will be aborted too.
 - Problem details: Executing a motion function block with BufferMode=MC_BufferMode#Aborting does not abort MC_MoveSuperImposed
- MC_Power fails to disable servos (SCR 3269)
 - Fix: MC_Power tracks Enabled input.
 - Details: If Enable input is toggled on-off-on very quickly, the function block can lose its internal state and not disable the axis.
- Increase PLC program size (SCR 3292)
 - Fix: Program memory size doubled from 400k to 800k.
 - Details: Large customer applications reported that the program memory usage exceeded the 80% threshold, though none actually ran out of program memory.
- Incorrect Commanded Position for MC_MoveRelative FB with repeated 2um moves (SCR 3331)
 - Fix: Moves can complete in one Mechatrolink cycle.
 - Problem details: Moves which should only take one Mechatrolink cycle instead took two Mechatrolink cycles. In the first cycle, the command position would move away from the target position, and in the second cycle the move would reach the target position.
- Add error condition checks for MC_TorqueControl.Direction (SCR 3312)
 - Fix: MC_TorqueControl generates an error if MC_TorqueControl.Direction is MC_Direction#shortest_way or MC_Direction#current_direction.

- Problem details: MC_TorqueControl.Direction defaulted MC_Direction#shortest_way or MC_Direction#current_direction to MC_Direction#positive_direction.
- MC_SetPosition unable to execute repeatedly (SCR 3341)
 - Fix: Rising edge of execute correctly detected after stopping and restarting the program.
 - Problem details: Due to conditional execution, the internal function block does not track the execute state correctly and does not detect the rising edge of execute if the program was stopped with the execute high. This issue affects all function blocks.

3.1.3 New issues

- Y_CamFileSelect: File extension is case sensitive (SCR 3311)
 - The file extension determines the type of cam table, and this mapping uses a case sensitive comparison.
- MC_StepRefPulse issue when triggered multiple times (SCR 3334)
 - When at the C pulse, executing MC_StepRefPulse sometimes will sometimes cause the motor to move without ever stopping.
- MC_Power does not clear cam state to zero (SCR 3353)
 - 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).
- Y_CamIn does not detect if the engage window is too small (SCR 3356)
 - If the window is too small, Y_CamIn will not engage.
- EndOfProfile should be held high for non-periodic cams (SCR 3357)
 - EndOfProfile is pulsed, but not held high in this situation.
- Y_CamFileSelect becomes unresponsive (SCR 3393)
 - 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.
- Y_CamOut causes error 4375 if executed while waiting to engage (SCR 3402)
 - 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.
 - If the cam state is 1, then use the MC_Stop block instead.
- Y_ResetMechatrolink anomalies (SCR 3420)
 - MC_Power will get kernel error 61713 if enabled while Y_ResetMechatrolink is executed.
- MC_StepRefPulse against a torque limit causes strange motion (SCR 3422)
 - The TorqueLimit input is not supported and its use is prohibited.
 - 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 moves in the opposite direction.
- commandedPositionNonCyclic (Parameter 1016) ignores axis offset (SCR 3423)
 - commandedPositionNonCyclic (Parameter 1016) is not affected by MC_SetPosition.
- Web Interface for MP2300siec disrupts operation of MotionWorks IDE (SCR 3428)
 - Communication between controller and web applet or MotionWorks IEC configuration tool can preempt the PLC scan.

- MC_Power fails if axis warning is set (SCR 3436)
 - If MC_Power.Enable=TRUE and there is a warning on the axis, MC_Power.Status is FALSE.
 - MC_Reset can be used to clear the warning.
- Y_ReadCamTable causes address bus error if Data has wrong data type (SCR 3440)
 - 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 had internal motion kernel error (SCR 3433)
 - The axis had an alarm so the function block could not execute, but the function block should have had a more reasonable alarm code.
 - The function block behaves correctly, but the error code is not informative. Also, users can check and clear alarms with MC_ReadAxisError and MC_Reset prior to executing MC_StepRefPulse.
- RMI does not support setting negative parameter values (SCR 3441)
 - It’s impossible to set drive parameters to negative values via the applet or the configuration tool.
- MC_ReadStatus does not report the correct state of the axis sometimes when there is an error (SCR 3450)
 - Specifically, when enabling an axis that has an error and is in the disabled state, the axis remains in the disabled state and does not transition to the error stop state.

3.2 EtherNet/IP

3.2.1 Bug fixes

- EtherNet/IP client stuck in reconnect when EtherNet/IP server is rebooted (SCR 3213)
 - Fix: Client now reconnects on a warm start.
 - Problem details: If the Ethernet/IP server is rebooted via the web interface or configuration tool while a client is connected, the client would be stuck trying to reconnect after the sever reboots. However, if the server was power cycled, the client would reconnect.
- E/IP I/O connections are dropped when TCP connection closed by server (SCR 3232)
 - EIP client should NOT be tying I/O connection death to TCP connection death

3.3 Modbus/TCP

3.3.1 Bug fixes

- Modbus/TCP poll period was inaccurate (SCR 2936)
 - Fix: Poll period now matches the value specified in the configuration file.
 - Problem details: On startup, the poll period was always 100ms regardless of the configuration value.

- Could not connect Modbus controller clients after Modbus server controller rebooted (SCR 2964)
 - Fix: Modbus clients can now reconnect.
 - Problem Details: After Modbus server reboots, the clients have watch dog errors and do not reconnect to the server. If the clients are restarted, then they connect correctly.
- Modbus not reconnecting to remote server after slave power cycle (SCR 3296)
 - Fix: Modbus can now reconnect.
 - Details: After power cycling the slave, a Red Lion HMI, the status word went to 2 from 4096 and then stayed at 2, even after the slave came back online. To reconnect, the PLC had to be restarted.
- Modbus master opening too many TCP ports with multiple I/O blocks (SCR 3230)
 - Fix: Multiple TCP ports no longer opening multiple I/O blocks to the same server.
 - Problem details: When the MP2300Siec is configured as a MODBUS master, and a write/read holding register group is added, the controller comes up with an I/O driver error and does not allow a cold start.
- Modbus I/O connection timeout causes I/O error after warm/cold boot (SCR 3246)
 - Fix: Modbus I/O can connect again after PLC stop/start without a reboot.
 - Problem details: When there's no Modbus slave available (i.e. the connection never completes), the initial boot gives a timeout status, but a following stop->cold/warm gives an I/O error instead
- Deadlock in the MBTCP (SCR 3352)
 - The MBTCP can experience a deadlock condition in some cases when cycling between cold start and stop states.

3.4 Hardware

3.4.1 New hardware support

- LIO-06
- Virtual Axes
- POT/NOT conditions are now alarms.

Details: *Previously, there was no action associated with the POT/NOT condition, so motion blocks would still be active.*

4 Known issues

4.1 Function Blocks

4.1.1 Bugs

- MC_StepRefPulse & MC_StepLimitSwitch
 - MC_StepRefPulse Active output still TRUE after 1 scan (SCR 3141)
Details: If MC_StepLimitSwitch is aborted with MC_Stop, the Active output is still TRUE after 1 scan.
Mitigating factor: When the motion actually stops, the Active output is then FALSE.
 - MC_StepRefPulse issue when triggered multiple times (SCR 3334)
Details: When at the C pulse, executing MC_StepRefPulse sometimes will sometimes cause the motor to move without ever stopping.
Mitigating factor: This issue is hard to repeat, and triggering MC_StepRefPulse multiple times is not a typical use case. Also, the distance and time limits can be used to check for errors.
 - MC_StepRefPulse against a torque limit causes strange motion (SCR 3422)
The TorqueLimit input is not supported and its use is prohibited.
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 moves in the opposite direction.

4.1.2 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.
Mitigating Factor: Only call MC_GearOut when gearing.
- MC_Power
 - A 95 being issued when MC_Power disabled (SCR 2810, 3065)
Mitigating Factor: User programs can clear this alarm.
 - MC_Power does not clear cam state to zero (SCR 3353)
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).
Mitigating Factor: Use the MC_Stop block to clear the cam state.
 - MC_Power fails if axis warning is set (SCR 3436)
Details: If MC_Power.Enable=TRUE and there is the “Axis was temporarily disabled” warning on the axis, MC_Power.Status is FALSE.
Mitigating factor: MC_Reset can be used to clear the warning.
- MC_ReadAxisError
 - Alarm does not match alarm shown on drive (SCR 2792)

Mitigating factor: The drive may have multiple alarms, and one of these is returned by MC_ReadAxisError

- MC_ReadStatus (Axis State Machine):
 - Incorrect axis state with MC_MoveSuperImposed. (SCR 2567)
Mitigating factor: Executing another motion block fixes the axis state.
 - No transition from ErrorStop to Disabled when MC_Power.Enable=False. (SCR 2822)
Mitigating factor: 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)
Mitigating factor: 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_SetPosition
 - commandedPositionNonCyclic (Parameter 1016) ignores axis offset (SCR 3423)
Details: commandedPositionNonCyclic (Parameter 1016) is not affected by MC_SetPosition.

- 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.
Mitigating factor: This issue does not occur with slower velocities (less than 1 rev/s) which are more typical.
 - MC_StepRefPulse Active output still TRUE after 1 scan (SCR 3141)
Details: If MC_StepLimitSwitch is aborted with MC_Stop, the Active output is still TRUE after 1 scan.
Mitigating factor: When the motion actually stops, the Active output is then FALSE.
 - MC_StepLimitSwitch only supports one LimitSwitchMode: MC_EdgeOn (3131)
Details: MC_StepLimitSwitch only works when detecting the rising edge of an input.
Mitigating factor: Application can be wired to work within this limitation.
 - Servo behavior with limit switches after homing (SCR 3133)
Details: Commanded an axis being homed to make a MC_MoveRelative back past its limit switch the axis causing the axis to move back and forth from the pulse to the limit switch repeatedly. Using MC_MoveRelative should not be allowed while homing.
Mitigating factor: MC_MoveRelative should not be used while homing.
 - MC_StepRefPulse and MC_StepLimitSwitch conflict with MC_TouchProbe (3170)

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.

Mitigating factor: The user can add program logic to avoid this situation.

- The slave axis will jump when changing the master's position MC_StepRefPulse (SCR 3237)

Details: The slave axis will jump, often causing position overflow drive alarms, when changing the master's position with MC_StepRefPulse.

Mitigating factor: For some applications, the user can add program logic to avoid this situation. For redundant axes, such as an X-X gantry configuration, there is no workaround.

- MC_StepRefPulse had internal motion kernel error (SCR 3433)

Details: The axis had an alarm so the function block could not execute, but the function block should have had a more reasonable alarm code.

Mitigating factor: The function block behaves correctly, but the error code is not informative. Also, users can check and clear alarms with MC_ReadAxisError and MC_Reset prior to executing MC_StepRefPulse.

- 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.

Mitigating factor: Changing control modes while moving has not been a requirement for a customer

- Y_CamFileSelect

- File extension is case sensitive (SCR 3311)

Details: The file extension determines the type of cam table, and this mapping uses a case sensitive comparison.

- 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.

Mitigating factor: 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.

- EndOfProfile should be held high for non-periodic cams (SCR 3357)

Details: EndOfProfile is pulsed, but not held high in this situation.

Mitigating factor: The customer should latch this signal if it is required.

- Y_CamIn engages after a scan 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.

Mitigating factor: To ensure a smooth transition into camming, 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.
 - Y_CamOut causes error 4375 if executed while waiting to engage (SCR 3402)
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.
Mitigating factor: If the cam state is 1, then use the MC_Stop block instead.
- Y_ReadCamTable
 - Y_ReadCamTable causes address bus error if Data has wrong data type (SCR 3440)
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.
Mitigating factor: Do not attach a BOOL data type to the Data input.
- Y_ResetMechatrolink
 - Y_ResetMechatrolink anomalies (SCR 3420)
Details: MC_Power will get kernel error 61713 if enabled while Y_ResetMechatrolink is executed.
Mitigating factor: Set the enabled input to any MC_Power blocks to off before using this block.

4.2 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.

4.3 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.
Mitigating factor: Increase the Mechatrolink update rate. See Application Design Guideline, doc number [AN.MCD.09.042](#) for suggested Mechatrolink update rates.

4.4 **SGDV Servo Drive**

- A94B alarm generated after Relative, Absolute, or Geared move. (SCR 3083)
Details: With SGDV 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.
Mitigating Factor: Set SGDV drive Pn140 to the default value of 0x0100.