Release Notes for MP2300Siec

Release 1.0.7 Build 4

Cumulative for changes from 1.0.4 Build 5

Yaskawa Electric America, Inc.

Document #		
Revision	Date	Comment
1	11/13/2008	Initial revision
2	11/24/2008	Added EIP issue from test case 652
3	11/24/2008	Updated header, footer, & TOC
4		
5		
6		
7		

1 Supported Function Blocks

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

- MC_AbortTrigger
- MC_FinishHoming
- MC_GearIn*
- MC_GearInPos*
- 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_Stop
- MC_TorqueControl
- MC_TouchProbe*
- MC_WriteBoolParameter
- MC_WriteParameter
- Y_ClearAlarms
- Y_HoldPosition**
- Y_ReadAlarm
- Y_ResetMechatrolink

* Indicates that this function block has a known issue.

** Indicates a 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 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_TroqueControl.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 1.0.4 Release

2.1 Function Blocks

2.1.1 Bug fixes

- All Function Blocks
  - Conditional execution of function blocks: To reduce CPU usage, the corresponding C function for a firmware function block is only called if the Execute input is TRUE or an output is TRUE. In the IDE, the PLCopenPlus firmware library must be deleted and the PLCopenPlus-v_2_1 firmware library must be installed to use the conditional execution function blocks. (SCR 3146)
  - The Axis wrap range was incorrectly computed for negative position scales. (SCR 3159)
- MC_GearIn/MC_GearInPos
  - Previously, the slave's commanded velocity was calculated using a backwards difference of the slave's commanded position. Now, the slave's commanded velocity is calculated directly from the master's commanded velocity. (SCR 3158)
  - MC_GearInPos works correctly when the MC_GearInPos.SlaveSyncPosition is not specified. (SCR 2845)
- MC_MoveSuperImposed
  - Previously, MC_MoveSuperimpose worked the first time, but subsequent moves with the same distance did nothing. If the distance increased, then the change in distance was added. MC_MoveSuperImposed now works correctly. (SCR 3184)
- MC_Power
  - If the controller detects that the command position deviates significantly from the feedback position, the controller will post an alarm causing motion to stop. If while this alarm is active, the drive is power cycled, then controller will not re-enable the drive (SCR 3209)
- MC_ReadParameter
  - MC_ReadParameter no longer reports an error when trying to read parameter 1130 ("Position Error". (SCR 3081)
  - Added parameters: Actual Position Cyclic (1005), Actual Position Non-cyclic (1006), Commanded Position Cyclic (1015), Commanded Position Non-cyclic (1016). (SCR 2857).
- MC_ReadAxisError
  - Alarms are now cached for quicker access and improved performance. (SCR 3109)
- MC_StepLimitSwitch
  - MC_StepLimitSwitch caused watchdog errors if MC_ReadAxisError.Execute=TRUE. (SCR 3187)
- MC_Stop
  - After MC_Stop.Done=TRUE, new moves were prohibited. According to the PLCopen specification, this is wrong. While MC_Stop.Excute=TRUE, the Axis should be in the stopping state and new moves should be prohibited (SCR 3216)

2.2 EtherNet/IP

2.2.1 Bug fixes

- Multiple EtherNet/IP client controllers can connect to an EtherNet/IP server controller. Previously, other client controllers would report an ownership conflict error. (SCR 3175)
- EtherNet/IP client controller successfully reconnects to an EtherNet/IP server controller that is rebooted via the web page. Peviously, the client controller would only reconnect if the server controller was power cycled.

2.2.2 Enhancements

- Controller now supports ‘Scheduled’ as a priority enumeration. (SCR 3168)

2.3 Modbus/TCP

2.3.1 Bug fixes

- Modbus master controller successfully reconnects to a Modbus Slave controller that is rebooted. Peviously, the Modbus master controller had to be rebooted too. (SCR 3167)

2.3.2 Enhancements

- The controller now supports Modbus retain memory for coils and registers. Previously, Modbus coils and registers would not persist through a power cycle regardless of weather the data was marked as retain data or not.

2.4 System

2.4.1 Enhancements

- Firmware supports the MP2310 hardware (SCR 2969, 3117)

3 Known issues

3.1 Function Blocks

3.1.1 Bugs

- MC_ReadStatus (Axis State Machine):
  - Incorrect axis state with MC_MoverSuperimpose. (SCR 2567)
    Mitigating factor: Executing another motion block fixes the axis state.
- 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_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. around.
- MC_TorqueControl
  - "InTorque" output not correct (SCR 3169)
    Details: InTorque output is TRUE when the commanded torque reaches the Torque input, not when the feedback torque reaches it.
    Mitigating factor: The feedback torque can be manually polled in the application.
- MC_MoveSuperImposed
  - MC_MoveSuperImposed does not abort when another motion (SCR 3128)
    Details: If MC_MoveSuperImposed is executing and another motion block executes with BufferMode set to aborting, the MC_MoveSuperImpose does not aborted

3.1.2 Usage Notes

- MC_ReadStatus (Axis State Machine):
  - 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 while encoder cable unplugged. (SCR 2822)
    Mitigating factor: In this situation, MC_Power.Error=True and MC_Power.ErrorID=4371 because the drive could not be enabled.
- 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_GearInPos and MC_GearIn
  - MC_GearIn and MC_GearInPos require correct axis order. (SCR 3053)
    Details: If the slave has a lower axis ID than the master axis, then slave will use the old command position of the master causing a position lag equal to the master speed multiplied by the Mechatrolink update rate.
    Mitigating factor: Ensure that the master axis has a lower axis ID than the slave axis.
  - Zero Numerator for MC_GearIn and MC_GearInPos causes an error. (SCR 3143)
    Details: If the Numerator=0 for MC_GearIn and MC_GearInPos, then the Error=TRUE and ErrorID=4665.
    Mitigating factor: The user can add additional logic to the application to use MC_Stop rather than setting Numerator to zero.
- 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_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.
  - MC_StepRefPulse and MC_StepLimitSwitch conflict with MC_TouchProbe (3133)
    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 application needs to ensure that these blocks are not executed simultaneously.
  - 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 c 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_StepLimitSwitch.Active still true after block is aborted (SCR 3141)
    Details: The Active output on MC_StepLimitSwitch is still true after 1 scan when the function block is aborted.
    Mitigating factor: The Aborted output is true so applications can work around this issue.
  - 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_Power
  - A 95 being issued when MC_Power disabled (SCR 2810)
    Mitigating Factor: User programs can clear this alarm.

- 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.
    Mitigating Factor: User programs can detect this situation and use Y_ResetMechatrolink.
- MC_SetPosition
  - The slave axis will jump when changing the master's position MC_SetPosition (SCR 3226)
    Details: The slave axis will jump, often causing position overflow drive alarms, when changing the master's position.
    Mitigating factor: The user can add program logic to avoid this situation.
- MC_MoveVelocity
  - Zero Velocity causes an error. (SCR 3142)
    Details: If the Velocity=0, then the Error=TRUE and ErrorID=4658.
    Mitigating factor: The user can add additional logic to the application to use MC_Stop rather than setting the velocity to zero.
- MC_TorqueControl
  - MC_TorqueControl requires MC_Stop before using any other motion function block. (SCR 3051)
    Details: MC_TorqueControl cannot 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.

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 Ethernet/IP

- MP2300Siec EIP gets ownership conflict error (SCR 3118)
  Details: If two MP2300Siec controllers are connected via EtherNetIP, an ownership conflict error may occur after a long period of communication.
- MP2300Siec EIP Slave disconnects from AB ControlLogix EIP Master. (SCR 3074)
  Details: When attempting to connect to an Allen Bradley ControlLogix EIP Master with the MP2300Siec controller as an EIP Slave, a timeout occurs. It appears as if the EIP task is being starved.
  Mitigating factor: By changing the Mechatrolink scan rate or the PLC scan rate, this issue is avoided.
- Stuck in reconnect (SCR 3213)
  Details: If the EtherNet/IP server is rebooted via the web page, the EtherNet/IP client will be stuck trying to reconnect. When this occurs, the client must be restarted.
  Mitigating factor: The workaround is to power cycle the EtherNet/IP server rather than reboot through the web interface.

3.4 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 Section **Error! Reference source not found.** (**Error! Reference source not found.**) for suggested Mechatrolink update rates.

3.5 Web Interface

- Microsoft Internet Explorer Version 6 crashes if closed when applet is showing (SCR 2929)
  Details: If the applet is showing and Microsoft Internet Explorer Version 6 is closed, then a dialog box pops up asking if the crash information should be sent to Microsoft.
  Mitigating factor: This issue does not occur with Microsoft Internet Explorer Version 7.

3.6 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.

3.7 System

- Default Gateway des not change to default when E-INIT switch is on. (SCR 3102)
  Details: Default Gateway des not change to default when E-INIT switch is on.
  Mitigating factor: The Default Gateway can be changed via the web interface.