



## Release Notes

### MC21x0 version 2.6

**Document last updated:** 3/8/2007

**Product names:** MC2140, MC2120, MC2110  
**Version control label:** mc21x0 version 2.6  
**IO device:** MC2000IOU20  
**CP device build date:** 3/7/2007

#### **Description:**

The MC21x0 is a motion control processor for DC servo motors and provides one to four axes of motion. This document details bug fixes and changes for this release.

#### **Known Issues:**

If the IO chip HostRdy signal (pin 8) is used for chip busy detection, the first instruction sent to the chipset after a power on or reset may be ignored or may produce a checksum error. It is recommended that in this configuration a NoOperation command be sent to the chipset as the first instruction after a power on or reset. If the *ReadStatus* operation is used to check the HostRdy state this problem does not occur.

To determine if the motion processor has generated an interrupt to the host, the host should check the state of the host interrupt signal. In most designs this signal is connected to an external interrupt input on the host processor and may be interrogated by a host processor instruction sequence. In addition, the state of this signal as generated by the motion processor is reported by the *ReadStatus* operation. The value returned by this instruction is only accurate when the host is issuing actual chip commands. If the only communication to the device is the execution of the *ReadStatus* operation the HostInterrupt bit will not reflect the current state of the signal.

#### **Changes/Fixes:**

##### Command Changes

Fixed - SrlEnable signal was intermittently staying active after an Update command.

Fixed a problem where an update that caused a command error would return a bad checksum.

Fixed a problem where zero parameter commands could return a bad checksum.

Fixed a problem that resulted in serial communications locking up in the event that a command packet was sent by the host with additional (unnecessary) bytes.

##### Profile Changes

Fixed a problem with S-Curve when certain move settings would cause a "Fractional Period">1, which would then cause the move to fail (excessive velocity and/or overshoot).

# Version 2.5

## **Known Issues:**

<p>If the IO chip HostRdy signal (pin 8) is used for chip busy detection, the first instruction sent to the chipset after a power on or reset may be ignored or may produce a checksum error. It is recommended that in this configuration a NoOperation command be sent to the chipset as the first instruction after a power on or reset. If the <i>ReadStatus</i> operation is used to check the HostRdy state this problem does not occur.</p>
--

## **Known Bugs:**

<p>If the Update instruction results in an instruction error the checksum returned by the chipset for the Update command will be incorrect. This checksum error can be safely ignored. The GetHostIOError instruction should be issued to determine the cause of the instruction error. This bug will be corrected in the next release.</p>
---

<p>A command sent via the serial interface that results in an instruction error does not set the instruction error bit in the event status register. The error code will be set in the status byte of the serial response packet. This bug will be corrected in the next release.</p>
---

## **Changes/Fixes:**

### Command Changes

<p>Fixed a 1<sup>st</sup> generation command incompatibility. GET_ACTL_POS_ERR now behaves as expected. Previously this command returned CommandedPosition minus ActualPosition, Now it returns ActualPosition minus CommandedPosition.</p>
---

<p>GetPositionError was changed so that it returns the position error used during the servo filter calculation instead of generating a real-time position error.</p>
--

<p>The parameter range for SetTrackingWindow was increased from 0 to <math>2^{15}-1</math>, to 0 to <math>2^{16}-1</math>.</p>
--

### Profile Changes

<p>Corrected a problem in velocity contouring profile mode related to very large velocity values. Prior to this fix, if the velocity was greater than half of full-scale the velocity was interpreted as being negative instead of positive.</p>
--

<p>Corrected problems in s-curve profile mode when velocity or acceleration are set close to their maximum values.</p>
--

<p>Corrected a problem in s-curve profile mode related to move length. Previously, a move greater in length than half of full-scale position could only be done in the forward direction. Moves of greater than half of full-scale are now allowed in both directions.</p>
--

<p>Corrected problems in trapezoidal profile mode when velocity or acceleration are set close to their maximum values.</p>
--

<p>Enhanced electronic gear profile mode to generate 16.16 values for commanded velocity and acceleration when the position source for the master axis is set to commanded. This allows the feed-forward terms of the servo filter to be used on the slave axis.</p>
--

### Motor Command Output Changes

none

### Registers and Signals Changes

Corrected a problem in the activity status register related to the limit switches. Prior to this fix if both the positive and negative limit switches were active both switches would have to go inactive before either limit bit would be cleared. Now both bits are completely independent in operation.

### Miscellaneous Changes

The buffered velocity is now zeroed when a motion error occurs. In previous versions this value was not cleared and so an Update after a motion error could result in unexpected motion.

When the motor mode is set to off, either directly or via a motion error the current integration error is now set to zero.

## Version 2.4

### **Known Issues:**

If the IO chip HostRdy signal (pin 8) is used for chip busy detection, the first instruction sent to the chipset after a power on or reset may be ignored or may produce a checksum error. It is recommended that in this configuration a NoOperation command be sent to the chipset as the first instruction after a power on or reset. If the *ReadStatus* operation is used to check the HostRdy state this problem does not occur.

### **Known Bugs:**

If the Update instruction results in an instruction error the checksum returned by the chipset for the Update command will be incorrect. This checksum error can be safely ignored. The GetHostIOError instruction should be issued to determine the cause of the instruction error.

A command sent via the serial interface that results in an instruction error does not set the instruction error bit in the event status register. The error code will be set in the status byte of the serial response packet.

### **Changes/Fixes:**

#### Command Changes

Fixed a 1<sup>st</sup> generation command incompatibility. SET\_LMT\_SENSE now behaves as expected. Previously the parameter for this command was inverted in comparison to its expected behavior.

Fixed a 1<sup>st</sup> generation command incompatibility. SET\_MTN\_CMPLT\_BRK now behaves as expected. Previously the command did not work.

#### Profile Changes

Corrected a problem where an axis in electronic gear profile mode could not be stopped using the SetStopMode command or a breakpoint with its action set to AbruptStop.

Corrected a problem in velocity contouring profile mode related to limit switch

handling. Prior to this fix, if a limit switch had been activated an attempt to move out of the limit switch would generate an error.
Corrected a problem in velocity contouring profile mode related to a change in the motor mode. Prior to this fix, if the motor mode was turned off while and axis was in motion, the axis would resume motion when the motor mode was turned on. The axis will now remain stationary after the motor mode is turned on until a SetVelocity command has been issued with a value greater than zero.
Corrected a problem in trapezoidal profile mode where a deceleration value of 1 could cause the profile to overshoot the destination position. Following the overshoot the profile would change direction and correctly complete at the destination position.
Corrected a problem in trapezoidal profile mode related to changing direction on the fly. Prior to this fix, if the start velocity was non-zero and a change in the destination position on the fly would require a change in direction the profile would not change direction.
Corrected a problem in external profile mode related to the limit switches. Prior to this fix, if and axis was switched into external profile mode when a limit switch was active the chip would reset.
Corrected a problem in external profile mode related to the SetProfileMode command. Prior to this fix, if the SetProfileMode command was sent while the external profile mode was in operation, the profile would immediately stop.
Corrected a problem in the SCurve profile that would cause the trajectory to automatically restart if a ClearPositionError command was given after a move in the negative direction completed.
Corrected a problem in the SCurve profile that resulted in an overshoot in phase 1 of the profile for extremely short moves. This would in turn create an error in the trajectory in Phase 5/6, which caused the trajectory to overshoot but settle at the correct destination.
Corrected a problem in velocity contouring profile mode that could cause the motion complete bit to not be set if the axis profile mode was changed to velocity contouring on-the-fly from either SCurve or Trapezoidal profile mode.
Corrected a problem that prevented External Profile Mode from operating correctly with synchronous RAM. In prior versions the strobe signal was no being de-asserted between read cycles.
Corrected a problem in velocity contouring profile mode related to limit switch handling. Prior to this fix, if a limit switch had been activated an attempt to move out of the limit switch would generate an error.

#### Motor Command Output Changes

none
------

#### Registers and Signals Changes

The InMotion operation was changed to more accurately reflect the real state of motion. Now, after an Update command that will start motion the InMotion bit is immediately set. Previously the InMotion bit would not have been set for up to one chip cycle. This change corrected the situation where the MotionComplete bit would not be set for moves less than one step.
--

## Miscellaneous Changes

Corrected a problem with analog inputs where the default conversion timing resulted in out of specification results being generated by the analog to digital converter.

Corrected a problem where the analog input values were incorrectly stored as signed values during trace. This would result in them being sign extended when they should not be because they are unsigned. They are now stored correctly.

Corrected a problem where parallel encoder input on one channel could corrupt the position of axes using incremental encoder input.

Corrected a problem In multi-drop serial mode. In previous versions the chip did not set the SrlEnable line to the active state when a software "Reset" command was executed.

## Version 2.3

### **Known Issues:**

If the IO chip HostRdy signal (pin 8) is used for chip busy detection, the first instruction sent to the chipset after a power on or reset may be ignored or may produce a checksum error. It is recommended that in this configuration a NoOperation command be sent to the chipset as the first instruction after a power on or reset. If the *ReadStatus* operation is used to check the HostRdy state this problem does not occur.

### **Known Bugs:**

When SetInterruptMask is non-zero and the corresponding EventStatus bit is set, parallel host to I/O communications can become corrupted. The corruption causes the second and subsequent data reads to equal the first. If checksums are being read a checksum error will occur. If 32-bit values are being read the second word will equal the first which may result in unexpected data. If only 16-bit words are being read without checksum verification no errors will occur. This is the case with a normal interrupt service routine which sends GetInterruptAxis, GetEventStatus and ResetEventStatus.

If an Update or MultiUpdate is issued that would start the external profile motion AND a limit switch is currently active it will cause a chip reset. The workaround is to use any other profile mode to first move out of the limit before starting the external profile mode motion. If the chip enters a limit when the axis is already in external profile mode the chip behaves as expected, stopping motion if the LimitSwitchMode is set to enabled.

### **Changes/Fixes:**

#### Command Changes

none

#### Profile Changes

none

#### Registers and Signals Changes

none

### Miscellaneous Changes

When an axis had its encoder source set to parallel, any axes that followed it (1,2,3 or 4) and had an incremental encoder source would read an incorrect value. This is now fixed.

When an analog input channel was stored using the trace facility, the value was incorrectly stored as a signed instead of unsigned value. This is now fixed.

## Version 2.2

### **Known Bugs:**

none

### **Changes/Fixes:**

#### Command Changes

none

#### Profile Changes

Resolved an issue that prevented External Profile Mode from operating correctly with synchronous RAM.

#### Registers and Signals Changes

none

#### Miscellaneous Changes

none

## Version 2.1

### **Known Bugs:**

none

### **Changes/Fixes:**

#### Command Changes

none

#### Profile Changes

Changes made to External Profile Mode as detailed below for compatibility with the Pathfinder contouring library.

External profile mode now responds correctly to limit switches. It also responds correctly to an AbruptStop set through a breakpoint or through a SetStopMode AbruptStop command. SmoothStop has no effect when the chipset is in external profile mode.

External profile mode now stops if it encounters an entry in the time buffer where the value is zero. This is a more convenient way of halting this profile mode than using a

time or motion complete based breakpoint.

Registers and Signals Changes

none

Miscellaneous Changes

none

## Version 2.0

***Known Bugs:***

none

***Changes/Fixes:***

Command Changes

none

Profile Changes

none

Registers and Signals Changes

none

Miscellaneous Changes

Version 2.0 and above of this chipset use a new IO that facilitates higher parallel communication speed. There is now no additional CP overhead if the checksum is read. "Get" commands can see a speed improvement of up to 30%.

In multi-drop serial mode, the chip previously did not respond with the expected status packet when a software "Reset" command was executed. This is now fixed.