



Release Notes

MC28x0 version 2.5

Document last updated: 4/11/2007

Product names: MC2840, MC2820, MC2810
Source control archive name: mc28x0 version 2.5
IO device: MC2000IOU20
CP device build date: 4/4/2007

Description:

The MC2800 is a motion control processor for brushless 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.

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

Changed SetPhaseAngle so that it will only accept positive values. Prior to this fix, the command accepted negative values for phase angle, which is not valid.

Profile Changes

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 an 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 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

none

Miscellaneous Changes

none

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.

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 situations this signal should be connected to an external interrupt input on the host processor. The state of this signal is also reported by the *ReadStatus* operation but it is only accurate when the host is issuing actual chip commands and not simply polling *ReadStatus*.

Known Bugs:

none

Changes/Fixes:

Command Changes

Fixed a 1st generation command incompatibility. SET_MTN_CMPLT_BRK now behaves as expected. Previously the command did not work.

Profile Changes

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 external profile mode related to the limit switches. Prior to this fix, if an 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 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 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.

Registers and Signals Changes

none

Miscellaneous Changes

Resolved an issue with analog inputs where the default conversion timing resulted in out of specification results. This has been fixed.

Resolved an issue with the wrong number of microsteps being generated when an axis was set to microstepping commutation.

Version 2.3

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.