



Release Notes

MC58x20 version 1.5

Document last updated: 7/24/2005

Product names: MC58420, MC58320, MC58220, MC58120

Source control archive name: mc58x20 version 1.5

Device Checksum:

584200015.bin	0xFCB19418
583200015.bin	0xFCCC31BF
582200015.bin	0xFCE83E10
581200015.bin	0xFD0652F3

IO device: MC50000IOABN

Date of build: 7/14/2005

Description:

The MC58x20 is a motion control processor for servo and stepper 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 device 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.

The maximum incremental encoder input rate is limited to 5M counts/second in this release, not 10M counts/second as listed in the device specification. This limitation will be corrected in a future release of the device.

Incompatibilities with previous version:

none

Known Bugs:

Refer to current Magellan bug list located at
http://www.pmdcorp.com/support/release_notes.cfm

Changes/Fixes:

Command Changes

90141	Executing SetEncoderSource 0 for axis>1 after a power on or reset corrupts the actual position. Fixed.
90138	Breakpoint motor off behaves differently to SetMotorMode Off. Fixed.
90134	SetMotorMode On can result in a motion error if the position error limit has been exceeded during open loop operation. Fixed.
90133	SetMotorType returns corrupted data if the sample time is changed. Fixed.
90130	SetStepRange only works if motor type is step. Fixed.
90129	Multiple checksum reads produce unexpected results for "Set" commands. Fixed.
90119	"Set" command error – same as bug#90129

Communication Changes

90139	A good serial response incorrectly clears the HostIOError register. Fixed
90136	In serial multi-drop mode the chip response latency scales with baud rate. Fixed.
90135	In serial idle-line multi-drop mode there is occasional bad communication. Fixed.
90127	If SetCANMode is issued while the processor is in serial multi-drop mode the chip will stop responding. Fixed.
90109	Serial communication to the processor is lost if a command is sent with the incorrect (too many) number of bytes. Fixed.

Trajectory Generation Changes

90131	Incorrect position data could be used when switching to electronic gearing mode. Fixed.
90122	S-curve starting position symmetry. Fixed.
90114	In trapezoidal profile mode, if the trajectory velocity is set to below the value of the StartVelocity once a trajectory is running, the trajectory should stop. Fixed.
90113	In trapezoidal mode, if the trajectory velocity has been set to zero (SetVelocity 0) and the StartVelocity is > 0 issuing an update results in the trajectory jumping to -1. Fixed.
90112	StopMode has no effect if StartVelocity is greater than zero. Fixed.

PWM/DAC Signal Output Changes

90140	In 2-phase PWM5050 mode MagC does not produce a 50% signal. Fixed.
90120	2-phase PWM S/M output produces incorrect phase B magnitude signal. Fixed.

Step Signal Output Changes

	none
--	------

Servo Filter Changes

90142	An invalid overflow can occur in the second biquad filter. Fixed.
90125	The motor bias is applied before the biquad filters but it should be applied after them. Fixed.
90123	The motor limit is applied before the biquad filters but it should be applied after them. Fixed.
90111	The biquad filter does not round-off a negative output value correctly. Fixed.

Commutation Changes

90110	Index-based phase correction does not work when phase pre-scale is enabled. Fixed.
-------	--

Registers and Signals Changes

90128	When in S-curve profile mode the maximum velocity flag located in the activity status register is not reliable. Fixed.
-------	--

Miscellaneous Changes

90137	GetActualVelocity is not cleared for step motor type when the axis stops moving. Fixed.
90115	If a trigger based trace stop is programmed, the trace stops prior to the final sample being stored. Fixed.
90108	A limit switch event does not clear the position error. Fixed.

Version 1.4

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 device 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.
--

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 <i>ReadStatus</i> 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 <i>ReadStatus</i> operation the HostInterrupt bit will not reflect the current state of the signal.
--

The maximum incremental encoder input rate is limited to 5M counts/second in this release, not 10M counts/second as listed in the device specification. This limitation will be corrected in a future release of the device.
--

Incompatibilities with previous version:

Format of CAN command response packet has changed. See below.

SetMotorType functionality has changed slightly. See below.

DAC output address of phase A/B reversed. See below.
--

Known Bugs:

None

Changes/Fixes:

Command Changes

Fixed a problem with SetEncoderToStepRatio.

Fixed a problem in SetActualPosition for an axis that had its motor type configured as pulse and direction or microstep.
--

Fixed a problem in SetPrescaleMode that resulted in this command having no effect unless InitializePhase or SetCommutationMode was also issued.

Modified SetMotorType so that when it is called it results in the same default values as when a reset has occurred for the selected motor type.

Communication Changes

Corrected a problem in CAN communication that resulted in a command with the wrong number of data words returning the wrong error code.

Changed the placement of the error code returned via CAN. Previously the error code was contained in the first byte of the returned packet. Now the first byte is always zero and the second byte contains the error code, or zero if no error occurred.
--

Trajectory Generation Changes

Corrected a problem in s-curve profile mode when the fractional time for motion in segment 4 is less than 1.0 cycles.
Corrected a problem in s-curve profile mode for moves that start and end at a large negative position.
Corrected problems in velocity contouring profile mode for moves that have very large velocities or acceleration.
Corrected a problem in trapezoidal profile mode for long negative moves with large velocities.
Corrected a problem in trapezoidal profile mode for moves with low velocities but high acceleration/deceleration.
Corrected a problem in trapezoidal profile mode for very fast moves that resulted in corrupted values for the final commanded velocity and acceleration.
Corrected a problem in trapezoidal profile mode when start velocity was not zero.
Corrected a problem in trapezoidal profile mode when using a high start velocity.
Corrected a problem in trapezoidal profile mode for moves with high velocity and acceleration that resulted in overshoot at the end of the move.
Corrected a problem in trapezoidal profile mode for moves where the move was greater than half of full scale.

PWM/DAC Signal Output Changes

When DAC output is selected, phase A is now output to the base address and phase B is output to the base address + 1. This make the operation of the device inline with the documentation.
Corrected a problem with unipolar DAC output when the value being output is full scale negative.

Step Signal Output Changes

Corrected a problem in the AtRest signal which caused it to go active prior to the completion of a move.
--

Commutation Changes

Corrected a problem in algorithmic phase initialization where the motor command is not set to zero immediately after the final stage of the initialization process. This previously resulted in a small jump of the motor at the end of the initialization process.
Corrected a problem with algorithmic phase initialization that caused it to not initialize the motor.

Registers and Signals Changes

Corrected a problem in SrlEnable line when using multi-drop mode. Following a reset this signal will go low. Previously it was high which prevented the chip from communicating.
Corrected a problem in SrlEnable line when using multi-drop mode. Following a SetSerialPortMode command that selects multi-drop mode this signal will go low. Previously it was high which prevented the chip from communicating.

Miscellaneous Changes

none

Version 1.3

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 device 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.

The maximum incremental encoder input rate is limited to 5M counts/second in this release, not 10M counts/second as listed in the device specification. This limitation will be corrected in a future release of the device.

Known Bugs:

none

Changes/Fixes:

Command Changes

none

Communication Changes

none

Trajectory Generation Changes

Corrected a problem in s-curve profile mode where an overshoot in position occurred for a specific set of parameters.

Removed the enhancements made to electronic gear profile mode. This profile mode now operates as it did in version 1.1.

PWM/DAC Signal Output Changes

none

Step Signal Output Changes

none

Commutation Changes

none

Registers and Signals Changes

none

Miscellaneous Changes

Corrected a problem listed in known bugs for the previous version. The PhaseC trace variable was being incorrectly stored as unsigned. It is now stored as signed.

Version 1.2

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 device 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.

The maximum incremental encoder input rate is limited to 5M counts/second in this release, not 10M counts/second as listed in the device specification. This limitation will be corrected in a future release of the device.

Known Bugs:

When using the trace feature and one of the trace variables is set to PhaseC the stored value is stored as unsigned instead of signed. The result is that the stored value is not correct. The actual internal value of PhaseC is correct, only the traced value is incorrect.

Changes/Fixes:

Command Changes

SetTrackingWindow now accepts values in the range 0 to $2^{16}-1$. Previously the maximum value allowed was $2^{15}-1$.

SetPhaseOffset now accepts values in the range 0 to $2^{15}-1$. Previously the command accepted values between -2^{15} and $2^{15}-1$. Setting the phase offset to a negative value is not valid.

GetHostIOError now only clears the command error bit in the parallel status register if it was called from the parallel interface. Prior to this change, calling GetHostIOError using any of the communication interfaces would clear the command error bit.

Communication Changes

The axis number returned by a CAN event notification message is now 0-based. So, if the message originated from axis number 1, the axis number byte will contain the value 0, for axis number 2 the value will be 1, etc... This change makes the message consistent with other chip commands, all of which use a 0-based axis number.

Corrected a problem with CAN communication when an invalid axis number is specified. Prior to this fix, the motion processor would not generate the expected error if a command was sent using an invalid axis number.

Trajectory Generation Changes

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.

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

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 distance are now allowed in both directions.

Corrected problems in trapezoidal profile mode when velocity or acceleration are set close to their maximum values.

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.

PWM/DAC Signal Output Changes

none

Step Signal Output Changes

none

Commutation Changes

Resolved a problem with automatic phase correction. Prior to this fix, if phase correction mode was enabled it did not operate as expected.

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

Corrected a problem with parallel encoder feedback. Previously the encoder modulus value did not function as documented.

Corrected a problem in algorithmic phase initialization. Prior to this fix algorithmic phase init did not operate as expected.

Resolved an issue with Hall-based phase initialization. Prior to this fix Hall-based

phase initialization did not operate as expected.
Corrected a problem in SetActualPosition when it is used on an axis configured for step motor output. Prior to this fix this command resulted in the actual position being set to double the specified value.
GetHostIOError now works as expected over the serial interface. Prior to this fix a command error generated via the serial interface did not set the host error code.
The buffered velocity is now zeroed when a motion error occurs. Prior to this fix, the buffered velocity was not zeroed and so an Update after a motion error could result in unexpected motion.
The sample time can now be set as low as 51.2 microseconds for single axis products. Prior to this fix, the minimum allowable sample time for all products was 102.4 microseconds.

Version 1.1

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 device 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.
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 <i>ReadStatus</i> 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 <i>ReadStatus</i> operation the HostInterrupt bit will not reflect the current state of the signal.
The maximum incremental encoder input rate is limited to 5M counts/second in this release, not 10M counts/second as listed in the device specification. This limitation will be corrected in a future release of the device.

Known Bugs:

none

Changes/Fixes:

Command Changes

SetMotorType now behaves exactly the same as if the device had been reset and the dipswitches had been set to the selected motor type. That is, upon executing this command all device parameters will be set to their default condition for the selected motor type.

Corrected a problem in SetCANMode where the baud rate was not set in accordance with the documentation for this command. This command now behaves as documented.

Trajectory Generation Changes

none

PWM/DAC Signal Output Changes

none

Step Signal Output Changes

none

Registers and Signals Changes

none

Miscellaneous Changes

none

Bug#	Description	Affects part numbers	Status	Fixed in version
		5xxxx<=1.4	FIXED	1.5
		5xx2x<=1.4	FIXED	1.5
		5xxxx<=1.4	FIXED	1.5
		5xxxx<=1.4	FIXED	1.5
		5xxxx<=1.4	FIXED	1.5
		5xxxx<=1.4	FIXED	1.5