PerformanceMotionDevices
+1 781 674 9860
engineering@pmdcorp.com

Release Notes
MC73110 version 2.1

Document last updated: 10/11/2006

Product name: MC73110
Source control archive name: MC73110 v2.1
Date of build: 9/26/2006

Description:
The MC73110 is a motor control processor providing digital current and velocity control of DC Brushless motors. This document details bug fixes and changes for this release.

Known Issues:

Timing specs for "fast-mode I2C timing" are violated when initializing the temperature sensor. The recommended temperature sensor will however operate correctly.
The value returned for GetLoopIntegral does not have the same scaling for all the loops. For Velocity Integrator loop, it is raw sum(err). For Velocity and Current loops, it is sum(err)/256. However, in both cases, the value actually read is the value used for comparison to Integration Limit.
After a reset, a rising edge on the ~PWMOutputDisable pin causes the AmpDisable pin to go low even though GetConditionMask 1= 0x0000.

Changes/Fixes:
Velocity Integrator Loop Changes

Fixed - Velocity integrator integral sum continuously increased regardless of position error.

MC73110 version 2.0

Known Issues:

Timing specs for "fast-mode I2C timing" are violated when initializing the temperature sensor. The recommended temperature sensor will however operate correctly.
The value returned for GetLoopIntegral does not have the same scaling for all the loops. For Velocity Integrator loop, it is raw sum(err). For Velocity and Current loops, it is sum(err)/256. However, in both cases, the value actually read is the value used for comparison to Integration Limit.

Known Bugs:

None

Changes/Fixes:

Command Changes

Added enums of 2 and 3 in SetCommutationMode for FOC/sinusoidal and FOC/hall-based.
Added enum of 2 in SetVelocityFeedbackSource for Hall-based estimation.
Added Set/GetPWMFrequency command, with choices of 20Khz and 40Khz.
Added Set/GetSPISyncMode to support special resync mode in SPI.
Added enum of 2 in SetPWMOutputMode for 6-signal, 3rd leg floats (only applicable when non-FOC, Hall-based commutation).
More bits available in SetConditionMask (amplifier error from event status, overvoltage and undervoltage from activity status).
Added GetBusVoltage – gets the bus voltage.
Added Set/GetBusVoltageLimits – sets/gets the thresholds for overvoltage and undervoltage conditions.

Communication Changes

Changed default com setting from 2 stop bits to 1 stop bit (still 57.6K, no parity).
Fixed problem with setting the serial port mode using commands from flash/eeprom.

Trajectory Generation Changes

none

PWM Signal Output Changes

Added support for 40Khz PWM output.
Added support for “3rd leg floats” 6-signal mode for case of Hall-based commutation, non-FOC.

Velocity Integrator Loop Changes

Integrator sum is now zeroed whenever PWM disabled by mask.

Velocity Loop Changes

Integrator sum is now zeroed whenever PWM disabled by mask.
Added option of Hall-based velocity estimation.

Current Loop Changes

Integrator sums are now zeroed whenever PWM disabled by mask.
FOC is now supported as optional commutation/currentloops.

Commutation Changes

FOC is now supported as optional commutation/currentloops.
In Hall-based commutation, removed 0.866 factor.

Registers and Signals Changes

Added bit in activity status that reflects state of PWMDisable condition mask (bit 7, set when condition mask is TRUE).
Added bit in event status that indicates if problem found processing user commands from flash/eeprom (bit 15, set when error).
Added “amplifier error” (bit 8) of event status to the available bits in condition masks. This allows transient events (such as overvoltage/undervoltage) to be made sticky, and then disable amplifier or PWM.
Added overvoltage and undervoltage bits to activity status (bits 10 and 11, set when the thresholds are exceeded).

Miscellaneous Changes

Added “double conversion” of all ADC signals to remove potential offset problems.
Added support for special mode of SPI where recovering sync with the data stream is possible, given that the data stream meets some constraints.
Added error checking when reading/executing commands from flash/eeprom. If error occurs, process is aborted, and bit is set in event status. Amplifier cannot be enabled until this bit is cleared.
Added reading and filtering of the bus voltage and comparison to thresholds to determine overvoltage and undervoltage conditions.

MC73110 version 1.5

Known Issues:

If SPI command input stream gets out of sync, there is no way to resync it.
Timing specs for “fast-mode I2C timing” are violated when initializing the temperature sensor. The recommended temperature sensor will however operate correctly.
The value returned for GetLoopIntegral does not have the same scaling for all the loops. For Velocity Integrator loop, it is raw sum(err). For Velocity and Current loops, it is sum(err)/256. However, in both cases, the value actually read is the value used for comparison to Integration Limit.
Hall commutation drives to 0.866*MotorCommand, not MotorCommand, as expected.
If UserCommand data is corrupted or incomplete, chip may spin forever trying to parse the usercommand data.
Depending what voltages are present on the ADC inputs, small offsets may be introduced in the ADC conversion process.

Known Bugs:

none

Changes/Fixes:

Command Changes

Fixed problem with value returned in GetLoopCommand(0).
StoreUserData command is now only allowed when amplifier is disabled (amplifier

disable output is active).

Communication Changes

Fixed problems with SrlEnable and bus turnaround for SCI multidrop.
Changed upper 2 baud rate settings to 230.4K and 460.8K, instead of 250K and 416K.

Trajectory Generation Changes

none

PWM Signal Output Changes

Fixed problem with SetOutputMode(0) command.

Velocity Integrator Loop Changes

Integrator sum is now zeroed whenever Ki=0.

Velocity Loop Changes

Integrator sum is now zeroed whenever Ki=0.

Current Loop Changes

Integrator sums are now zeroed whenever Ki=0.

Commutation Changes

Changed relationship between “phase angle” and commutation waveforms.

Registers and Signals Changes

none

Miscellaneous Changes

none

MC73110 version 1.4

No official release done for this version.

MC73110 version 1.3

Known Issues:

Using GetLoopCommand(0) for the velocity integrator loop when velocity input is SPI or Analog gives erroneous results—it gets the value from the output of the velocity integrator rather than the input, as it should.
Timing specs for “fast-mode I2C timing” are violated when initializing the temperature sensor. The recommended temperature sensor will however operate correctly.
The value returned for GetLoopIntegral does not have the same scaling for all the loops. For Velocity Integrator loop, it is raw sum(err). For Velocity and Current loops, it is sum(err)/256. However, in both cases, the value actually read is the value used for comparison to Integration Limit.

Known Bugs:

none

Changes/Fixes:

Command Changes

none

Communication Changes

none

Trajectory Generation Changes

none

PWM Signal Output Changes

The ~PWMOutputDisable input signal now immediately tri-states the PWM outputs.

Velocity Integrator Loop Changes

Loop is now disabled when Amplifier Disable is active.

Velocity Loop Changes

Loop is now disabled when Amplifier Disable is active.

Current Loop Changes

Loops are now disabled when Amplifier Disable is active.
Fixed problem with ADC conversions that resulted in new data for current loops coming only every other cycle. Now the loops truly run at the specified rate.

Commutation Changes

Fixed problem with phase angle from Hall sensors being corrupted by non-Hall bits set in SignalSense register.

Registers and Signals Changes

none

Miscellaneous Changes

none

MC73110 version 1.2

Known Issues:

Using GetLoopCommand(0) for the velocity integrator loop when velocity input is SPI or Analog gives erroneous results—it gets the value from the output of the velocity integrator rather than the input, as it should.
Timing specs for “fast-mode I2C timing” are violated when initializing the temperature sensor. The recommended temperature sensor will however operate correctly.
The value returned for GetLoopIntegral does not have the same scaling for all the loops. For Velocity Integrator loop, it is raw sum(err). For Velocity and Current loops, it is sum(err)/256. However, in both cases, the value actually read is the value used for comparison to Integration Limit.

Known Bugs:

See above.

Changes/Fixes:

Command Changes

GetAnalog() now returns signed values that reflect the ADC reading added to the user-specified offset less the ADC offset (1/2 full scale). Previously, the values returned were raw unsigned ADC readings.
Added GetCommandedPos command—returns current TrgPos, the integrated velocity.
Fixed bugs in Set/Get SampleTime.
Fixed problem in SetMotorMode so that it does not cause glitches in operation if called to set the motor on when it is already on.
Disabled 0 (none) as option for SetPhaseCorrectionMode. Previously, it was allowed.
GetActualVelocity now returns the actual velocity regardless of the system state. Previously, the actual velocity was only computed when motor was on, and velocity loop was enabled.
Modified the way negation is done in current loop so that all Set/Get user functions return values of the correct polarity. Previously, some were the negative of what they should have been.
Modified values returned by GetLoopIntegral(1, 2, and 3) so that returned values are in same scaling as the parameter for SetLoopGain(integration limit) for those loops.
Modified SetVelocityScalar so it must have a parameter of 3 or more. Previously, it allowed 1 or more.

Communication Changes

none

Trajectory Generation Changes

Corrected a problem in Velocity Contouring that caused acceleration in the wrong direction when velocity was very high and negative.

PWM/DAC Signal Output Changes

none

Velocity Integrator Loop Changes

Improved phase delay and predictability of velocity integrator loop.
When the loop is disabled or motor is off, both the velocity integrator and the integrator in the position PID filter are explicitly cleared.

Velocity Loop Changes

Improved phase delay and predictability of velocity loop execution.
When the loop is disabled or motor is off, the integrator in the PID is now explicitly cleared.
Added saturation to velocity feedback calculation so that loop does not settle on erroneous velocity if scalar was set too large for velocity.
Added saturation on velocity feedforward calculation so that control is not lost when scalar is set too large for velocity trajectory.
Fixed problem with truncation in velocity feedback filter, which caused the filter output to never decay to zero when state was negative.
Fixed bias subtracted from ADC readings for AnalogCmd and TachFeedback. Was 0x8000, now its 0x7FE0 (1/2 full scale).
Changed initialization of VelocityScalar to be 3 instead of 1.

Current Loop Changes

When the loop is disabled or motor is off, the integrator in the PI is now explicitly cleared.

Commutation Changes

Improved Hall-based commutation, Hall-based phase correction, and Hall-based phase init.

Registers and Signals Changes

Cleaned up use of SignalSense so that users cannot inadvertently invert signals that should not be allowed by writing non-zero values to “reserved” bits of SignalSense.
Removed AmpErr from mask of applicable bits for Condition Masks.
Changed reset value of PWMDisableMask to 0x2000 (ESTOP enabled).
Fixed problem with signal sense of index bit—previously, SignalSense register setting did not actually affect the sense used.

Miscellaneous Changes

Fixed problem with I2C temperature reading—it now expects a signed value from the temperature sensor. Also, changed default reading if no sensor or problems reading sensor to 0x7FFF from 0xFFFF.
Corrected problems with reading of I2C temperature sensor. Also, spec for “fast-mode I2C timing” is now met when reading the sensor.
Increased acquisition window on ADC sample/hold circuit from 50ns to 200ns.
Fixed problem with ability of user commands (from Flash or EEPROM) to affect the

SampleTime. Previously, the SampleTime was reset from after initialization commands were executed from Flash or EEPROM.

MC73110 version 1.1

No official release done for this version.