

---

## C-Motion Functions Library

C-Motion library is written in ANSI C language. It contains low level motion functions that directly communicate with the PMD MC2xxx series of motion processors. The functions are also down compatible with the MC1xxx processor series.

The library routines can be used in the Windows or DOS environments. Based on these functions the user can develop a set of sophisticated procedures for an advanced motion system. However, with these functions alone one can already perform basic motion control.

The C-Motion routines are organized by categories. They perform a direct axis motion parameter setup, download and upload servo filter gains, provide motion processor performance status and control many other functions of brush, brushless, stepper and microstepper processor types.

### Instructions Summary by Functional Category

#### Breakpoints and Interrupts

<b>ClearInterrupt</b>	Reset interrupting line
<b>GetBreakpoint</b>	Get breakpoint type
<b>GetBreakpointValue</b>	Get breakpoint comparison value
<b>GetInterruptAxis</b>	Get the axes with pending interrupts
<b>GetInterruptMask</b>	Get interrupt mask
<b>SetBreakpoint</b>	Set breakpoint type
<b>SetBreakpointValue</b>	Set breakpoint comparison value
<b>SetInterruptMask</b>	Set interrupt mask

#### Commutation

<b>GetCommutationMode</b>	Get the commutation mode
<b>GetNumberPhases</b>	Get the number of phases
<b>GetPhaseAngle</b>	Get current commutation phase angle
<b>GetPhaseCommand</b>	Get the motor output command for a given phase A, B or C
<b>GetPhaseCorrectionMode</b>	Get phase correction mode
<b>GetPhaseCounts</b>	Get number of encoder counts per commutation cycle
<b>GetPhaseInitializeMode</b>	Get phase initialization mode
<b>GetPhaseInitializeTime</b>	Get the time parameters for the algorithmic phase initialization
<b>GetPhaseOffset</b>	Get phase offset value
<b>GetPhasePrescale</b>	Get phasing prescaler
<b>InitializePhase</b>	Perform phase initialization procedure
<b>SetCommutationMode</b>	Set the commutation mode (Hall-based, sinusoidal or microstepping)
<b>SetNumberPhases</b>	Set the number of phases (1, 2 or 3)
<b>SetPhaseAngle</b>	Set current commutation phase angle
<b>SetPhaseCorrectionMode</b>	Set phase correction mode (on or off)
<b>SetPhaseCounts</b>	Set number of encoder counts per commutation cycle
<b>SetPhaseInitializeMode</b>	Set phase initialization method (Hall-based or algorithmic)
<b>SetPhaseInitializeTime</b>	Set the time parameters for algorithmic phase initialization
<b>SetPhaseOffset</b>	Set phase offset value
<b>SetPhasePrescale</b>	Set commutation prescaler mode (enable or disable)

## Digital Servo Filter

<b>ClearPositionError</b>	Set position error to 0
<b>GetAutoStopMode</b>	Get auto stop mode
<b>GetDerivative</b>	Get the derivative of the error signal
<b>GetDerivativeTime</b>	Get derivative sampling time
<b>GetIntegral</b>	Get integrated position error value
<b>GetIntegrationLimit</b>	Get integration limit
<b>GetKaff</b>	Get acceleration feedforward gain
<b>GetKd</b>	Get derivative gain
<b>GetKi</b>	Get integral gain
<b>GetKout</b>	Get servo filter output scaler
<b>GetKp</b>	Get proportional gain
<b>GetKvff</b>	Get velocity feedforward gain
<b>GetMotorBias</b>	Get motor output bias
<b>GetMotorLimit</b>	Get motor output limit
<b>GetPositionError</b>	Get actual position error
<b>GetPositionErrorLimit</b>	Get position error limit
<b>SetAutoStopMode</b>	Set auto stop on position error (on or off)
<b>SetDerivativeTime</b>	Set derivative sampling time
<b>SetIntegrationLimit</b>	Set integration limit
<b>SetKaff</b>	Set acceleration feedforward gain
<b>SetKd</b>	Set derivative gain
<b>SetKi</b>	Set integral gain
<b>SetKout</b>	Set servo filter output scaler
<b>SetKp</b>	Set proportional gain
<b>SetKvff</b>	Set velocity feedforward gain
<b>SetMotorBias</b>	Set motor output bias
<b>SetMotorLimit</b>	Set motor output limit
<b>SetPositionErrorLimit</b>	Set maximum position error limit

## Encoder

<b>GetActualPosition</b>	Get the actual encoder position
<b>GetActualVelocity</b>	Get the actual encoder velocity
<b>GetCaptureSource</b>	Get capture source
<b>GetCaptureValue</b>	Get current axis position capture value and reset the capture
<b>GetEncoderModulus</b>	Get the full scale range of the parallel-word encoder
<b>GetEncoderSource</b>	Get encoder type
<b>GetEncoderToStepRatio</b>	Get encoder count to step ration
<b>SetActualPosition</b>	Set the actual encoder position
<b>SetCaptureSource</b>	Set capture source (home or index)
<b>SetEncoderModulus</b>	Set the full scale range of the parallel-word encoder
<b>SetEncoderSource</b>	Set encoder type (incremental or 16-bit parallel word)
<b>SetEncoderToStepRatio</b>	Set encoder count to step ratio

## External RAM

<b>GetBufferLength</b>	Get the length of a memory buffer
<b>GetBufferReadIndex</b>	Get the buffer read pointer for a particular buffer
<b>GetBufferStart</b>	Get the start location of a memory buffer
<b>GetBufferWriteIndex</b>	Get the buffer write pointer for a particular buffer
<b>ReadBuffer</b>	Read a long word value from a buffer memory location
<b>SetBufferLength</b>	Set the length of a memory buffer
<b>SetBufferReadIndex</b>	Set the buffer read pointer for a particular buffer
<b>SetBufferStart</b>	Set the start location of a memory buffer
<b>SetBufferWriteIndex</b>	Set the buffer write pointer for a particular buffer
<b>WriteBuffer</b>	Write a long word value to a buffer memory location

## Motor output

<b>GetCurrentMotorCommand</b>	Read the current motor command value
<b>GetMotorCommand</b>	Read buffered motor output command
<b>GetMotorMode</b>	Get motor loop mode
<b>GetOutputMode</b>	Get output mode
<b>SetMotorCommand</b>	Set direct value to motor output register
<b>SetMotorMode</b>	Set motor loop mode (on or off)
<b>SetOutputMode</b>	Set motor output mode (PWM sign-magnitude, PWM 50% or DAC)

## Profile Generation

<b>GetAcceleration</b>	Get acceleration limit
<b>GetCommandedAcceleration</b>	Get commanded (instantaneous desired) acceleration
<b>GetCommandedPosition</b>	Get commanded (instantaneous desired) position
<b>GetCommandedVelocity</b>	Get commanded (instantaneous desired) velocity
<b>GetDeceleration</b>	Get deceleration limit
<b>GetGearMaster</b>	Get electronic gear mode master axis and source
<b>GetGearRatio</b>	Get command electronic gear rate
<b>GetJerk</b>	Get jerk limit
<b>GetPosition</b>	Get destination position
<b>GetProfileMode</b>	Get current profile mode
<b>GetStartVelocity</b>	Get start velocity
<b>GetStop</b>	Get stop command: abrupt, smooth or none
<b>GetVelocity</b>	Get velocity limit
<b>MultiUpdate</b>	Multiple axis immediate parameter update
<b>SetAcceleration</b>	Set acceleration limit
<b>SetDeceleration</b>	Set deceleration limit
<b>SetGearMaster</b>	Set the master axis and source (actual or target-based)
<b>SetGearRatio</b>	Set command electronic gear ratio
<b>SetJerk</b>	Set jerk limit
<b>SetPosition</b>	Set position limit
<b>SetProfileMode</b>	Set profile mode (S-curve, trapezoidal, velocity-contouring or electronic gear)
<b>SetStartVelocity</b>	Set start velocity
<b>SetStop</b>	Set stop command: abrupt, smooth or none
<b>SetVelocity</b>	Set velocity limit
<b>Update</b>	Immediate parameter update

## Servo Loop Control

<b>GetAxisMode</b>	Get axis mode
<b>GetLimitSwitchMode</b>	Get limit switch mode
<b>GetMotionCompleteMode</b>	Get the motion complete mode
<b>GetSampleTime</b>	Get servo loop sample time
<b>GetSettleTime</b>	Get the axis-settled time
<b>GetSettleWindow</b>	Get the settle-window boundary value
<b>GetTime</b>	Get current chipset time (number of servo loops)
<b>GetTrackingWindow</b>	Get the tracking window boundary value
<b>SetAxisMode</b>	Set axis operation mode (enabled or disabled)
<b>SetLimitSwitchMode</b>	Set limit switching (on or off)
<b>SetMotionCompleteMode</b>	Set the motion complete mode (actual or target-based)
<b>SetSampleTime</b>	Set servo loop sample time
<b>SetSettleTime</b>	Set the axis settled-time
<b>SetSettleWindow</b>	Set the settle-window boundary
<b>SetTrackingWindow</b>	Set the tracking-window boundary

## Status register and AxisOut Indicator

<b>GetActivityStatus</b>	Get activity status
<b>GetAxisOutSource</b>	Get axis out signal monitor source
<b>GetEventStatus</b>	Get event status word
<b>GetSignal</b>	Get the current axis Signal Status register
<b>GetSignalSense</b>	Get the interpretation of the Signal Status bits
<b>ResetEventStatus</b>	Reset bits in event status word
<b>SetAxisOutSource</b>	Set axis out signal monitor source
<b>SetSignalSense</b>	Set the interpretation of the Signal Status bits

## Traces

<b>GetTraceCount</b>	Get the number of traced data points
<b>GetTraceMode</b>	Get the trace mode
<b>GetTracePeriod</b>	Get the trace period
<b>GetTraceStart</b>	Get the trace start condition
<b>GetTraceStatus</b>	Get the trace status word
<b>GetTraceStop</b>	Get the trace stop condition
<b>GetTraceVariable</b>	Get a trace variable setting
<b>SetTraceMode</b>	Set the trace mode (rolling or one-time)
<b>SetTracePeriod</b>	Set the trace period
<b>SetTraceStart</b>	Start the trace
<b>SetTraceStop</b>	Stop the trace
<b>SetTraceVariable</b>	Set variable (i.e. data) to be traced

## Miscellaneous

<b>GetDiagnosticPortMode</b>	Get the diagnostic port valid instruction mode
<b>GetHostIOError</b>	Get the most recent I/O error mode
<b>GetSerialPort</b>	Read serial-port configuration data
<b>GetVersion</b>	Get chipset software version information
<b>NoOperation</b>	Perform no operation, used to verify communications
<b>ReadIO</b>	Read user defined I/O value
<b>Reset</b>	Reset chipset
<b>SetDiagnosticPortMode</b>	Set the diagnostic port valid instruction mode (limited or full)
<b>SetSerialPort</b>	Set serial-port configuration data
<b>WriteIO</b>	Write user defined I/O value