Data Structures

MCAXISCONFIG
- int cbSize
- int ModuleType
- int ModuleLocation
- int MotorType
- int CaptureModes
- int CapturePoints
- int CaptureAndCompare
- double HighRate
- double MediumRate
- double LowRate
- double HighStepMin
- double HighStepMax
- double MediumStepMin
- double MediumStepMax
- double LowStepMin
- double LowStepMax

MCCONTOUR
- double VectorAccel
- double VectorDecel
- double VectorVelocity
- double VelocityOverride

MCCOMMUTATION
- int cbSize
- double PhaseA
- double PhaseB
- int Divisor
- int PreScale
- int Repeat

MCMOTIONEX
- int cbSize
- double Acceleration
- double Deceleration
- double Velocity
- double MinVelocity
- short int Direction
- double Torque
- double Deadband
- double DeadbandDelay
- short int StepSize
- short int Current
- short int HardLimitMode
- short int SoftLimitMode
- double SoftLimitLow
- double SoftLimitHigh
- short int EnableAmpFault

MCFILTEREX
- int cbSize
- double Gain
- double IntegralGain
- double IntegrationLimit
- int IntegralOption
- double DerivativeGain
- double DerSamplePeriod
- double FollowingError
- double VelocityGain
- double AccelGain
- double DecelGain
- double EncoderScaling
- int UpdateRate

MCJOG
- double Acceleration
- double MinVelocity
- double Deadband
- double Gain
- double Offset

MCPARAMEX
- int cbSize
- int ID
- int ControllerType
- int NumberAxes
- int MaximumAxes
- int MaximumModules
- int Precision
- int DigitalIO
- int AnalogInput
- int AnalogOutput
- int PointStorage
- int CanDoScaling
- int CanDoContouring
- int CanChangeProfile
- int CanChangeRates
- int SoftLimits
- int MultiTasking
- int AmpFault

MCSCALE
- double Constant
- double Offset
- double Rate
- double Scale
- double Zero
- double Time

Constants

Name	Value
MC_ALL_AXES	0
MC_ABSOLUTE	0
MC_BLOCK_COMPOUND	0
MC_BLOCK_TASK	1
MC_BLOCK_MACRO	2
MC_BLOCK_RESETM	3
MC_BLOCK_CANCEL	4
MC_BLOCK_CONTR_USER	5
MC_BLOCK_CONTR_LIN	6
MC_BLOCK_CONTR_CW	7
MC_BLOCK_CONTR_CCW	8
MC_CAPTURE_ACTUAL	16
MC_CAPTURE_ADVANCED	256
MC_CAPTURE_ERROR	32
MC_CAPTURE_OPTIMAL	64
MC_CAPTURE_TORQUE	128
MC_COMPARE_DISABLE	0
MC_COMPARE_ENABLE	1
MC_COMPARE_STATIC	1
MC_COMPARE_TOGGLE	2
MC_COMPARE_ONESHOT	3
MC_COMPARE_INVERT	0x0080
MC_COUNT_CAPTURE	1
MC_COUNT_COMPARE	2
MC_COUNT_CONTOUR	4
MC_COUNT_FILTER	8
MC_COUNT_FILTERMAX	16
MC_CURRENT_FULL	1
MC_CURRENT_HALF	2
MC_DIO_FIXED	256
MC_DIO_HIGH	4
MC_DIO_INPUT	1
MC_DIO_LATCH	16
MC_DIO_LATCHABLE	512
MC_DIO_LOW	8
MC_DIO_OUTPUT	2
MC_DIO_STEPPER	1024
MC_DIR_POSITIVE	1
MC_DIR_NEGATIVE	2
MC_IM_OPENLOOP	0
MC_IM_OPENLOOP	1
MC_INT_NORMAL	0
MC_INT_FREEZE	1
MC_INT_ZERO	2
MC_LIMIT_ABRUPT	4
MC_LIMIT_BOTH	3
MC_LIMIT_MINUS	2
MC_LIMIT_OFF	0
MC_LIMIT_PLUS	1
MC_LIMIT_SMOOTH	8
MC_LIMIT_INVERT	0x80
MC_LRN_POSITION	1
MC_LRN_TARGET	2
MC_MAX_ID	15
MC_MODE_CONTOUR	0
MC_MODE_GAIN	1
MC_MODE_POSITION	2
MC_MODE_TORQUE	3
MC_MODE_UNKNOWN	5
MC_MODE_VELOCITY	4
MC_OM_BIPOLAR	0
MC_OM_UNIPOLAR	1
MC_OM_PULSE_DIR	0
MC_OM_CW_CCW	1
MC_OM_BI_PWM	2
MC_OM_UNI_PWM	3
MC_OPEN_ASCII	1
MC_OPEN_BINARY	2
MC_OPEN_EXCLUSIVE	0x8000
MC_PHASE_STD	0
MC_PHASE_REV	1
MC_PROF_UNKNOWN	0
MC_PROF_TRAPEZOID	1
MC_PROF_SCURVE	2
MC_PROF_PARABOLIC	4
MC_RATE_LOW	1
MC_RATE_MEDIUM	2
MC_RATE_HIGH	4
MC_RELATIVE	1
MC_STAT_BUSY	0
MC_STAT_MTR_ENABLE	1
MC_STAT_MODE_VEL	2
MC_STAT_TRAJ	3
MC_STAT_DIR	4
MC_STAT_JOG_ENAB	5
MC_STAT_HOMED	6
MC_STAT_ERROR	7

Name	Value
MC_STAT_LOOK_INDEX	8
MC_STAT_LOOK_EDGE	9
MC_STAT_BREAKPOINT	10
MC_STAT_FOLLOWING	11
MC_STAT_AMP_ENABLE	12
MC_STAT_AMP_FAULT	13
MC_STAT_PLIM_ENAB	14
MC_STAT_PLIM_TRIP	15
MC_STAT_MLIM_ENAB	16
MC_STAT_MLIM_TRIP	17
MC_STAT_PSOFT_ENAB	18
MC_STAT_PSOFT_TRIP	19
MC_STAT_MSOFT_ENAB	20
MC_STAT_MSOFT_TRIP	21
MC_STAT_INP_INDEX	22
MC_STAT_INP_HOME	23
MC_STAT_INP_AMP	24
MC_STAT_INP_AUX	25
MC_STAT_INP_PLIM	26
MC_STAT_INP_MLIM	27
MC_STAT_INP_USER1	28
MC_STAT_INP_USER2	29
MC_STAT_PHASE	30
MC_STAT_FULL_STEP	31
MC_STAT_HALF_STEP	32
MC_STAT_JOGGING	33
MC_STAT_PJOG_ENAB	34
MC_STAT_PJOG_ON	35
MC_STAT_MJOG_ENAB	36
MC_STAT_MJOG_ON	37
MC_STAT_INP_PJOG	38
MC_STAT_INP_MJOG	39
MC_STAT_STOPPING	40
MC_STAT_PROG_DIR	41
MC_STAT_AT_TARGET	42
MC_STAT_ACCEL	43
MC_STAT_MODE_POS	44
MC_STAT_MODE_TRQE	45
MC_STAT_MODE_ARC	46
MC_STAT_MODE_CNTR	47
MC_STAT_MODE_SLAVE	48
MC_STAT_LMT_ABORT	49
MC_STAT_LMT_STOP	50
MC_STAT_CAPTURE	51
MC_STAT_RECORD	52
MC_STAT_SYNC	53
MC_STAT_MODE_LIN	54
MC_STAT_INDEX_FOUND	55
MC_STAT_POS_CAPT	56
MC_STAT_NULL	57
MC_STEP_FULL	1
MC_STEP_HALF	2
MC_TYPE_DOUBLE	4
MC_TYPE_FLOAT	3
MC_TYPE_LONG	2
MC_TYPE_NONE	0
MC_TYPE_REG	1
MC_TYPE_STRING	5
NO_CONTROLLER	0
NONE	0
DCXPC100	1
DCXAT100	2
DCXAT200	3
DC2PC100	4
DC2STN	5
DCXAT300	6
DCXPCI300	7
DCXPCI100	8
MC100	5
MC110	4
MC150	6
MC160	7
MC200	0
MC210	16
MC260	1
MC300	2
MC302	22
MC320	162
MC360	3
MC362	23
MC400	8
MC500	12
MF300	10
MF310	9
NO_MODULE	15
DC2SERVO	254
DC2STEPPER	255

MCAPI

Programming Interface

Quick Reference Card

Version 3.2

Precision MicroControl Corporation
2075-N Corte del Nogal
Carlsbad, CA 92009 • USA

Tel: (760) 930-0101
Fax: (760) 930-0222

Web: http://www.pmccorp.com
E-Mail: support@pmccorp.com

Function Summary

void MCAbort(*HCTRLR hCtlr, WORD wAxis*);
long int MCArcCenter(*HCTRLR hCtlr, WORD wAxis, short int type, double position*);
long int MCArcEndAngle(*HCTRLR hCtlr, WORD wAxis, short int type, double angle*);
long int MCArcRadius(*HCTRLR hCtlr, WORD wAxis, double radius*);
long int MCBlockBegin(*HCTRLR hCtlr, long int mode, long int num*);
long int MCBlockEnd(*HCTRLR hCtlr, long int* pTaskID*);
long int MCCancelTask(*HCTRLR hCtlr, long int TaskID*);
long int MCCaptureData(*HCTRLR hCtlr, WORD wAxis, long int points, double period, double delay*);
short int MCClose(*HCTRLR hCtlr*);
long int MCConfigureCompare(*HCTRLR hCtlr, WORD wAxis, double* values, long int num, double inc, long int mode, double period*);
short int MCConfigureDigitalIO(*HCTRLR hCtlr, WORD channel, WORD mode*);
long int MCContourDistance(*HCTRLR hCtlr, WORD wAxis, double Distance*);
long int MCContourPath(*HCTRLR hCtlr, WORD wAxis, WORD Mode, char* pBuffer*);
long int MCDecodeStatus(*HCTRLR hCtlr, DWORD status, long int bit*);
void MCDirection(*HCTRLR hCtlr, WORD wAxis, WORD Dir*);
long int MCEdgeArm(*HCTRLR hCtlr, WORD wAxis, double position*);
void MCEnableAxis(*HCTRLR hCtlr, WORD wAxis, short int state*);
long int MCEnableBacklash(*HCTRLR hCtlr, WORD wAxis, double backlash, short int state*);
long int MCEnableCapture(*HCTRLR hCtlr, WORD wAxis, long int count*);
long int MCEnableCompare(*HCTRLR hCtlr, WORD wAxis, long int flag*);
long int MCEnableDigitalFilter(*HCTRLR hCtlr, WORD wAxis, long int state*);
void MCEnableDigitalIO(*HCTRLR hCtlr, WORD channel, short int state*);
void MCEnableGearing(*HCTRLR hCtlr, WORD wAxis, WORD Master, double ratio, short int state*);
void MCEnableJog(*HCTRLR hCtlr, WORD wAxis, short int state*);
void MCEnableSync(*HCTRLR hCtlr, WORD wAxis, short int state*);
void MCErrorNotify(*HWND hWnd, HCTRLR hCtlr, DWORD ErrorMask*);
long int MCFindAuxEncIdx(*HCTRLR hCtlr, WORD wAxis, double Position*);
long int MCFindEdge(*HCTRLR hCtlr, WORD wAxis, double position*);
long int MCFindIndex(*HCTRLR hCtlr, WORD wAxis, double position*);
long int MCGetAccelerationEx(*HCTRLR hCtlr, WORD wAxis, double* acceleration*);
WORD MCGetAnalog(*HCTRLR hCtlr, WORD Channel*);
long int MCGetAuxEncIdxEx(*HCTRLR hCtlr, WORD wAxis, double* index*);
long int MCGetAuxEncPosEx(*HCTRLR hCtlr, WORD wAxis, double* position*);
long int MCGetAxisConfiguration(*HCTRLR hCtlr, WORD wAxis, MCAXISCONFIG* pAxisCfg*);
long int MCGetBreakpointEx(*HCTRLR hCtlr, WORD wAxis, double* breakpoint*);
long int MCGetCaptureData(*HCTRLR hCtlr, WORD wAxis, long int type, long int start, long int points, double* pData*);
long int MCGetConfigurationEx(*HCTRLR hCtlr, MCPARAMEX* pParam*);
short int MCGetContourConfig(*HCTRLR hCtlr, WORD wAxis, MCCONTOUR* pCntr*);
long int MCGetContouringCount(*HCTRLR hCtlr, WORD wAxis*);
long int MCGetCount(*HCTRLR hCtlr, WORD wAxis, long int type, long int* count*);
long int MCGetDecelerationEx(*HCTRLR hCtlr, WORD wAxis, double* Deceleration*);
long int MCGetDigitalFilter(*HCTRLR hCtlr, WORD wAxis, double* coeff, long int num, long int* actual*);
short int MCGetError(*HCTRLR hCtlr*);
short int MCGetDigitalIO(*HCTRLR hCtlr, WORD channel*);
long int MCGetDigitalIOConfig(*HCTRLR hCtlr, WORD channel, WORD* mode*);
short int MCGetError(*HCTRLR hCtlr*);
long int MCGetFilterConfigEx(*HCTRLR hCtlr, WORD wAxis, MCFILTEREx* pFilter*);
long int MCGetFollowingError(*HCTRLR hCtlr, WORD wAxis, double* error*);

Function Summary

long int MCGetGain(*HCTRLR hCtlr, WORD wAxis, double* gain*);
long int MCGetIndexEx(*HCTRLR hCtlr, WORD wAxis, double* index*);
long int MCGetInstalledModules(*HCTRLR hCtlr, int modules[ ], int size*);
short int MCGetJogConfig(*HCTRLR hCtlr, WORD wAxis, MCJOG* pJog*);
long int MCGetLimits(*HCTRLR hCtlr, WORD wAxis, short int* HardLimitMode, short int* SoftLimitMode, double* SoftLimitLow, double* SoftLimitHigh*);
long int MCGetModuleInputMode(*HCTRLR hCtlr, WORD wAxis, long int* mode*);
long int MCGetMotionConfigEx(*HCTRLR hCtlr, WORD wAxis, MCMOTIONEX* pMtn*);
long int MCGetOperatingMode(*HCTRLR hCtlr, WORD wAxis, long int* mode*);
long int MCGetOptimalEx(*HCTRLR hCtlr, WORD wAxis, double* optimal*);
long int MCGetPositionEx(*HCTRLR hCtlr, WORD wAxis, double* position*);
long int MCGetProfile(*HCTRLR hCtlr, WORD wAxis, WORD* profile*);
long int MCGetRegister(*HCTRLR hCtlr, long int register, void* value, long int type*);
short int MCGetScale(*HCTRLR hCtlr, WORD wAxis, MCSCALE* pScale*);
long int MCGetServoOutputPhase(*HCTRLR hCtlr, WORD wAxis, WORD* phase*);
DWORD MCGetStatus(*HCTRLR hCtlr, WORD wAxis*);
long int MCGetTargetEx(*HCTRLR hCtlr, WORD wAxis, double* target*);
long int MCGetTorque(*HCTRLR hCtlr, WORD wAxis, double* torque*);
long int MCGetVectorVelocity(*HCTRLR hCtlr, WORD wAxis, double* velocity*);
long int MCGetVelocityEx(*HCTRLR hCtlr, WORD wAxis, double* velocity*);
DWORD MCGetVersion(*HCTRLR hCtlr*);
long int MCGoEx(*HCTRLR hCtlr, WORD wAxis, double param*);
void MCGoHome(*HCTRLR hCtlr, WORD wAxis*);
long int MCIndexArm(*HCTRLR hCtlr, WORD wAxis, double position*);
long int MCIsAtTarget(*HCTRLR hCtlr, WORD wAxis, double timeout*);
long int MCIsDigitalFilter(*HCTRLR hCtlr, WORD wAxis*);
long int MCIsEdgeFound(*HCTRLR hCtlr, WORD wAxis, double timeout*);
long int MCIsIndexFound(*HCTRLR hCtlr, WORD wAxis, double timeout*);
long int MCIsStopped(*HCTRLR hCtlr, WORD wAxis, double timeout*);
long int MCLearnPoint(*HCTRLR hCtlr, WORD wAxis, long int index, WORD mode*);
void MCMacroCall(*HCTRLR hCtlr, WORD macro*);
void MCMoveAbsolute(*HCTRLR hCtlr, WORD wAxis, double position*);
void MCMoveRelative(*HCTRLR hCtlr, WORD wAxis, double distance*);
long int MCMoveToPoint(*HCTRLR hCtlr, WORD wAxis, long int index*);
HCTRLR MCOpen(*short int hCtlr, WORD mode, LPCSTR pszName*);
long int MCReopen(*HCTRLR hCtlr, WORD NewMode*);
long int MCRepeat(*HCTRLR hCtlr, long int count*);
void MCReset(*HCTRLR hCtlr, WORD wAxis*);
void MCSetAcceleration(*HCTRLR hCtlr, WORD wAxis, double rate*);
void MCSetAnalog(*HCTRLR hCtlr, WORD wChannel, WORD value*);
void MCSetAuxEncPos(*HCTRLR hCtlr, WORD wAxis, double position*);
long int MCSetCommutation(*HCTRLR hCtlr, WORD wAxis, MCCOMMUTATION* pCommutation*);
short int MCSetContourConfig(*HCTRLR hCtlr, WORD wAxis, MCCONTOUR* pCntr*);
void MCSetDeceleration(*HCTRLR hCtlr, WORD wAxis, double rate*);
long int MCSetDigitalFilter(HCTRLR hCtlr, WORD wAxis, double* coeff, long int num);
long int MCSetFilterConfigEx(*HCTRLR hCtlr, WORD wAxis, MCFILTEREX* pFilter*);
long int MCSetGain(*HCTRLR hCtlr, WORD wAxis, double gain*);
short int MCSetJogConfig(*HCTRLR hCtlr, WORD wAxis, MCJOG* pJog*);
long int MCSetLimits(*HCTRLR hCtlr, WORD wAxis, short int HardLimitMode, short int SoftLimitMode, double SoftLimitLow, double SoftLimitHigh*);
long int MCSetModuleInputMode(*HCTRLR hCtlr, WORD wAxis, long int mode*);
void MCSetModuleOutputMode(*HCTRLR hCtlr, WORD wAxis, WORD mode*);
long int MCSetMotionConfigEx(*HCTRLR hCtlr, WORD wAxis, MCMOTIONEX* pMtn*);
void MCSetOperatingMode(*HCTRLR hCtlr, WORD wAxis, WORD ctl, WORD mode*);

Function Summary

void MCSetPosition(*HCTRLR hCtlr, WORD wAxis, double position*);
void MCSetProfile(*HCTRLR hCtlr, WORD wAxis, WORD mode*);
long int MCSetRegister(*HCTRLR hCtlr, long int register, void* value, long int type*);
short int MCSetScale(*HCTRLR hCtlr, WORD wAxis, MCSCALE* pScale*);
void MCSetServoOutputPhase(*HCTRLR hCtlr, WORD wAxis, WORD phase*);
long int MCSetTimeoutEx(*HCTRLR hCtlr, double TimeOut, double* OldTimeOut*);
long int MCSetTorque(*HCTRLR hCtlr, WORD wAxis, double torque*);
long int MCSetVectorVelocity(*HCTRLR hCtlr, WORD wAxis, double velocity*);
void MCSetVelocity(*HCTRLR hCtlr, WORD wAxis, double velocity*);
void MCStop(*HCTRLR hCtlr, WORD wAxis*);
long int MCTranslateErrorEx(*short int error, LPSTR pBuffer, long int length*);
void MCWait(*HCTRLR hCtlr, double period*);
void MCWaitForDigitalIO(*HCTRLR hCtlr, WORD channel, short int state*);
long int MCWaitForEdge(*HCTRLR hCtlr, WORD wAxis, short int state*);
long int MCWaitForIndex(*HCTRLR hCtlr, WORD wAxis*);
void MCWaitForPosition(*HCTRLR hCtlr, WORD wAxis, double position*);
void MCWaitForRelative(*HCTRLR hCtlr, WORD wAxis, double distance*);
void MCWaitForStop(*HCTRLR hCtlr, WORD wAxis, double period*);
void MCWaitForTarget(*HCTRLR hCtlr, WORD wAxis, double period*);
long int MCDLG_AboutBox(*HWND hWnd, LPCSTR pTitle, long int BitmapID*);
LPCSTR MCDLG_CommandFileExt(*long int type, long int flags, LPSTR pBuffer, long int length*);
long int MCDLG_ConfigureAxis(*HWND hWnd, HCTRLR hCtlr, WORD wAxis, long int flags, LPCSTR pTitle*);
LPCSTR MCDLG_ControllerDescEx(*long int type, long int flags, LPSTR pBuffer, long int length*);
long int MCDLG_ControllerInfo(*HWND hWnd, HCTRLR hCtlr, long int flags, LPCSTR szTitle*);
long int MCDLG_DownloadFile(*HWND hWnd, HCTRLR hCtlr, long int flags, LPCSTR zFileName*);
long int MCDLG_Initialize(*void*);
long int MCDLG_ListControllers(*short int IDArray[ ], short int size*);
LPCSTR MCDLG_ModuleDescEx(*long int type, long int flags, LPSTR pBuffer, long int length*);
long int MCDLG_RestoreAxis(*HCTRLR hCtlr, WORD wAxis, long int flags, LPCSTR PrivateIniFile*);
long int MCDLG_RestoreDigitalIO(*HCTRLR hCtlr, WORD StartChannel, WORD EndChannel, LPCSTR PrivateIniFile*);
long int MCDLG_SaveAxis(*HCTRLR hCtlr, WORD wAxis, long int flags, LPCSTR PrivateIniFile*);
long int MCDLG_SaveDigitalIO(*HCTRLR hCtlr, WORD StartChannel, WORD EndChannel, LPCSTR PrivateIniFile*);
long int MCDLG_Scaling(*HWND hWnd, HCTRLR hCtlr, WORD wAxis, long int flags, LPCSTR pTitle*);
short int MCDLG_SelectController(*HWND hWnd, short int CurrentID, long int flags, LPCSTR pTitle*);
long int pmccmdex(*HCTRLR hCtlr, WORD wAxis, WORD cmd, void* argument, long int type*);
short int pmcgetc(*HCTRLR hCtlr*);
void pmcgetram(*HCTRLR hCtlr, WORD offset, void* pBuffer, short int bytes*);
short int pmcgets(*HCTRLR hCtlr, char* pBuffer, short int bytes*);
short int pmcputc(*HCTRLR hCtlr, short int Char*);
void pmcputram(*HCTRLR hCtlr, WORD offset, void* pBuffer, short int bytes*);
short int pmcputs(*HCTRLR hCtlr, char* pBuffer*);
short int pmcrdy(*HCTRLR hCtlr*);
long int pmcrpyex(*HCTRLR hCtlr, void* Reply, long int type*);