Data Structures

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

MCMOTION
- double Acceleration
- double Deceleration
- double Velocity
- double MinVelocity
- short int Direction
- double Gain
- 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
- short int Rate

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

MCFILTER
- double DerivativeGain
- double DerSamplePeriod
- double IntegralGain
- double IntegrationLimit
- double VelocityGain
- double AccelGain
- double DecelGain
- double FollowingError

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

MCPARAM
- short int ID
- short int ControllerType
- short int NumberAxes
- short int DigitalIO
- short int AnalogInput
- short int AnalogOutput
- short int AxisType[8]
- short int PointStorage
- short int CanDoScaling
- short int CanDoContouring
- short int CanChangeProfile
- short int CanChangeRates
- short int SoftLimits
- short int MultiTasking
- short int AmpFault

Error Codes

#	Description
0	no error
1	no controller at this ID
2	driver out of handles
3	cannot open - exclusive mode
4	already open in different mode
5	mode not supported
6	couldn't initialize the driver
7	controller not present
8	memory allocation error
9	reserved
10	reserved
11	feature not supported
12	function is obsolete
13	invalid controller handle
14	invalid window handle
15	axis number out of range
16	axis doesn't support feature
17	cannot use MC_ALL_AXES
18	parameter was out of range
19	illegal constant value
20	unexpected or unknown reply
21	controller failed to reply
22	reply size incorrect
23	wrong axis for reply
24	command / reply out of sync
25	controller timeout
26	block mode error
27	RS232 error

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_CENTER_ABS	0
MC_CENTER_REL	1
MC_CURRENT_FULL	1
MC_CURRENT_HALF	2
MC_DATA_ACTUAL	0
MC_DATA_ERROR	2
MC_DATA_OPTIMAL	1
MC_DATA_TORQUE	3
MC_DIO_FLASH	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_DIR_POSITIVE	1
MC_DIR_NEGATIVE	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_VELOCITY	4
MC_OM_BIPOLAR	0
MC_OM_UNIPOLAR	1
MC_OM_PULSE_DIR	0
MC_OM_CW_CCW	1
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
MC_STAT_LOOK_INDEX	8
MC_STAT_LOOK_EDGE	9
MC_STAT_BREAKPOINT	10
MC_STAT_FOLLOWING	11
MC_STAT_AMP_ENABLE	12

Name	Value
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_STEP_FULL	1
MC_STEP_HALF	2
MC_TYPE_LONG	2
MC_TYPE_DOUBLE	4
NO_CONTROLLER	0
NONE	0
DCXPC100	1
DCXAT100	2
DCXAT200	3
DC2PC100	4
DC2STN	5
DCXAT300	6
MC100	5
MC110	4
MC150	6
MC160	7
MC200	0
MC210	16
MC260	1
MC300	2
MC310	18
MC360	3
MC400	8
MC500	12
MF300	10
MF310	9
NO_MODULE	15
DC2SERVO	254
DC2STEPPER	255

MCAPI

Programming Interface

Quick Reference Card

Version 2.2

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 nType, double Position*);
long int MCArcEndAngle(*HCTRLR hCtlr, WORD wAxis, short int nType, double Angle*);
long int MCArcRadius(*HCTRLR hCtlr, WORD wAxis, double Radius*);
long int MCArcCenter(*HCTRLR hCtlr, WORD wAxis, short int nType, double Position*);
long int MCBlockBegin(*HCTRLR hCtlr, long int lMode, long int lNum*);
long int MCBlockEnd(*HCTRLR hCtlr, long int* *lTaskID*);
long int MCCancelTask(*HCTRLR hCtlr, long int lTaskID*);
long int MCCaptureData(*HCTRLR hCtlr, WORD wAxis, long int lPoints, double Period, double Delay*);
short int MCClose(*HCTRLR hCtlr*);
short int MCConfigureDigitalIO(*HCTRLR hCtlr, WORD wChannel, WORD wMode*);
long int MCContourDistance(*HCTRLR hCtlr, WORD wAxis, double Distance*);
long int MCContourPath(*HCTRLR hCtlr, WORD wAxis, WORD wMode, char* lpBuffer*);
long int MCDecodeStatus(*HCTRLR hCtlr, DWORD dwStatus, long int lBit*);
void MCDirection(*HCTRLR hCtlr, WORD wAxis, WORD wDir*);
void MCEnableAxis(*HCTRLR hCtlr, WORD wAxis, short int bState*);
long int MCEnableBacklash(*HCTRLR hCtlr, WORD wAxis, double Backlash, short int bState*);
void MCEnableDigitalIO(*HCTRLR hCtlr, WORD wChannel, short int bState*);
void MCEnableGearing(*HCTRLR hCtlr, WORD wAxis, WORD wMaster, double ratio, short int bState*);
void MCEnableJog(*HCTRLR hCtlr, WORD wAxis, short int bState*);
void MCEnableSync(*HCTRLR hCtlr, WORD wAxis, short int bState*);
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 wChannel*);
long int MCGetAuxEncIdxEx(*HCTRLR hCtlr, WORD wAxis, double* Index*);
long int MCGetAuxEncPosEx(*HCTRLR hCtlr, WORD wAxis, double* Position*);
long int MCGetBreakpointEx(*HCTRLR hCtlr, WORD wAxis, double* Breakpoint*);
long int MCGetCaptureData(*HCTRLR hCtlr, WORD wAxis, long int lType, long int lStart, long int lPoints, double* lpData*);
void MCGetConfiguration(*HCTRLR hCtlr, MCPARAM* lpParam*);
short int MCGetContourConfig(*HCTRLR hCtlr, WORD wAxis, MCCONTOUR* lpContour*);
long int MCGetContouringCount(*HCTRLR hCtlr, WORD wAxis*);
long int MCGetDecelerationEx(*HCTRLR hCtlr, WORD wAxis, double* Deceleration*);
short int MCGetError(*HCTRLR hCtlr*);
short int MCGetDigitalIO(*HCTRLR hCtlr, WORD wChannel*);
short int MCGetDigitalIOConfig(*HCTRLR hCtlr, WORD wChannel, WORD* wMode*);
short int MCGetFilterConfig(*HCTRLR hCtlr, WORD wAxis, MCFILTER* lpFilter*);
long int MCGetFollowingError(*HCTRLR hCtlr, WORD wAxis, double* Error*);
long int MCGetGain(*HCTRLR hCtlr, WORD wAxis, double* Gain*);
long int MCGetIndexEx(*HCTRLR hCtlr, WORD wAxis, double* Index*);

Function Summary

short int MCGetJogConfig(*HCTRLR hCtlr, WORD wAxis, MCJOG* lpJog*);
long int MCGetLimits(*HCTRLR hCtlr, WORD wAxis, short int* HardLimitMode, short int* SoftLimitMode, double* SoftLimitLow, double* SoftLimitHigh*);
short int MCGetMotionConfig(*HCTRLR hCtlr, WORD wAxis, MCMOTION* lpMotion*);
long int MCGetOptimalEx(*HCTRLR hCtlr, WORD wAxis, double* Optimal*);
long int MCGetPositionEx(*HCTRLR hCtlr, WORD wAxis, double* Position*);
ong int MCGetProfile(*HCTRLR hCtlr, WORD wAxis, WORD* wProfile*);
long int MCGetRegister(*HCTRLR hCtlr, long int nRegister, void* Value, long int nType*);
short int MCGetScale(*HCTRLR hCtlr, WORD wAxis, MCSCALE* lpScale*);
long int MCGetServoOutputPhase(*HCTRLR hCtlr, WORD wAxis, WORD* wPhase*);
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 MCIsStoppedt(*HCTRLR hCtlr, WORD wAxis, double Timeout*);
long int MCLearnPoint(*HCTRLR hCtlr, WORD wAxis, long int lIndex, WORD wMode*);
void MCMacroCall(*HCTRLR hCtlr, WORD wMacro*);
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 lIndex*);
HCTRLR MCOpen(*short int hCtlr, WORD wMode, LPCSTR lpszName*);
long int MCReopen(*HCTRLR hCtlr, WORD wNewMode*);
long int MCRepeat(*HCTRLR hCtlr, long int nCount*);
void MCReset(*HCTRLR hCtlr, WORD wAxis*);
void MCSetAcceleration(*HCTRLR hCtlr, WORD wAxis, double Rate*);
void MCSetAnalog(*HCTRLR hCtlr, WORD wChannel, WORD wValue*);
void MCSetAuxEncPos(*HCTRLR hCtlr, WORD wAxis, double Position*);
short int MCSetContourConfig(*HCTRLR hCtlr, WORD wAxis, MCCONTOUR* lpContour*);
void MCSetDeceleration(*HCTRLR hCtlr, WORD wAxis, double Rate*);
short int MCSetFilterConfig(*HCTRLR hCtlr, WORD wAxis, MCFILTER* lpFilter*);
long int MCSetGain(*HCTRLR hCtlr, WORD wAxis, double Gain*);
short int MCSetJogConfig(*HCTRLR hCtlr, WORD wAxis, MCJOG* lpJog*);
long int MCSetLimits(*HCTRLR hCtlr, WORD wAxis, short int HardLimitMode, short int SoftLimitMode, double SoftLimitLow, double SoftLimitHigh*);
void MCSetModuleOutputMode(*HCTRLR hCtlr, WORD wAxis, WORD wMode*);
short int MCSetMotionConfig(*HCTRLR hCtlr, WORD wAxis, MCMOTION* lpMotion*);
void MCSetOperatingMode(*HCTRLR hCtlr, WORD wAxis, WORD wControlAxis, WORD mode*);
void MCSetPosition(*HCTRLR hCtlr, WORD wAxis, double Position*);
void MCSetProfile(*HCTRLR hCtlr, WORD wAxis, WORD wMode*);
long int MCSetRegister(*HCTRLR hCtlr, long int nRegister, void* Value, long int nType*);

Function Summary

short int MCSetScale(*HCTRLR hCtlr, WORD wAxis, MCSCALE* lpScale*);
void MCSetServoOutputPhase(*HCTRLR hCtlr, WORD wAxis, WORD wPhase*);
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 nError, LPSTR szBuffer, long int nLength*);
void MCWait(*HCTRLR hCtlr, double Period*);
void MCWaitForDigitalIO(*HCTRLR hCtlr, WORD wChannel, short int bState*);
long int MCWaitForEdge(*HCTRLR hCtlr, WORD wAxis, short int bState*);
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 szTitle, long int BitmapID*);
long int MCDLG_ConfigureAxis(*HWND hWnd, HCTRLR hCtlr, WORD wAxis, long int iFlags, LPCSTR szTitle*);
LPCSTR MCDLG_ControllerDesc(*short int nType, LPSTR szBuffer, long int nLength*);
long int MCDLG_ControllerInfo(*HWND hWnd, HCTRLR hCtlr, long int iFlags, LPCSTR szTitle*);
long int MCDLG_DownloadFile(*HWND hWnd, HCTRLR hCtlr, long int iFlags, LPCSTR szFileName*);
long int MCDLG_Initialize(*void*);
long int MCDLG_ListControllers(*short int IDArray[], short int Size*);
LPCSTR MCDLG_ModuleDesc(*short int nType, LPSTR szBuffer, long int nLength*);
long int MCDLG_RestoreAxis(*HCTRLR hCtlr, WORD wAxis, long int iFlags, LPCSTR PrivateIniFile*);
long int MCDLG_RestoreDigitalIO(*HCTRLR hCtlr, WORD wStartChannel, WORD wEndChannel, LPCSTR PrivateIniFile*);
long int MCDLG_SaveAxis(*HCTRLR hCtlr, WORD wAxis, long int iFlags, LPCSTR PrivateIniFile*);
long int MCDLG_SaveDigitalIO(*HCTRLR hCtlr, WORD wStartChannel, WORD wEndChannel, LPCSTR PrivateIniFile*);
long int MCDLG_Scaling(*HWND hWnd, HCTRLR hCtlr, WORD wAxis, long int iFlags, LPCSTR szTitle*);
short int MCDLG_SelectController(*HWND hWnd, short int CurrentID, long int iFlags, LPCSTR szTitle*);

short int pmccmd(*HCTRLR hCtlr, short int nBytes, void* lpBuffer*);
long int pmccmdex(*HCTRLR hCtlr, WORD wAxis, WORD wCmd, void* Argument, long int nType*);
short int pmcgetc(*HCTRLR hCtlr*);
void pmcgetram(*HCTRLR hCtlr, WORD wOffset, void* lpBuffer, short int nBytes*);
short int pmcgets(*HCTRLR hCtlr, char* lpszBuffer, short int nBytes*);
short int pmcputc(*HCTRLR hCtlr, short int nChar*);
void pmcputram(*HCTRLR hCtlr, WORD wOffset, void* lpBuffer, short int nBytes*);
short int pmcputs(*HCTRLR hCtlr, char* lpszBuffer*);
short int pmcrdy(*HCTRLR hCtlr*);
short int pmcrpy(*HCTRLR hCtlr, short int nBytes, char* lpBuffer*);
long int pmcrpyex(*HCTRLR hCtlr, void* Reply, long int nType*);