

XPS Unified Command Interface V2.3.0

Based on XPS Unified V1.8.x

<i>Dll .NET function return</i>	22
<i>IsDeviceConnected</i>	23
<i>OpenInstrument</i>	24
<i>CloseInstrument</i>	25
<i>SetTimeout</i>	26
<i>WriteToInstrument</i>	27
<i>GetVerCommand</i>	28
<i>MultipleAxesPDSAAxisByAxisExecution</i>	30
<i>MultipleAxesPDSASpiralStepExecution</i>	31
<i>MultipleAxesPDSASpiralContinuousExecution</i>	32
<i>MultipleAxesPDSADichotomyExecution</i>	33
<i>MultipleAxesPDSARasterExecution</i>	34
<i>MultipleAxesPDSAEscaladeStepExecution</i>	35
<i>MultipleAxesPDSAEscaladeContinuousExecution</i>	36
<i>GetVerCommand</i>	38
<i>SetGantryMode</i>	39
<i>GetGantryMode</i>	40
<i>InitializeAndHomeXY</i>	41
<i>InitializeAndHomeY</i>	42
<i>InitializeAndHomeX</i>	43
<i>MoveAbsolute</i>	44
<i>MoveSlice</i>	45
<i>WaitMotionEnd</i>	46
<i>AbortMove</i>	47
<i>GetAccParams</i>	48
<i>SetAccParams</i>	49
<i>GetVelParams</i>	50

<i>SetVelParams</i>	51
<i>GetXVelParams</i>	52
<i>SetXVelParams</i>	53
<i>GetYVelParams</i>	54
<i>SetYVelParams</i>	55
<i>SetJogVelocity</i>	56
<i>GetJogVelocity</i>	57
<i>SetJogAcceleration</i>	58
<i>GetJogAcceleration</i>	59
<i>StartJog</i>	60
<i>EndJog</i>	61
<i>GetCurrentPosition</i>	62
<i>SetZone</i>	63
<i>GetZone</i>	64
<i>SetVarX</i>	65
<i>SetVarXSecondary</i>	66
<i>SetVarY</i>	67
<i>GetVarX</i>	68
<i>GetVarXSecondary</i>	69
<i>GetVarY</i>	70
<i>DataCollectionBufferReset</i>	71
<i>DataCollectionBufferAndTimeReset</i>	72
<i>DataCollectionRequest</i>	73
<i>DataCollectionTimeStampGet</i>	74
<i>DataCollectionTimeStampReset</i>	75
<i>SetPiston</i>	76
<i>GetPistonState</i>	77
<i>SetBrake</i>	78
<i>GetBrakeState</i>	79
<i>ZygoGetVerInterfero</i>	80
<i>ZygoAmplitudeGet</i>	81
<i>ZygoADCDiagnosticStatusGet</i>	82

<i>ZygoPositionGet</i>	83
<i>ZygoResetX</i>	84
<i>ZygoResetY</i>	85
<i>ZygoSetOffsetX</i>	86
<i>ZygoSetOffsetY</i>	87
<i>ZygoSetPEGParams</i>	88
<i>ZygoGetPEGLastCommunicationTime</i>	89
<i>FirmwareVersionGet</i>	90
<i>FirmwareBuildVersionNumberGet</i>	91
<i>ExternalModuleTemplateNumberGet</i>	92
<i>InstallerVersionGet</i>	93
<i>Reboot</i>	94
<i>RestartApplication</i>	95
<i>ControllerMotionKernelTimeLoadGet</i>	96
<i>ControllerRTTimeGet</i>	97
<i>ControllerSlaveStatusGet</i>	98
<i>ControllerSlaveStatusStringGet</i>	99
<i>ControllerSynchronizeCorrectorISR</i>	100
<i>ControllerStatusGet</i>	101
<i>ControllerStatusRead</i>	102
<i>ControllerStatusStringGet</i>	103
<i>ElapsedTimeGet</i>	104
<i>ErrorStringGet</i>	105
<i>Login</i>	106
<i>LoginS</i>	107
<i>CloseAllOtherSockets</i>	108
<i>HardwareDriverAndStageGet</i>	109
<i>HardwareDateAndTimeGet</i>	110
<i>HardwareDateAndTimeSet</i>	111
<i>FileScriptHistoryRename</i>	112
<i>FileGatheringRename</i>	113
<i>INTServitudesStatusGet</i>	114

<i>INTServitudesCommandGet</i>	115
<i>TCLScriptExecute</i>	116
<i>TCLScriptExecuteAndWait</i>	117
<i>TCLScriptExecuteWithPriority</i>	118
<i>TCLScriptKill</i>	119
<i>TCLScriptKillAll</i>	120
<i>TCLScriptRunningListGet</i>	121
<i>TimerGet</i>	122
<i>TimerSet</i>	123
<i>CleanTmpFolder</i>	124
<i>CleanCoreDumpFolder</i>	125
<i>GlobalArrayGet</i>	126
<i>GlobalArraySet</i>	127
<i>DoubleGlobalArrayGet</i>	128
<i>DoubleGlobalArraySet</i>	129
<i>PositionerMagneticTrackPositionAtHomeGet</i>	130
<i>PositionerMagneticTrackPositionAtHomeSet</i>	131
<i>PositionerMotorDualSinForceBalanceGet</i>	132
<i>PositionerMotorDualSinForceBalanceSet</i>	133
<i>ZygoConnectToServer</i>	134
<i>ZygoReset</i>	135
<i>ZygoReadWord</i>	136
<i>ZygoReadLong</i>	137
<i>ZygoWriteWord</i>	138
<i>ZygoWriteLong</i>	139
<i>ZygoSendAndReceive</i>	140
<i>ZygoDisconnectFromServer</i>	141
<i>ZygoEthernetCommunicationStatusGet</i>	142
<i>ZygoInterferometerStatusGet</i>	143
<i>ZygoStartInterferometer</i>	144
<i>ZygoStartBoardP2</i>	145
<i>ZygoErrorStatusStringGet</i>	146

<i>ZygoErrorStatusGet</i>	147
<i>ZygoStatusStringGet</i>	148
<i>ZygoStatusGet</i>	149
<i>ZygoRegisterSet</i>	150
<i>ZygoRegisterGet</i>	151
<i>EventAdd</i>	152
<i>EventGet</i>	153
<i>EventRemove</i>	154
<i>EventWait</i>	155
<i>EventActionSetAndStart</i>	156
<i>EventExtendedConfigurationTriggerSet</i>	158
<i>EventExtendedConfigurationTriggerGet</i>	159
<i>EventExtendedConfigurationActionSet</i>	160
<i>EventExtendedConfigurationActionGet</i>	161
<i>EventExtendedStart</i>	162
<i>EventExtendedAllGet</i>	163
<i>EventExtendedGet</i>	164
<i>EventExtendedRemove</i>	165
<i>EventExtendedWait</i>	166
<i>GatheringConfigurationGet</i>	167
<i>GatheringConfigurationSet</i>	168
<i>GatheringCurrentNumberGet</i>	169
<i>GatheringCurrentIndexGet</i>	170
<i>GatheringStopAndSave</i>	171
<i>GatheringDataAcquire</i>	172
<i>GatheringDataGet</i>	173
<i>GatheringDataMultipleLinesGet</i>	174
<i>GatheringReset</i>	175
<i>GatheringRun</i>	176
<i>GatheringRunAppend</i>	177
<i>GatheringStop</i>	178
<i>GatheringExternalConfigurationSet</i>	179

<i>GatheringExternalConfigurationGet</i>	180
<i>GatheringExternalCurrentNumberGet</i>	181
<i>GatheringExternalDataGet</i>	182
<i>GatheringExternalStopAndSave</i>	183
<i>GPIOAnalogGet</i>	184
<i>GPIOAnalogSet</i>	185
<i>GPIOAnalogGainGet</i>	186
<i>GPIOAnalogGainSet</i>	187
<i>GPIOAnalogRangeConfigurationGet</i>	188
<i>GPIOAnalogRangeConfigurationSet</i>	189
<i>GPIODigitalGet</i>	190
<i>GPIODigitalSet</i>	191
<i>GPIODigitalPulseWidthGet</i>	192
<i>GPIODigitalPulseWidthSet</i>	193
<i>GroupAccelerationCurrentGet</i>	194
<i>GroupAccelerationSetpointGet</i>	195
<i>GroupAnalogTrackingModeEnable</i>	196
<i>GroupAnalogTrackingModeDisable</i>	197
<i>GroupCorrectorOutputGet</i>	198
<i>GroupCurrentFollowingErrorGet</i>	199
<i>GroupHomeSearch</i>	200
<i>GroupHomeSearchAndRelativeMove</i>	201
<i>GroupInitialize</i>	202
<i>GroupInitializeNoEncoderReset</i>	203
<i>GroupInitializeWithEncoderCalibration</i>	204
<i>GroupGantryModeGet</i>	205
<i>GroupGantryModeSet</i>	206
<i>GroupInterlockDisable</i>	207
<i>GroupInterlockEnable</i>	208
<i>GroupJogParametersSet</i>	209
<i>GroupJogParametersGet</i>	210
<i>GroupJogCurrentGet</i>	211

<i>GroupJogModeEnable</i>	212
<i>GroupJogModeDisable</i>	213
<i>GroupKill</i>	214
<i>GroupMotionDisable</i>	215
<i>GroupMotionEnable</i>	216
<i>GroupMotionStatusGet</i>	217
<i>GroupMoveAbort</i>	218
<i>GroupExternalProfilerEnable</i>	219
<i>GroupExternalProfilerDisable</i>	220
<i>GroupMoveAbortFast</i>	221
<i>GroupGoToNotReferenced</i>	222
<i>GroupMoveAbsolute</i>	223
<i>GroupMoveEndWait</i>	224
<i>GroupMoveRelative</i>	225
<i>GroupMoveRelativeSimulated</i>	226
<i>GroupPositionCurrentGet</i>	227
<i>GroupPositionSetpointGet</i>	228
<i>GroupPositionTargetGet</i>	229
<i>GroupReferencingActionExecute</i>	230
<i>GroupReferencingStart</i>	231
<i>GroupReferencingStop</i>	232
<i>GroupStatusGet</i>	233
<i>GroupStatusStringGet</i>	234
<i>GroupVelocityCurrentGet</i>	235
<i>GroupVelocitySetpointGet</i>	236
<i>ExternalInterlockErrorStringGet</i>	237
<i>ExternalInterlockGroupErrorGet</i>	238
<i>ExternalInterlockLiftPinErrorGet</i>	239
<i>KillAll</i>	240
<i>GroupInitializeFromRegistration</i>	241
<i>GroupReadyAtPosition</i>	242
<i>GroupKillWithRegistration</i>	243

<i>PositionerDriverStatusGet</i>	244
<i>PositionerDriverStatusStringGet</i>	245
<i>PositionerErrorGet</i>	246
<i>PositionerErrorRead</i>	247
<i>PositionerErrorStringGet</i>	248
<i>PositionerHardwareStatusGet</i>	249
<i>PositionerHardwareStatusStringGet</i>	250
<i>PositionersEncoderIndexDifferenceGet</i>	251
<i>PositionerGantryEndReferencingPositionGet</i>	252
<i>PositionerStageParameterGet</i>	253
<i>PositionerStageParameterSet</i>	254
<i>PositionerCorrectorTypeGet</i>	255
<i>PositionerCorrectorFilterListGet</i>	256
<i>PositionerWarningFollowingErrorSet</i>	257
<i>PositionerWarningFollowingErrorGet</i>	258
<i>PositionerCompensationPositionFilterSet</i>	259
<i>PositionerCompensationPositionFilterGet</i>	260
<i>PositionerCorrectorPlantFeedForwardDelaySet</i>	261
<i>PositionerCorrectorPlantFeedForwardDelayGet</i>	262
<i>PositionerCompensationDisturbanceDisable</i>	263
<i>PositionerCompensationDisturbanceEnable</i>	264
<i>PositionerCompensationDisturbanceFileLoad</i>	265
<i>PositionerCompensationDisturbanceStatusGet</i>	266
<i>PositionerCorrectorAutoTuning</i>	267
<i>PositionerAccelerationAutoScaling</i>	268
<i>PositionerExcitationSignalGet</i>	269
<i>PositionerExcitationSignalSet</i>	270
<i>PositionerExcitationSignalCorrectorOutSet</i>	271
<i>PositionerCurrentVelocityAccelerationFiltersSet</i>	272
<i>PositionerCurrentVelocityAccelerationFiltersGet</i>	273
<i>PositionerEncoderAmplitudeValuesGet</i>	274
<i>PositionerEncoderCalibrationParametersGet</i>	275

<i>PositionerEncoderLinearCorrectionGet</i>	276
<i>PositionerEncoderLinearCorrectionSet</i>	277
<i>PositionerRawEncoderPositionGet</i>	278
<i>PositionerCorrectorPIDFFAccelerationSet</i>	279
<i>PositionerCorrectorPIDFFAccelerationGet</i>	281
<i>PositionerCorrectorPIDAccelerationFilterSet</i>	283
<i>PositionerCorrectorPIDAccelerationFilterGet</i>	284
<i>PositionerCorrectorDamperFilterSet</i>	285
<i>PositionerCorrectorDamperFilterGet</i>	286
<i>PositionerCorrectorPostFFSet</i>	287
<i>PositionerCorrectorPostFFGet</i>	288
<i>PositionerCorrectorExcitationSignalGainSet</i>	289
<i>PositionerCorrectorExcitationSignalGainGet</i>	290
<i>PositionerCorrectorPIDBaseSet</i>	291
<i>PositionerCorrectorPIDBaseGet</i>	292
<i>PositionerPositionCompareScanAccelerationLimitGet</i>	293
<i>PositionerPositionCompareScanAccelerationLimitSet</i>	294
<i>PositionerCorrectorPIDDualFFVoltageSet</i>	295
<i>PositionerCorrectorPIDDualFFVoltageGet</i>	297
<i>PositionerCorrectorPIDFFVelocitySet</i>	299
<i>PositionerCorrectorPIDFFVelocityGet</i>	301
<i>PositionerCorrectorPIPositionSet</i>	303
<i>PositionerCorrectorPIPositionGet</i>	304
<i>PositionerCorrectorDualSet</i>	305
<i>PositionerCorrectorDualGet</i>	306
<i>PositionerCompensationDualLoopNotchFilterSet</i>	307
<i>PositionerCompensationDualLoopNotchFilterGet</i>	308
<i>PositionerCompensationDualLoopPhaseCorrectionFilterSet</i>	309
<i>PositionerCompensationDualLoopPhaseCorrectionFilterGet</i>	310
<i>PositionerDriverFiltersGet</i>	311
<i>PositionerDriverFiltersSet</i>	312
<i>PositionerDriverPositionOffsetsGet</i>	313

<i>PositionerPreCorrectorExcitationSignalGet</i>	314
<i>PositionerPreCorrectorExcitationSignalSet</i>	315
<i>PositionerBacklashSet</i>	316
<i>PositionerBacklashGet</i>	317
<i>PositionerBacklashEnable</i>	318
<i>PositionerBacklashDisable</i>	319
<i>PositionerMotionDoneGet</i>	320
<i>PositionerMotionDoneSet</i>	321
<i>PositionerHardInterpolatorFactorGet</i>	322
<i>PositionerHardInterpolatorFactorSet</i>	323
<i>PositionerHardInterpolatorPositionGet</i>	324
<i>PositionerPositionCompareGet</i>	325
<i>PositionerPositionCompareSet</i>	326
<i>PositionerPositionCompareEnable</i>	327
<i>PositionerPositionCompareDisable</i>	328
<i>PositionerPositionCompareAquadBAwaysEnable</i>	329
<i>PositionerPositionCompareAquadBWindowedGet</i>	330
<i>PositionerPositionCompareAquadBWindowedSet</i>	331
<i>PositionerTimeFlasherGet</i>	332
<i>PositionerTimeFlasherSet</i>	333
<i>PositionerTimeFlasherEnable</i>	334
<i>PositionerTimeFlasherDisable</i>	335
<i>PositionerPositionComparePulseParametersGet</i>	336
<i>PositionerPositionComparePulseParametersSet</i>	337
<i>PositionerPositionCompareAquadBPrescalerSet</i>	338
<i>PositionerPositionCompareAquadBPrescalerGet</i>	339
<i>PositionerCompensatedPCOAbort</i>	340
<i>PositionerCompensatedPCOCurrentStatusGet</i>	341
<i>PositionerCompensatedPCOEnable</i>	342
<i>PositionerCompensatedPCOFromFile</i>	343
<i>PositionerCompensatedPCOLoadToMemory</i>	344
<i>PositionerCompensatedPCOMemoryReset</i>	345

<i>PositionerCompensatedPCOPrepare</i>	346
<i>PositionerCompensatedPCOSet</i>	347
<i>PositionerCompensatedFastPCOAbort</i>	348
<i>PositionerCompensatedFastPCOCurrentStatusGet</i>	349
<i>PositionerCompensatedFastPCOEnable</i>	350
<i>PositionerCompensatedFastPCOFromFile</i>	351
<i>PositionerCompensatedFastPCOLoadToMemory</i>	352
<i>PositionerCompensatedFastPCOMemoryReset</i>	353
<i>PositionerCompensatedFastPCOPrepare</i>	354
<i>PositionerCompensatedFastPCOSet</i>	355
<i>PositionerCompensatedFastPCOPulseParametersGet</i>	356
<i>PositionerCompensatedFastPCOPulseParametersSet</i>	357
<i>PositionerCompensationEncoderNotchFilterSet</i>	358
<i>PositionerCompensationEncoderNotchFilterGet</i>	359
<i>PositionerCompensationNotchFilterSet</i>	360
<i>PositionerCompensationNotchFilterGet</i>	361
<i>PositionerCompensationPhaseCorrectionFilterSet</i>	362
<i>PositionerCompensationPhaseCorrectionFilterGet</i>	363
<i>PositionerCorrectorNotchFiltersSet</i>	364
<i>PositionerCorrectorNotchFiltersGet</i>	365
<i>PositionerCompensationPreFeedForwardFrequencyNotchFilterGet</i>	366
<i>PositionerCompensationPreFeedForwardFrequencyNotchFilterSet</i>	367
<i>PositionerCompensationPreFeedForwardSpatialNotchFilterGet</i>	368
<i>PositionerCompensationPreFeedForwardSpatialNotchFilterSet</i>	369
<i>PositionerCompensationPreFeedForwardPhaseCorrectionFilterGet</i>	370
<i>PositionerCompensationPreFeedForwardPhaseCorrectionFilterSet</i>	371
<i>PositionerCompensationFrequencyNotchsGet</i>	372
<i>PositionerCompensationFrequencyNotchsSet</i>	374
<i>PositionerCompensationSpatialPeriodicNotchsGet</i>	375
<i>PositionerCompensationSpatialPeriodicNotchsSet</i>	377
<i>PositionerCompensationPostExcitationLowPassFilterGet</i>	379
<i>PositionerCompensationPostExcitationLowPassFilterSet</i>	380

<i>PositionerCompensationPostExcitationFrequencyNotchFilterGet</i>	381
<i>PositionerCompensationPostExcitationFrequencyNotchFilterSet</i>	382
<i>PositionerCompensationPostExcitationNotchModeFilterGet</i>	383
<i>PositionerCompensationPostExcitationNotchModeFilterSet</i>	384
<i>PositionerCompensationPostExcitationPhaseCorrectionFilterGet</i>	385
<i>PositionerCompensationPostExcitationPhaseCorrectionFilterSet</i>	386
<i>PositionerCompensationLowPassTwoFilterGet</i>	387
<i>PositionerCompensationLowPassTwoFilterSet</i>	388
<i>PositionerCompensationNotchModeFiltersGet</i>	389
<i>PositionerCompensationNotchModeFiltersSet</i>	390
<i>PositionerCompensationPhaseCorrectionFiltersGet</i>	391
<i>PositionerCompensationPhaseCorrectionFiltersSet</i>	392
<i>PositionerAnalogTrackingPositionParametersGet</i>	393
<i>PositionerAnalogTrackingPositionParametersSet</i>	394
<i>PositionerAnalogTrackingVelocityParametersGet</i>	395
<i>PositionerAnalogTrackingVelocityParametersSet</i>	396
<i>PositionerJogMaximumVelocityAndAccelerationGet</i>	397
<i>PositionerMaximumVelocityAndAccelerationGet</i>	398
<i>PositionerUserTravelLimitsGet</i>	399
<i>PositionerUserTravelLimitsSet</i>	400
<i>PositionerSGammaExactVelocityAjustedDisplacementGet</i>	401
<i>PositionerSGammaMoveResultGet</i>	402
<i>PositionerSGammaParametersGet</i>	404
<i>PositionerSGammaParametersSet</i>	405
<i>PositionerSGammaVelocityAndAccelerationSet</i>	406
<i>PositionerSGammaPreviousMotionTimesGet</i>	407
<i>MultipleAxesPTVerification</i>	408
<i>MultipleAxesPTVerificationResultGet</i>	409
<i>MultipleAxesPTExecution</i>	410
<i>MultipleAxesPTParametersGet</i>	411
<i>MultipleAxesPTPulseOutputSet</i>	412
<i>MultipleAxesPTPulseOutputGet</i>	413

<i>MultipleAxesPTLoadToMemory</i>	414
<i>MultipleAxesPTResetInMemory</i>	415
<i>XYPTVerification</i>	416
<i>XYPTVerificationResultGet</i>	417
<i>XYPTExecution</i>	418
<i>XYPTParametersGet</i>	419
<i>XYPTPulseOutputSet</i>	420
<i>XYPTPulseOutputGet</i>	421
<i>XYPTLoadToMemory</i>	422
<i>XYPTResetInMemory</i>	423
<i>TZPTVerification</i>	424
<i>TZPTVerificationResultGet</i>	425
<i>TZPTExecution</i>	426
<i>TZPTParametersGet</i>	427
<i>TZPTPulseOutputSet</i>	428
<i>TZPTPulseOutputGet</i>	429
<i>TZPTLoadToMemory</i>	430
<i>TZPTResetInMemory</i>	431
<i>HexapodLegsPTVerification</i>	432
<i>HexapodLegsPTVerificationResultGet</i>	433
<i>HexapodLegsPTExecution</i>	434
<i>HexapodLegsPTParametersGet</i>	435
<i>HexapodLegsPTPulseOutputSet</i>	436
<i>HexapodLegsPTPulseOutputGet</i>	437
<i>HexapodLegsPTSave</i>	438
<i>HexapodLegsPTLoadToMemory</i>	439
<i>HexapodLegsPTResetInMemory</i>	440
<i>MultipleAxesPVTVerification</i>	441
<i>MultipleAxesPVTVerificationResultGet</i>	442
<i>MultipleAxesPVTExecution</i>	443
<i>MultipleAxesPVTPParametersGet</i>	444
<i>MultipleAxesPVTPPulseOutputSet</i>	445

<i>MultipleAxesPVTPulseOutputGet</i>	446
<i>MultipleAxesPVTLoadToMemory</i>	447
<i>MultipleAxesPVTResetInMemory</i>	448
<i>XYPVTVerification</i>	449
<i>XYPVTVerificationResultGet</i>	450
<i>XYPVTExecution</i>	451
<i>XYPVTParametersGet</i>	452
<i>XYPVTPulseOutputSet</i>	453
<i>XYPVTPulseOutputGet</i>	454
<i>XYPVTLoadToMemory</i>	455
<i>XYPVTResetInMemory</i>	456
<i>TZPVTVerification</i>	457
<i>TZPVTVerificationResultGet</i>	458
<i>TZPVTExecution</i>	459
<i>TZPVTParametersGet</i>	460
<i>TZPVTPulseOutputSet</i>	461
<i>TZPVTPulseOutputGet</i>	462
<i>TZPVTLoadToMemory</i>	463
<i>TZPVTResetInMemory</i>	464
<i>HexapodLegsPVTVerification</i>	465
<i>HexapodLegsPVTVerificationResultGet</i>	466
<i>HexapodLegsPVTEExecution</i>	467
<i>HexapodLegsPVTPParametersGet</i>	468
<i>HexapodLegsPVTPulseOutputSet</i>	469
<i>HexapodLegsPVTPulseOutputGet</i>	470
<i>HexapodLegsPVTSave</i>	471
<i>HexapodLegsPVTLoadToMemory</i>	472
<i>HexapodLegsPVTRResetInMemory</i>	473
<i>XYLineArcVerification</i>	474
<i>XYLineArcVerificationResultGet</i>	475
<i>XYLineArcExecution</i>	476
<i>XYLineArcParametersGet</i>	477

<i>XYLineArcPulseOutputSet</i>	478
<i>XYLineArcPulseOutputGet</i>	479
<i>XYZSplineVerification</i>	480
<i>XYZSplineVerificationResultGet</i>	481
<i>XYZSplineExecution</i>	482
<i>XYZSplineParametersGet</i>	483
<i>XYZSplinePulseOutputSet</i>	484
<i>XYZSplinePulseOutputGet</i>	485
<i>SingleAxisSlaveModeEnable</i>	486
<i>SingleAxisSlaveModeDisable</i>	487
<i>SingleAxisSlaveParametersSet</i>	488
<i>SingleAxisSlaveParametersGet</i>	489
<i>SpindleSlaveModeEnable</i>	490
<i>SpindleSlaveModeDisable</i>	491
<i>SpindleSlaveParametersSet</i>	492
<i>SpindleSlaveParametersGet</i>	493
<i>GroupSpinParametersSet</i>	494
<i>GroupSpinParametersGet</i>	495
<i>GroupSpinCurrentGet</i>	496
<i>GroupSpinModeStop</i>	497
<i>MultipleAxesDisableShutter</i>	498
<i>MultipleAxesScanPositions</i>	499
<i>MultipleAxesGetShutterPositions</i>	501
<i>MultipleAxesSetShutterPositions</i>	502
<i>MultipleAxesTraceNextScan</i>	503
<i>XYClampDisable</i>	504
<i>XYClampEnable</i>	505
<i>PositionerFeedforwardAccDisable</i>	506
<i>PositionerFeedforwardAccEnable</i>	507
<i>PositionerFeedforwardAccGet</i>	508
<i>PositionerFeedforwardAccSet</i>	509
<i>PositionerFeedforwardAccStatusGet</i>	510

<i>PositionerFeedforwardPositionDisable</i>	511
<i>PositionerFeedforwardPositionEnable</i>	512
<i>PositionerFeedforwardPositionGet</i>	513
<i>PositionerFeedforwardPositionSet</i>	514
<i>PositionerFeedforwardPositionStatusGet</i>	515
<i>GroupBrakeSet</i>	516
<i>GroupBrakeStateGet</i>	517
<i>XYCrossTalkCompensationMotorDecouplingSet</i>	518
<i>XYCrossTalkCompensationMotorDecouplingGet</i>	519
<i>XYGroupPositionPCORawEncoderGet</i>	520
<i>XYGroupPositionCorrectedProfilerGet</i>	521
<i>XYDualCorrectorMappingGet</i>	522
<i>XYDualCorrectorMappingSet</i>	523
<i>XYGantryDestressingMappingGet</i>	524
<i>XYGantryDestressingMappingSet</i>	525
<i>XYMappingGet</i>	526
<i>XYMappingSet</i>	527
<i>XYScanNextGet</i>	528
<i>XYScanNextSet</i>	529
<i>XYScanMoveAbsolute</i>	530
<i>XYScanMoveRelative</i>	531
<i>XYScanExecutePVT</i>	532
<i>XYScanShutterDisable</i>	533
<i>XYScanShutterEnable</i>	534
<i>XYShutterInterlockMonitoringDisable</i>	535
<i>XYShutterInterlockMonitoringEnable</i>	536
<i>XYShutterInterlockParametersGet</i>	537
<i>XYShutterInterlockParametersSet</i>	539
<i>XYZGroupPositionCorrectedProfilerGet</i>	541
<i>XYZGroupPositionPCORawEncoderGet</i>	542
<i>HexapodPositionCurrentGet</i>	543
<i>HexapodPositionSetpointGet</i>	544

<i>HexapodPositionTargetGet</i>	545
<i>HexapodMoveAbsolute</i>	546
<i>HexapodMoveIncremental</i>	547
<i>HexapodCoordinatesGet</i>	548
<i>HexapodCoordinateSystemSet</i>	550
<i>HexapodCoordinateSystemGet</i>	551
<i>HexapodMoveIncrementalControl</i>	552
<i>HexapodMoveIncrementalControlWithTargetVelocity</i>	553
<i>HexapodMoveIncrementalControlPulseAndGatheringSet</i>	554
<i>HexapodMoveIncrementalControlLimitGet</i>	555
<i>HexapodSGammaParametersDistanceGet</i>	556
<i>MaskClampDisable</i>	557
<i>MaskClampEnable</i>	558
<i>MaskClampStateGet</i>	559
<i>SingleAxisThetaClampDisable</i>	560
<i>SingleAxisThetaClampEnable</i>	561
<i>SingleAxisThetaFeedforwardParametersGet</i>	562
<i>SingleAxisThetaFeedforwardParametersSet</i>	563
<i>SingleAxisThetaFeedforwardJerkParametersGet</i>	564
<i>SingleAxisThetaFeedforwardJerkParametersSet</i>	565
<i>SingleAxisThetaSlaveModeEnable</i>	566
<i>SingleAxisThetaSlaveModeDisable</i>	567
<i>SingleAxisThetaSlaveParametersGet</i>	568
<i>SingleAxisThetaSlaveParametersSet</i>	569
<i>TZMotorDecouplingMatrixGet</i>	570
<i>TZMotorDecouplingMatrixSet</i>	571
<i>TZMotorDecouplingModeGet</i>	572
<i>TZMotorDecouplingModeSet</i>	573
<i>TZEncoderCouplingMatrixGet</i>	574
<i>TZEncoderCouplingMatrixSet</i>	575
<i>TZEncoderCouplingModeGet</i>	576
<i>TZEncoderCouplingModeSet</i>	577

<i>TZMappingModeGet</i>	578
<i>TZMappingModeSet</i>	579
<i>TZTrackingCutOffFrequencyGet</i>	580
<i>TZTrackingCutOffFrequencySet</i>	581
<i>TZFocusModeEnable</i>	582
<i>TZFocusModeDisable</i>	583
<i>TZTrackingUserMaximumZZZTargetDifferenceGet</i>	584
<i>TZTrackingUserMaximumZZZTargetDifferenceSet</i>	585
<i>ExternalModuleSocketReserve</i>	586
<i>ExternalModuleSocketFree</i>	587
<i>ExternalModuleFirmwareVersionGet</i>	588
<i>ExternalModuleErrorStringGet</i>	589
<i>ExternalModuleISRTimeDurationsGet</i>	590
<i>LoginCheck</i>	591
<i>LoginSCheck</i>	592
<i>GroupAllPositionTrace</i>	593
<i>GroupMotorMatrixTrace</i>	594
<i>GroupMotorMatrixInverseTrace</i>	595
<i>GroupPositionCurrentRawGet</i>	596
<i>PositionerMotorOutputOffsetGet</i>	597
<i>PositionerMotorOutputOffsetSet</i>	598
<i>PositionerMotorReferencePositionGet</i>	599
<i>PositionerEncoderAbsoluteHardConfigurationGet</i>	600
<i>SingleAxisThetaPositionRawGet</i>	601
<i>SingleAxisThetaPositionRawCorrectedGet</i>	602
<i>EEPROMCIESet</i>	603
<i>EEPROMDACOffsetCIESet</i>	604
<i>EEPROMDriverSet</i>	605
<i>EEPROMINTSet</i>	606
<i>CPUCoreAndBoardSupplyVoltagesGet</i>	607
<i>CPUTemperatureAndFanSpeedGet</i>	608
<i>CIEHeaderGet</i>	609

<i>CIEReset</i>	610
<i>ActionListGet</i>	611
<i>ActionExtendedListGet</i>	612
<i>APIExtendedListGet</i>	613
<i>APIListGet</i>	614
<i>APIListStandardGet</i>	615
<i>APIListAMATGet</i>	616
<i>TCLScriptsListGet</i>	617
<i>TrajectoryFilesListGet</i>	618
<i>ControllerStatusListGet</i>	619
<i>ErrorListGet</i>	620
<i>EventListGet</i>	621
<i>GatheringListGet</i>	622
<i>GatheringExtendedListGet</i>	623
<i>GatheringExternalListGet</i>	624
<i>GroupStatusListGet</i>	625
<i>HardwareInternalListGet</i>	626
<i>ObjectsListGet</i>	627
<i>PositionerErrorListGet</i>	628
<i>PositionerHardwareStatusListGet</i>	629
<i>PositionerDriverStatusListGet</i>	630
<i>ReferencingActionListGet</i>	631
<i>ReferencingSensorListGet</i>	632
<i>SystemIniParameterGet</i>	633
<i>SystemIniParameterSet</i>	634
<i>SystemRefParameterGet</i>	635
<i>SystemRefParameterSet</i>	636
<i>FirmwareRefParameterGet</i>	637
<i>GatheringUserDatasGet</i>	638
<i>GatheringUserDatasReset</i>	640
<i>SocketsStatusGet</i>	641
<i>TestTCP</i>	642

<i>OptionalModuleExecute</i>	643
<i>OptionalModuleKill</i>	644
<i>CIERegister32ValueGet</i>	645
<i>CIERegister64ValueGet</i>	646
<i>CIERegisterValueSet</i>	647
<i>DRV11StatusGet</i>	648
<i>DebugTraceCommunicationReset</i>	649
<i>DebugTraceCommunicationSave</i>	650
<i>RunTraceloggerProcessWithTimeSetting</i>	651
<i>RunTraceloggerProcessWithRollingBuffer</i>	652
<i>CreateQNXEvent</i>	653
<i>StartEventsAcquisition</i>	654
<i>StopEventsAcquisition</i>	655
<i>EventTriggerSet</i>	656
<i>DebugISRReset</i>	657
<i>DebugISRCorrectorLoadRatioGet</i>	658
<i>DebugISRProfilerLoadRatioGet</i>	659
<i>DebugISRServitudesLoadRatioGet</i>	660
<i>DebugISRCorrectorUsageGet</i>	661
<i>DebugISRProfilerUsageGet</i>	662
<i>DebugISRServitudesUsageGet</i>	663
<i>DebugCorrectorTimeUsageGet</i>	664
<i>DebugProfilerTimeUsageGet</i>	666
<i>DebugServitudesTimeUsageGet</i>	667
<i>ControllerMotionKernelMinMaxTimeLoadGet</i>	668
<i>ControllerMotionKernelMinMaxTimeLoadReset</i>	670
<i>ControllerMotionKernelPeriodMinMaxGet</i>	671
<i>ControllerMotionKernelPeriodMinMaxReset</i>	672
<i>ISRCorrectorCompensateOvverrunNumberGet</i>	673
<i>ISRCorrectorCompensateOvverrunNumberReset</i>	674
<i>RunQconn</i>	675
<i>RelockFiles</i>	676

<i>UnlockFiles</i>	677
<i>Crash</i>	678
<i>RunPidin</i>	679

DII .NET function return

Function error code: 0 in success and -1 on failure

- 0 : Success
- -1 : Function failed => check **errstring** parameter to get description

"errstring" parameter

This parameter "errstring" is a string that contains the error description from system or XPS.

System error description:

- -1, Invalid Socket ID.
- -2, System exception (Communication error / timeout). *[Exception description]*
- -10000, No response received from the instrument.

XPS error code: (refer to XPS Unified Programmer's manual)

XPS Error description get from "ErrorStringGet" function	
• 2	<i>Error to ignore</i>
• 1	<i>TCL interpretor error: wrong syntax</i>
• 0	<i>Successful command</i>
• -1	<i>Busy socket: previous command not yet finished</i>
• -2	<i>TCP timeout</i>
• -3	<i>String command too long</i>
• -4	<i>Unknown command</i>
• ...	<i>...</i>

IsDeviceConnected

Syntax

C# prototype

```
bool IsDeviceConnected ()
```

Python prototype

```
[state] IsDeviceConnected ()
```

Parameters

Input parameters

None

Output parameters

None

Return

(int) state: true if connected and false if not connected

Description

This function is used to know if the device is connected or not.

OpenInstrument

Syntax

C# prototype

int OpenInstrument (string address, int port, int timeout)

Python prototype

[error] OpenInstrument (address, port, timeout)

Parameters

Input parameters

(string) address: IP address of instrument.

(int) port: Port number.

(int) timeout: ReceiveTimeout in milliseconds

Output parameters

None

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to create and open a socket. Send Timeout is set to 1 second.

CloseInstrument

Syntax

C# prototype

```
int CloseInstrument ( )
```

Python prototype

```
[error] CloseInstrument ( )
```

Parameters

Input parameters

None

Output parameters

None

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to close the current socket.

SetTimeout

Syntax

C# prototype

int SetTimeout (int SendingTimeout, int ReadingTimeout)

Python prototype

[error] SetTimeout (SendingTimeout, ReadingTimeout)

Parameters

Input parameters

(int) SendingTimeout: Sending timeout in milliseconds.

(int) ReadingTimeout: Reading timeout in milliseconds.

Output parameters

None

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to configure socket timeout for sending and reading.

WriteToInstrument

Syntax

C# prototype

int WriteToInstrument (string command, out string response, out string errstring)

Python prototype

[error, response, errstring] WriteToInstrument (command)

Parameters

Input parameters

(string) command: Command string to send

Output parameters

(string) response: Response string returned by the device

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to send a command and read a rseponse.

GetVerCommand

Syntax

C# prototype

```
int GetVerCommand(out string Version, out string errstring)
```

Python prototype

```
[Version, errstring] GetVerCommand ()
```

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetVerCommand command which is used to Return firmware version. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSAAxisByAxisExecution

Syntax

C# prototype

```
int MultipleAxesPDSAAxisByAxisExecution(out Int32 IsTresholdReached, out double SignalValue, out double
SearchTime, string[] PositionerName, double[] Range, out double[] PositionerPeakLocation, Int32 nbItems, out
string errstring)
```

Python prototype

```
[IsTresholdReached, SignalValue, SearchTime, PositionerPeakLocation, errstring]
MultipleAxesPDSAAxisByAxisExecution (PositionerName, Range, nbItems)
```

Parameters

Input parameters

(string[]) PositionerName: PositionerName

(double[]) Range: Range

(Int32) nbItems: nbItems

Output parameters

(Int32_i) IsTresholdReached: IsTresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double[]) PositionerPeakLocation: PositionerPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSAAxisByAxisExecution command which is used to AxisByAxis Search algorithm. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSASpiralStepExecution

Syntax

C# prototype

```
int MultipleAxesPDSASpiralStepExecution(string FirstPositionerName, string SecondPositionerName, double Step,
double Range, out Int32 IsTresholdReached, out double SignalValue, out double SearchTime, out double
FirstPositionerPeakLocation, out double SecondPositionerPeakLocation, out string errstring)
```

Python prototype

```
[IsTresholdReached, SignalValue, SearchTime, FirstPositionerPeakLocation, SecondPositionerPeakLocation, errstring]
MultipleAxesPDSASpiralStepExecution (FirstPositionerName, SecondPositionerName, Step, Range)
```

Parameters

Input parameters

(string) FirstPositionerName: FirstPositionerName

(string) SecondPositionerName: SecondPositionerName

(double) Step: Step

(double) Range: Range

Output parameters

(Int32_i) IsTresholdReached: IsTresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double) FirstPositionerPeakLocation: FirstPositionerPeakLocation

(double) SecondPositionerPeakLocation: SecondPositionerPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSASpiralStepExecution command which is used to SpiralStep Search algorithm. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSASpiralContinuousExecution

Syntax

C# prototype

```
int MultipleAxesPDSASpiralContinuousExecution(string FirstPositionerName, string SecondPositionerName, double Step, double Range, double Speed, out Int32 IsTresholdReached, out double SignalValue, out double SearchTime, out double FirstPositionerPeakLocation, out double SecondPositionerPeakLocation, out string errstring)
```

Python prototype

```
[IsTresholdReached, SignalValue, SearchTime, FirstPositionerPeakLocation, SecondPositionerPeakLocation, errstring]  
MultipleAxesPDSASpiralContinuousExecution (FirstPositionerName, SecondPositionerName, Step, Range, Speed)
```

Parameters

Input parameters

(string) FirstPositionerName: FirstPositionerName

(string) SecondPositionerName: SecondPositionerName

(double) Step: Step

(double) Range: Range

(double) Speed: Speed

Output parameters

(Int32_i) IsTresholdReached: IsTresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double) FirstPositionerPeakLocation: FirstPositionerPeakLocation

(double) SecondPositionerPeakLocation: SecondPositionerPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSASpiralContinuousExecution command which is used to SpiralContinuous Search algorithm. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSADichotomyExecution

Syntax

C# prototype

```
int MultipleAxesPDSADichotomyExecution(double ReductionFactor, double MaxReductionFactor, out Int32
IsTresholdReached, out double SignalValue, out double SearchTime, string[] PositionerName, double[] Step, out
double[] PositionerPeakLocation, Int32 nbItems, out string errstring)
```

Python prototype

```
[IsTresholdReached, SignalValue, SearchTime, PositionerPeakLocation, errstring]
MultipleAxesPDSADichotomyExecution (ReductionFactor, MaxReductionFactor, PositionerName, Step, nbItems)
```

Parameters

Input parameters

(double) ReductionFactor: ReductionFactor

(double) MaxReductionFactor: MaxReductionFactor

(string[]) PositionerName: PositionerName

(double[]) Step: Step

(Int32) nbItems: nbItems

Output parameters

(Int32_i) IsTresholdReached: IsTresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double[]) PositionerPeakLocation: PositionerPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSADichotomyExecution command which is used to Dichotomy Search algorithm. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSARasterExecution

Syntax

C# prototype

```
int MultipleAxesPDSARasterExecution(string FirstPositionerName, string SecondPositionerName, double Length,
double Step, double Width, out Int32 IsThresholdReached, out double SignalValue, out double SearchTime, out
double FirstPositionerPeakLocation, out double SecondPositionerPeakLocation, out string errstring)
```

Python prototype

```
[IsThresholdReached, SignalValue, SearchTime, FirstPositionerPeakLocation, SecondPositionerPeakLocation, errstring]
MultipleAxesPDSARasterExecution (FirstPositionerName, SecondPositionerName, Length, Step, Width)
```

Parameters

Input parameters

(string) FirstPositionerName: FirstPositionerName

(string) SecondPositionerName: SecondPositionerName

(double) Length: Length

(double) Step: Step

(double) Width: Width

Output parameters

(Int32_i) IsThresholdReached: IsThresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double) FirstPositionerPeakLocation: FirstPositionerPeakLocation

(double) SecondPositionerPeakLocation: SecondPositionerPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSARasterExecution command which is used to Raster Search algorithm. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSAEscaladeStepExecution

Syntax

C# prototype

```
int MultipleAxesPDSAEscaladeStepExecution(string GroupName, double XYStep, double XYrange, double ZStep,
out Int32 IsThresholdReached, out double SignalValue, out double SearchTime, out double
FirstPositionerPeakLocation, out double SecondPositionerPeakLocation, out double ZPeakLocation, out string
errstring)
```

Python prototype

```
[IsThresholdReached, SignalValue, SearchTime, FirstPositionerPeakLocation, SecondPositionerPeakLocation,
ZPeakLocation, errstring] MultipleAxesPDSAEscaladeStepExecution (GroupName, XYStep, XYrange, ZStep)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) XYStep: XYStep

(double) XYrange: XYrange

(double) ZStep: ZStep

Output parameters

(Int32_i) IsThresholdReached: IsThresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double) FirstPositionerPeakLocation: FirstPositionerPeakLocation

(double) SecondPositionerPeakLocation: SecondPositionerPeakLocation

(double) ZPeakLocation: ZPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSAEscaladeStepExecution command which is used to EscaladeStep Search algorithm. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPDSAEscaladeContinuousExecution

Syntax

C# prototype

```
int MultipleAxesPDSAEscaladeContinuousExecution(string GroupName, double XYStep, double XYrange, double ZStep, double Speed, out Int32 IsThresholdReached, out double SignalValue, out double SearchTime, out double FirstPositionerPeakLocation, out double SecondPositionerPeakLocation, out double ZPeakLocation, out string errstring)
```

Python prototype

```
[IsThresholdReached, SignalValue, SearchTime, FirstPositionerPeakLocation, SecondPositionerPeakLocation, ZPeakLocation, errstring] MultipleAxesPDSAEscaladeContinuousExecution (GroupName, XYStep, XYrange, ZStep, Speed)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) XYStep: XYStep

(double) XYrange: XYrange

(double) ZStep: ZStep

(double) Speed: Speed

Output parameters

(Int32_i) IsThresholdReached: IsThresholdReached

(double) SignalValue: SignalValue

(double) SearchTime: SearchTime

(double) FirstPositionerPeakLocation: FirstPositionerPeakLocation

(double) SecondPositionerPeakLocation: SecondPositionerPeakLocation

(double) ZPeakLocation: ZPeakLocation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPDSAEscaladeContinuousExecution command which is used to EscaladeContinuous Search algorithm. Refer to the XPS Programmer's manual to get the command description.

GetVerCommand

Syntax

C# prototype

```
int GetVerCommand(out string Version, out string errstring)
```

Python prototype

```
[Version, errstring] GetVerCommand ()
```

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetVerCommand command which is used to Return firmware version. Refer to the XPS Programmer's manual to get the command description.

SetGantryMode

Syntax

C# prototype

```
int SetGantryMode(string Option, out string errstring)
```

Python prototype

```
[errstring] SetGantryMode (Option)
```

Parameters

Input parameters

(string) Option: Option

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetGantryMode command which is used to Set Gantry mode . Refer to the XPS Programmer's manual to get the command description.

GetGantryMode

Syntax

C# prototype

```
int GetGantryMode(out string Option, out string errstring)
```

Python prototype

```
[Option, errstring] GetGantryMode ()
```

Parameters

Input parameters

None

Output parameters

(string) Option: Option

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetGantryMode command which is used to Set Gantry mode . Refer to the XPS Programmer's manual to get the command description.

InitializeAndHomeXY

Syntax

C# prototype

```
int InitializeAndHomeXY(string Option, out string errstring)
```

Python prototype

```
[errstring] InitializeAndHomeXY (Option)
```

Parameters

Input parameters

(string) Option: Option

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous InitializeAndHomeXY command which is used to Do a Home search on X and after Y axis. Refer to the XPS Programmer's manual to get the command description.

InitializeAndHomeY

Syntax

C# prototype

```
int InitializeAndHomeY( out string errstring)
```

Python prototype

```
[errstring] InitializeAndHomeY ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous InitializeAndHomeY command which is used to Do a Home search on Y axis. It's always the scale. Refer to the XPS Programmer's manual to get the command description.

InitializeAndHomeX

Syntax

C# prototype

```
int InitializeAndHomeX(string Option, out string errstring)
```

Python prototype

```
[errstring] InitializeAndHomeX (Option)
```

Parameters

Input parameters

(string) Option: Option

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous InitializeAndHomeX command which is used to Do a Home search on X axis. Refer to the XPS Programmer's manual to get the command description.

MoveAbsolute

Syntax

C# prototype

```
int MoveAbsolute(double PositonAbsoluteX_um, double PositionAbsoluteY_um, out string errstring)
```

Python prototype

```
[errstring] MoveAbsolute (PositonAbsoluteX_um, PositionAbsoluteY_um)
```

Parameters

Input parameters

(double) PositonAbsoluteX_um: PositonAbsoluteX_um

(double) PositionAbsoluteY_um: PositionAbsoluteY_um

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MoveAbsolute command which is used to Moves the stage to the end position. The positions are defined in um. Refer to the XPS Programmer's manual to get the command description.

MoveSlice

Syntax

C# prototype

```
int MoveSlice(double y_end_um, double x_end_um, double scan_angle_urad, out string errstring)
```

Python prototype

```
[errstring] MoveSlice (y_end_um, x_end_um, scan_angle_urad)
```

Parameters

Input parameters

(double) y_end_um: y_end_um

(double) x_end_um: x_end_um

(double) scan_angle_urad: scan_angle_urad

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MoveSlice command which is used to Moves the stage to the end position. Begin a linear interpolation to move to . Refer to the XPS Programmer's manual to get the command description.

WaitMotionEnd

Syntax

C# prototype

```
int WaitMotionEnd(double time_out_ms, double y_end_um, double x_end_um, out string errstring)
```

Python prototype

```
[errstring] WaitMotionEnd (time_out_ms, y_end_um, x_end_um)
```

Parameters

Input parameters

(double) time_out_ms: time_out_ms

(double) y_end_um: y_end_um

(double) x_end_um: x_end_um

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous WaitMotionEnd command which is used to Wait the end of motion and checks the target positions. Refer to the XPS Programmer's manual to get the command description.

AbortMove

Syntax

C# prototype

```
int AbortMove( out string errstring)
```

Python prototype

```
[errstring] AbortMove ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous AbortMove command which is used to Abort motion. Refer to the XPS Programmer's manual to get the command description.

GetAccParams

Syntax

C# prototype

```
int GetAccParams(out Int32 x_acc_time_ms, out Int32 x_smooth_factor_ms, out Int32 y_acc_time_ms, out Int32 y_smooth_factor_ms, out string errstring)
```

Python prototype

```
[x_acc_time_ms, x_smooth_factor_ms, y_acc_time_ms, y_smooth_factor_ms, errstring] GetAccParams ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) x_acc_time_ms: x_acc_time_ms

(Int32_i) x_smooth_factor_ms: x_smooth_factor_ms

(Int32_i) y_acc_time_ms: y_acc_time_ms

(Int32_i) y_smooth_factor_ms: y_smooth_factor_ms

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetAccParams command which is used to Read acceleration parameters. Smooth factor = jerk time and all parameters unit in msec. Refer to the XPS Programmer's manual to get the command description.

SetAccParams

Syntax

C# prototype

```
int SetAccParams(Int32 x_acc_time_ms, Int32 x_smooth_factor_ms, Int32 y_acc_time_ms, Int32 y_smooth_factor_ms,  
out string errstring)
```

Python prototype

```
[errstring] SetAccParams (x_acc_time_ms, x_smooth_factor_ms, y_acc_time_ms, y_smooth_factor_ms)
```

Parameters

Input parameters

(Int32) x_acc_time_ms: x_acc_time_ms

(Int32) x_smooth_factor_ms: x_smooth_factor_ms

(Int32) y_acc_time_ms: y_acc_time_ms

(Int32) y_smooth_factor_ms: y_smooth_factor_ms

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetAccParams command which is used to Update acceleration parameters. Smooth factor = jerk time and all parameters unit in msec. Refer to the XPS Programmer's manual to get the command description.

GetVelParams

Syntax

C# prototype

```
int GetVelParams(out double vel_x, out double vel_y, out string errstring)
```

Python prototype

```
[vel_x, vel_y, errstring] GetVelParams ()
```

Parameters

Input parameters

None

Output parameters

(double) vel_x: vel_x

(double) vel_y: vel_y

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetVelParams command which is used to Read velocity parameters. All parameters unit in microns/sec. Refer to the XPS Programmer's manual to get the command description.

SetVelParams

Syntax

C# prototype

```
int SetVelParams(double vel_x, double vel_y, out string errstring)
```

Python prototype

```
[errstring] SetVelParams (vel_x, vel_y)
```

Parameters

Input parameters

(double) vel_x: vel_x

(double) vel_y: vel_y

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetVelParams command which is used to Update velocity parameters. All parameters unit in microns/sec. Refer to the XPS Programmer's manual to get the command description.

GetXVelParams

Syntax

C# prototype

```
int GetXVelParams(out double vel_x, out string errstring)
```

Python prototype

```
[vel_x, errstring] GetXVelParams ()
```

Parameters

Input parameters

None

Output parameters

(double) vel_x: vel_x

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetXVelParams command which is used to Read velocity parameter for X axis. Parameter unit in microns/sec. Refer to the XPS Programmer's manual to get the command description.

SetXVelParams

Syntax

C# prototype

```
int SetXVelParams(double vel_x, out string errstring)
```

Python prototype

```
[errstring] SetXVelParams (vel_x)
```

Parameters

Input parameters

(double) vel_x: vel_x

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetXVelParams command which is used to Update velocity parameter for X axis. Parameter unit in microns/sec. Refer to the XPS Programmer's manual to get the command description.

GetYVelParams

Syntax

C# prototype

```
int GetYVelParams(out double vel_y, out string errstring)
```

Python prototype

```
[vel_y, errstring] GetYVelParams ()
```

Parameters

Input parameters

None

Output parameters

(double) vel_y: vel_y

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetYVelParams command which is used to Read velocity parameter for Y axis. Parameter unit in microns/sec. Refer to the XPS Programmer's manual to get the command description.

SetYVelParams

Syntax

C# prototype

```
int SetYVelParams(double vel_y, out string errstring)
```

Python prototype

```
[errstring] SetYVelParams (vel_y)
```

Parameters

Input parameters

(double) vel_y: vel_y

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetYVelParams command which is used to Update velocity parameter for Y axis. Parameter unit in microns/sec. Refer to the XPS Programmer's manual to get the command description.

SetJogVelocity

Syntax

C# prototype

```
int SetJogVelocity(double vx, double vy, Int32 joystickAck_Timeout_ms, out string errstring)
```

Python prototype

```
[errstring] SetJogVelocity (vx, vy, joystickAck_Timeout_ms)
```

Parameters

Input parameters

(double) vx: vx

(double) vy: vy

(Int32) joystickAck_Timeout_ms: joystickAck_Timeout_ms

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetJogVelocity command which is used to Modify Jog velocity. Refer to the XPS Programmer's manual to get the command description.

GetJogVelocity

Syntax

C# prototype

```
int GetJogVelocity(out double vx, out double vy, out Int32 joystickAck_Timeout_ms, out string errstring)
```

Python prototype

```
[vx, vy, joystickAck_Timeout_ms, errstring] GetJogVelocity ()
```

Parameters

Input parameters

None

Output parameters

(double) vx: vx

(double) vy: vy

(Int32_i) joystickAck_Timeout_ms: joystickAck_Timeout_ms

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetJogVelocity command which is used to Get Jog velocity. Refer to the XPS Programmer's manual to get the command description.

SetJogAcceleration

Syntax

C# prototype

```
int SetJogAcceleration(Int32 x_acceleration_Time_ms, Int32 x_smooth_factor_ms, Int32 y_acceleration_Time_ms,
Int32 y_smooth_factor_ms, out string errstring)
```

Python prototype

```
[errstring] SetJogAcceleration (x_acceleration_Time_ms, x_smooth_factor_ms, y_acceleration_Time_ms,
y_smooth_factor_ms)
```

Parameters

Input parameters

(Int32) x_acceleration_Time_ms: x_acceleration_Time_ms

(Int32) x_smooth_factor_ms: x_smooth_factor_ms

(Int32) y_acceleration_Time_ms: y_acceleration_Time_ms

(Int32) y_smooth_factor_ms: y_smooth_factor_ms

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetJogAcceleration command which is used to Modify Jog acceleration. Refer to the XPS Programmer's manual to get the command description.

GetJogAcceleration

Syntax

C# prototype

```
int GetJogAcceleration(out Int32 x_acceleration_Time_ms, out Int32 x_smooth_factor_ms, out Int32
y_acceleration_Time_ms, out Int32 y_smooth_factor_ms, out string errstring)
```

Python prototype

```
[x_acceleration_Time_ms, x_smooth_factor_ms, y_acceleration_Time_ms, y_smooth_factor_ms, errstring]
GetJogAcceleration ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) x_acceleration_Time_ms: x_acceleration_Time_ms

(Int32_i) x_smooth_factor_ms: x_smooth_factor_ms

(Int32_i) y_acceleration_Time_ms: y_acceleration_Time_ms

(Int32_i) y_smooth_factor_ms: y_smooth_factor_ms

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetJogAcceleration command which is used to Get Jog acceleration. Refer to the XPS Programmer's manual to get the command description.

StartJog

Syntax

C# prototype

```
int StartJog( out string errstring)
```

Python prototype

```
[errstring] StartJog ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous StartJog command which is used to Enable Jog mode. Refer to the XPS Programmer's manual to get the command description.

EndJog

Syntax

C# prototype

```
int EndJog( out string errstring)
```

Python prototype

```
[errstring] EndJog ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EndJog command which is used to Disable Jog mode. Refer to the XPS Programmer's manual to get the command description.

GetCurrentPosition

Syntax

C# prototype

```
int GetCurrentPosition(out double y_position_um, out double x_position_um, out Int32 x1_position_cnts, out Int32 x2_position_cnts, out Int32 y_laser_position_cnts, out Int32 x_laser_position_cnts, out string errstring)
```

Python prototype

```
[y_position_um, x_position_um, x1_position_cnts, x2_position_cnts, y_laser_position_cnts, x_laser_position_cnts, errstring] GetCurrentPosition ()
```

Parameters

Input parameters

None

Output parameters

(double) y_position_um: y_position_um

(double) x_position_um: x_position_um

(Int32_i) x1_position_cnts: x1_position_cnts

(Int32_i) x2_position_cnts: x2_position_cnts

(Int32_i) y_laser_position_cnts: y_laser_position_cnts

(Int32_i) x_laser_position_cnts: x_laser_position_cnts

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetCurrentPosition command which is used to Read current positions. Refer to the XPS Programmer's manual to get the command description.

SetZone

Syntax

C# prototype

```
int SetZone(double x_center_um, double y_center_um, double radius_um, double hysteresis_um, out string  
errstring)
```

Python prototype

```
[errstring] SetZone (x_center_um, y_center_um, radius_um, hysteresis_um)
```

Parameters

Input parameters

(double) x_center_um: x_center_um

(double) y_center_um: y_center_um

(double) radius_um: radius_um

(double) hysteresis_um: hysteresis_um

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetZone command which is used to Set Zone - output signal when inside a circle. Refer to the XPS Programmer's manual to get the command description.

GetZone

Syntax

C# prototype

```
int GetZone(out double x_center_um, out double y_center_um, out double radius_um, out double hysteresis_um,  
out string errstring)
```

Python prototype

```
[x_center_um, y_center_um, radius_um, hysteresis_um, errstring] GetZone ()
```

Parameters

Input parameters

None

Output parameters

(double) x_center_um: x_center_um

(double) y_center_um: y_center_um

(double) radius_um: radius_um

(double) hysteresis_um: hysteresis_um

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetZone command which is used to Get Zone parameters. Refer to the XPS Programmer's manual to get the command description.

SetVarX

Syntax

C# prototype

int SetVarX(string NameParameter, double value, out string errstring)

Python prototype

[errstring] SetVarX (NameParameter, value)

Parameters

Input parameters

(string) NameParameter: NameParameter

(double) value: value

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetVarX command which is used to Set var parameter for X. Refer to the XPS Programmer's manual to get the command description.

SetVarXSecondary

Syntax

C# prototype

int SetVarXSecondary(string NameParameter, double value, out string errstring)

Python prototype

[errstring] SetVarXSecondary (NameParameter, value)

Parameters

Input parameters

(string) NameParameter: NameParameter

(double) value: value

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetVarXSecondary command which is used to Set var parameter X secondary. Refer to the XPS Programmer's manual to get the command description.

SetVarY

Syntax

C# prototype

int SetVarY(string NameParameter, double value, out string errstring)

Python prototype

[errstring] SetVarY (NameParameter, value)

Parameters

Input parameters

(string) NameParameter: NameParameter

(double) value: value

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetVarY command which is used to Set var parameter for Y. Refer to the XPS Programmer's manual to get the command description.

GetVarX

Syntax

C# prototype

int GetVarX(string NameParameter, out double value, out string errstring)

Python prototype

[value, errstring] GetVarX (NameParameter)

Parameters

Input parameters

(string) NameParameter: NameParameter

Output parameters

(double) value: value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetVarX command which is used to Get var parameter for X. Refer to the XPS Programmer's manual to get the command description.

GetVarXSecondary

Syntax

C# prototype

```
int GetVarXSecondary(string NameParameter, out double value, out string errstring)
```

Python prototype

```
[value, errstring] GetVarXSecondary (NameParameter)
```

Parameters

Input parameters

(string) NameParameter: NameParameter

Output parameters

(double) value: value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetVarXSecondary command which is used to Get var parameter X secondary. Refer to the XPS Programmer's manual to get the command description.

GetVarY

Syntax

C# prototype

```
int GetVarY(string NameParameter, out double value, out string errstring)
```

Python prototype

```
[value, errstring] GetVarY (NameParameter)
```

Parameters

Input parameters

(string) NameParameter: NameParameter

Output parameters

(double) value: value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetVarY command which is used to Get var parameter for Y. Refer to the XPS Programmer's manual to get the command description.

DataCollectionBufferReset

Syntax

C# prototype

```
int DataCollectionBufferReset( out string errstring)
```

Python prototype

```
[errstring] DataCollectionBufferReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DataCollectionBufferReset command which is used to Reset data collection buffer. Refer to the XPS Programmer's manual to get the command description.

DataCollectionBufferAndTimeReset

Syntax

C# prototype

```
int DataCollectionBufferAndTimeReset( out string errstring)
```

Python prototype

```
[errstring] DataCollectionBufferAndTimeReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DataCollectionBufferAndTimeReset command which is used to Reset data collection buffer and time stamp. Refer to the XPS Programmer's manual to get the command description.

DataCollectionRequest

Syntax

C# prototype

```
int DataCollectionRequest(Int32 NbRequestBlocks, out Int32 NbReturnBlocks, out string Frame, out string errstring)
```

Python prototype

```
[NbReturnBlocks, Frame, errstring] DataCollectionRequest (NbRequestBlocks)
```

Parameters

Input parameters

(Int32) NbRequestBlocks: NbRequestBlocks

Output parameters

(Int32_i) NbReturnBlocks: NbReturnBlocks

(string) Frame: Frame

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DataCollectionRequest command which is used to Get data collection . Refer to the XPS Programmer's manual to get the command description.

DataCollectionTimeStampGet

Syntax

C# prototype

```
int DataCollectionTimeStampGet(out double TimeStamp, out string errstring)
```

Python prototype

```
[TimeStamp, errstring] DataCollectionTimeStampGet ()
```

Parameters

Input parameters

None

Output parameters

(double) TimeStamp: TimeStamp

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DataCollectionTimeStampGet command which is used to Get data collection time stamp. Refer to the XPS Programmer's manual to get the command description.

DataCollectionTimeStampReset

Syntax

C# prototype

```
int DataCollectionTimeStampReset( out string errstring)
```

Python prototype

```
[errstring] DataCollectionTimeStampReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DataCollectionTimeStampReset command which is used to Reset data collection time stamp. Refer to the XPS Programmer's manual to get the command description.

SetPiston

Syntax

C# prototype

```
int SetPiston(Int32 Command, out string errstring)
```

Python prototype

```
[errstring] SetPiston (Command)
```

Parameters

Input parameters

(Int32) Command: Command

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetPiston command which is used to Set Piston command . Refer to the XPS Programmer's manual to get the command description.

GetPistonState

Syntax

C# prototype

```
int GetPistonState(out Int32 CommandState, out Int32 isEngaged, out Int32 isReleased, out Int32 LiftPinUPInterlock, out string errstring)
```

Python prototype

```
[CommandState, isEngaged, isReleased, LiftPinUPInterlock, errstring] GetPistonState ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) CommandState: CommandState

(Int32_i) isEngaged: isEngaged

(Int32_i) isReleased: isReleased

(Int32_i) LiftPinUPInterlock: LiftPinUPInterlock

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetPistonState command which is used to Get Piston State. Refer to the XPS Programmer's manual to get the command description.

SetBrake

Syntax

C# prototype

int SetBrake(Int32 Command, out string errstring)

Python prototype

[errstring] SetBrake (Command)

Parameters

Input parameters

(Int32) Command: Command

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SetBrake command which is used to Set Brake command . Refer to the XPS Programmer's manual to get the command description.

GetBrakeState

Syntax

C# prototype

```
int GetBrakeState(out Int32 CommandState, out string errstring)
```

Python prototype

```
[CommandState, errstring] GetBrakeState ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) CommandState: CommandState

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GetBrakeState command which is used to Get Brake State . Refer to the XPS Programmer's manual to get the command description.

ZygoGetVerInterfero

Syntax

C# prototype

int ZygoGetVerInterfero(out string Version, out string errstring)

Python prototype

[Version, errstring] ZygoGetVerInterfero ()

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoGetVerInterfero command which is used to Return Interferometer firmware version. Refer to the XPS Programmer's manual to get the command description.

ZygoAmplitudeGet

Syntax

C# prototype

```
int ZygoAmplitudeGet(out Int32 ZygoReferenceSignalStatus, out Int32 Meas1Signal, out Int32 Meas2Signal, out string errstring)
```

Python prototype

```
[ZygoReferenceSignalStatus, Meas1Signal, Meas2Signal, errstring] ZygoAmplitudeGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) ZygoReferenceSignalStatus: ZygoReferenceSignalStatus

(Int32_i) Meas1Signal: Meas1Signal

(Int32_i) Meas2Signal: Meas2Signal

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoAmplitudeGet command which is used to Get the reference channel status and measuring channel amplitude. Refer to the XPS Programmer's manual to get the command description.

ZygoADCDiagnosticStatusGet

Syntax

C# prototype

int ZygoADCDiagnosticStatusGet(Int32 axis, Int32 ADCMuxNumber, out string ADCDiagStatus, out string errstring)

Python prototype

[ADCDiagStatus, errstring] ZygoADCDiagnosticStatusGet (axis, ADCMuxNumber)

Parameters

Input parameters

(Int32) axis: axis

(Int32) ADCMuxNumber: ADCMuxNumber

Output parameters

(string) ADCDiagStatus: ADCDiagStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoADCDiagnosticStatusGet command which is used to Get the diagnostic ADC status. Refer to the XPS Programmer's manual to get the command description.

ZygoPositionGet

Syntax

C# prototype

```
int ZygoPositionGet(out Int64 PositionY, out Int64 PositionX, out string errstring)
```

Python prototype

```
[PositionY, PositionX, errstring] ZygoPositionGet ()
```

Parameters

Input parameters

None

Output parameters

(Int64) PositionY: PositionY

(Int64) PositionX: PositionX

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoPositionGet command which is used to Return Zygo Interferometer positions. Refer to the XPS Programmer's manual to get the command description.

ZygoResetX

Syntax

C# prototype

```
int ZygoResetX( out string errstring)
```

Python prototype

```
[errstring] ZygoResetX ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoResetX command which is used to Reset interferometer axis relates to axis X. Refer to the XPS Programmer's manual to get the command description.

ZygoResetY

Syntax

C# prototype

```
int ZygoResetY( out string errstring)
```

Python prototype

```
[errstring] ZygoResetY ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoResetY command which is used to Reset interferometer axis relates to axis Y. Refer to the XPS Programmer's manual to get the command description.

ZygoSetOffsetX

Syntax

C# prototype

```
int ZygoSetOffsetX(Int32 offset_x, out string errstring)
```

Python prototype

```
[errstring] ZygoSetOffsetX (offset_x)
```

Parameters

Input parameters

(Int32) offset_x: offset_x

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoSetOffsetX command which is used to Write the offset register of interferometer relates to axis X. Refer to the XPS Programmer's manual to get the command description.

ZygoSetOffsetY

Syntax

C# prototype

```
int ZygoSetOffsetY(Int32 offset_y, out string errstring)
```

Python prototype

```
[errstring] ZygoSetOffsetY (offset_y)
```

Parameters

Input parameters

(Int32) offset_y: offset_y

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoSetOffsetY command which is used to Write the offset register of interferometer relates to axis Y. Refer to the XPS Programmer's manual to get the command description.

ZygoSetPEGParams

Syntax

C# prototype

```
int ZygoSetPEGParams(Int32 P1, Int32 P2, UInt32 delta1, Int32 K1, UInt32 delta2, Int32 K2, Int32 ControlWord, out
string errstring)
```

Python prototype

```
[errstring] ZygoSetPEGParams (P1, P2, delta1, K1, delta2, K2, ControlWord)
```

Parameters

Input parameters

(Int32) P1: P1

(Int32) P2: P2

(UInt32) delta1: delta1

(Int32) K1: K1

(UInt32) delta2: delta2

(Int32) K2: K2

(Int32) ControlWord: ControlWord

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoSetPEGParams command which is used to Write these parameters from Zygo interferometer. Refer to the XPS Programmer's manual to get the command description.

ZygoGetPEGLastCommunicationTime

Syntax

C# prototype

int ZygoGetPEGLastCommunicationTime(out double LastCommunicationTime, out string errstring)

Python prototype

[LastCommunicationTime, errstring] ZygoGetPEGLastCommunicationTime ()

Parameters

Input parameters

None

Output parameters

(double) LastCommunicationTime: LastCommunicationTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoGetPEGLastCommunicationTime command which is used to Get the last communication time to configure Zygo PEG. Refer to the XPS Programmer's manual to get the command description.

FirmwareVersionGet

Syntax

C# prototype

```
int FirmwareVersionGet(out string Version, out string errstring)
```

Python prototype

```
[Version, errstring] FirmwareVersionGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous FirmwareVersionGet command which is used to Return firmware version from firmware.ref. Refer to the XPS Programmer's manual to get the command description.

FirmwareBuildVersionNumberGet

Syntax

C# prototype

```
int FirmwareBuildVersionNumberGet(out string Version, out string errstring)
```

Python prototype

```
[Version, errstring] FirmwareBuildVersionNumberGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous FirmwareBuildVersionNumberGet command which is used to Return firmware build version number. Refer to the XPS Programmer's manual to get the command description.

ExternalModuleTemplateNameGet

Syntax

C# prototype

```
int ExternalModuleTemplateNameGet(out string TemplateNumber, out string errstring)
```

Python prototype

```
[TemplateNumber, errstring] ExternalModuleTemplateNameGet ()
```

Parameters

Input parameters

None

Output parameters

(string) TemplateNumber: TemplateNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalModuleTemplateNameGet command which is used to Return expected external module template number. Refer to the XPS Programmer's manual to get the command description.

InstallerVersionGet

Syntax

C# prototype

```
int InstallerVersionGet(out string Version, out string errstring)
```

Python prototype

```
[Version, errstring] InstallerVersionGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous InstallerVersionGet command which is used to Return installer pack version. Refer to the XPS Programmer's manual to get the command description.

Reboot

Syntax

C# prototype

```
int Reboot( out string errstring)
```

Python prototype

```
[errstring] Reboot ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous Reboot command which is used to Reboot the controller. Refer to the XPS Programmer's manual to get the command description.

RestartApplication

Syntax

C# prototype

```
int RestartApplication( out string errstring)
```

Python prototype

```
[errstring] RestartApplication ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous RestartApplication command which is used to Restart the Controller. Refer to the XPS Programmer's manual to get the command description.

ControllerMotionKernelTimeLoadGet

Syntax

C# prototype

```
int ControllerMotionKernelTimeLoadGet(out double CPUTotalLoadRatio, out double CPUCorrectorLoadRatio, out double CPUProfilerLoadRatio, out double CPUServitudesLoadRatio, out string errstring)
```

Python prototype

```
[CPUTotalLoadRatio, CPUCorrectorLoadRatio, CPUProfilerLoadRatio, CPUServitudesLoadRatio, errstring]  
ControllerMotionKernelTimeLoadGet ()
```

Parameters

Input parameters

None

Output parameters

(double) CPUTotalLoadRatio: CPUTotalLoadRatio

(double) CPUCorrectorLoadRatio: CPUCorrectorLoadRatio

(double) CPUProfilerLoadRatio: CPUProfilerLoadRatio

(double) CPUServitudesLoadRatio: CPUServitudesLoadRatio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerMotionKernelTimeLoadGet command which is used to Get controller motion kernel time load. Refer to the XPS Programmer's manual to get the command description.

ControllerRTTimeGet

Syntax

C# prototype

```
int ControllerRTTimeGet(out double CurrentRTPeriod, out double CurrentRTUsage, out string errstring)
```

Python prototype

```
[CurrentRTPeriod, CurrentRTUsage, errstring] ControllerRTTimeGet ()
```

Parameters

Input parameters

None

Output parameters

(double) CurrentRTPeriod: CurrentRTPeriod

(double) CurrentRTUsage: CurrentRTUsage

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerRTTimeGet command which is used to Get controller corrector period and calculation time. Refer to the XPS Programmer's manual to get the command description.

ControllerSlaveStatusGet

Syntax

C# prototype

```
int ControllerSlaveStatusGet(out Int32 SlaveControllerStatus, out string errstring)
```

Python prototype

```
[SlaveControllerStatus, errstring] ControllerSlaveStatusGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) SlaveControllerStatus: SlaveControllerStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerSlaveStatusGet command which is used to Read slave controller status. Refer to the XPS Programmer's manual to get the command description.

ControllerSlaveStatusStringGet

Syntax

C# prototype

```
int ControllerSlaveStatusStringGet(Int32 SlaveControllerStatusCode, out string SlaveControllerStatusString, out string errstring)
```

Python prototype

```
[SlaveControllerStatusString, errstring] ControllerSlaveStatusStringGet (SlaveControllerStatusCode)
```

Parameters

Input parameters

(Int32) SlaveControllerStatusCode: SlaveControllerStatusCode

Output parameters

(string) SlaveControllerStatusString: SlaveControllerStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerSlaveStatusStringGet command which is used to Return the slave controller status string. Refer to the XPS Programmer's manual to get the command description.

ControllerSynchronizeCorrectorISR

Syntax

C# prototype

```
int ControllerSynchronizeCorrectorISR(string ModeString, out string errstring)
```

Python prototype

```
[errstring] ControllerSynchronizeCorrectorISR (ModeString)
```

Parameters

Input parameters

(string) ModeString: ModeString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerSynchronizeCorrectorISR command which is used to Synchronize controller corrector ISR. Refer to the XPS Programmer's manual to get the command description.

ControllerStatusGet

Syntax

C# prototype

```
int ControllerStatusGet(out Int32 ControllerStatus, out string errstring)
```

Python prototype

```
[ControllerStatus, errstring] ControllerStatusGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) ControllerStatus: ControllerStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerStatusGet command which is used to Get controller current status and reset the status. Refer to the XPS Programmer's manual to get the command description.

ControllerStatusRead

Syntax

C# prototype

```
int ControllerStatusRead(out Int32 ControllerStatus, out string errstring)
```

Python prototype

```
[ControllerStatus, errstring] ControllerStatusRead ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) ControllerStatus: ControllerStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerStatusRead command which is used to Read controller current status. Refer to the XPS Programmer's manual to get the command description.

ControllerStatusStringGet

Syntax

C# prototype

```
int ControllerStatusStringGet(Int32 ControllerStatusCode, out string ControllerStatusString, out string errstring)
```

Python prototype

```
[ControllerStatusString, errstring] ControllerStatusStringGet (ControllerStatusCode)
```

Parameters

Input parameters

(Int32) ControllerStatusCode: ControllerStatusCode

Output parameters

(string) ControllerStatusString: ControllerStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerStatusStringGet command which is used to Return the controller status string. Refer to the XPS Programmer's manual to get the command description.

ElapsedTimeGet

Syntax

C# prototype

```
int ElapsedTimeGet(out double ElapsedTime, out string errstring)
```

Python prototype

```
[ElapsedTime, errstring] ElapsedTimeGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ElapsedTime: ElapsedTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ElapsedTimeGet command which is used to Return elapsed time from controller power on. Refer to the XPS Programmer's manual to get the command description.

ErrorStringGet

Syntax

C# prototype

```
int ErrorStringGet(Int32 ErrorCode, out string ErrorString, out string errstring)
```

Python prototype

```
[ErrorString, errstring] ErrorStringGet (ErrorCode)
```

Parameters

Input parameters

(Int32) ErrorCode: ErrorCode

Output parameters

(string) ErrorString: ErrorString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ErrorStringGet command which is used to Return the error string corresponding to the error code. Refer to the XPS Programmer's manual to get the command description.

Login

Syntax

C# prototype

```
int Login(string Name, string Password, out string errstring)
```

Python prototype

```
[errstring] Login (Name, Password)
```

Parameters

Input parameters

(string) Name: Name

(string) Password: Password

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous Login command which is used to Log in. Refer to the XPS Programmer's manual to get the command description.

LoginS

Syntax

C# prototype

```
int LoginS(string Name, string CryptedPassword, out string errstring)
```

Python prototype

```
[errstring] LoginS (Name, CryptedPassword)
```

Parameters

Input parameters

(string) Name: Name

(string) CryptedPassword: CryptedPassword

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous LoginS command which is used to Log in . Refer to the XPS Programmer's manual to get the command description.

CloseAllOtherSockets

Syntax

C# prototype

```
int CloseAllOtherSockets( out string errstring)
```

Python prototype

```
[errstring] CloseAllOtherSockets ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CloseAllOtherSockets command which is used to Close all socket beside the one used to send this command. Refer to the XPS Programmer's manual to get the command description.

HardwareDriverAndStageGet

Syntax

C# prototype

```
int HardwareDriverAndStageGet(Int32 PlugNumber, out string DriverName, out string StageName, out string errstring)
```

Python prototype

```
[DriverName, StageName, errstring] HardwareDriverAndStageGet (PlugNumber)
```

Parameters

Input parameters

(Int32) PlugNumber: PlugNumber

Output parameters

(string) DriverName: DriverName

(string) StageName: StageName

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HardwareDriverAndStageGet command which is used to Smart hardware. Refer to the XPS Programmer's manual to get the command description.

HardwareDateAndTimeGet

Syntax

C# prototype

```
int HardwareDateAndTimeGet(out string DateAndTime, out string errstring)
```

Python prototype

```
[DateAndTime, errstring] HardwareDateAndTimeGet ()
```

Parameters

Input parameters

None

Output parameters

(string) DateAndTime: DateAndTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HardwareDateAndTimeGet command which is used to Return hardware date and time. Refer to the XPS Programmer's manual to get the command description.

HardwareDateAndTimeSet

Syntax

C# prototype

```
int HardwareDateAndTimeSet(string DateAndTime, out string errstring)
```

Python prototype

```
[errstring] HardwareDateAndTimeSet (DateAndTime)
```

Parameters

Input parameters

(string) DateAndTime: DateAndTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HardwareDateAndTimeSet command which is used to Set hardware date and time. Refer to the XPS Programmer's manual to get the command description.

FileScriptHistoryRename

Syntax

C# prototype

```
int FileScriptHistoryRename(string TCLFileName, out string errstring)
```

Python prototype

```
[errstring] FileScriptHistoryRename (TCLFileName)
```

Parameters

Input parameters

(string) TCLFileName: TCLFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous FileScriptHistoryRename command which is used to Rename History.tcl. Refer to the XPS Programmer's manual to get the command description.

FileGatheringRename

Syntax

C# prototype

```
int FileGatheringRename(string FileName, out string errstring)
```

Python prototype

```
[errstring] FileGatheringRename (FileName)
```

Parameters

Input parameters

(string) FileName: FileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous FileGatheringRename command which is used to Rename Gathering.dat. Refer to the XPS Programmer's manual to get the command description.

INTServitudesStatusGet

Syntax

C# prototype

```
int INTServitudesStatusGet(out Int16 INTServitudesStatus, out string errstring)
```

Python prototype

```
[INTServitudesStatus, errstring] INTServitudesStatusGet ()
```

Parameters

Input parameters

None

Output parameters

(Int16) INTServitudesStatus: INTServitudesStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous INTServitudesStatusGet command which is used to Read INT servitudes status. Refer to the XPS Programmer's manual to get the command description.

INTServitudesCommandGet

Syntax

C# prototype

```
int INTServitudesCommandGet(out Int16 INTServitudesCommand, out string errstring)
```

Python prototype

```
[INTServitudesCommand, errstring] INTServitudesCommandGet ()
```

Parameters

Input parameters

None

Output parameters

(Int16) INTServitudesCommand: INTServitudesCommand

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous INTServitudesCommandGet command which is used to Read INT servitudes command. Refer to the XPS Programmer's manual to get the command description.

TCLScriptExecute

Syntax

C# prototype

```
int TCLScriptExecute(string TCLFileName, string TaskName, string ParametersList, out string errstring)
```

Python prototype

```
[errstring] TCLScriptExecute (TCLFileName, TaskName, ParametersList)
```

Parameters

Input parameters

(string) TCLFileName: TCLFileName

(string) TaskName: TaskName

(string) ParametersList: ParametersList

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptExecute command which is used to Execute a TCL script from a TCL file. Refer to the XPS Programmer's manual to get the command description.

TCLScriptExecuteAndWait

Syntax

C# prototype

```
int TCLScriptExecuteAndWait(string TCLFileName, string TaskName, string InputParametersList, out string
OutputParametersList, out string errstring)
```

Python prototype

```
[OutputParametersList, errstring] TCLScriptExecuteAndWait (TCLFileName, TaskName, InputParametersList)
```

Parameters

Input parameters

(string) TCLFileName: TCLFileName

(string) TaskName: TaskName

(string) InputParametersList: InputParametersList

Output parameters

(string) OutputParametersList: OutputParametersList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptExecuteAndWait command which is used to Execute a TCL script from a TCL file and wait the end of execution to return. Refer to the XPS Programmer's manual to get the command description.

TCLScriptExecuteWithPriority

Syntax

C# prototype

```
int TCLScriptExecuteWithPriority(string TCLFileName, string TaskName, string TaskPriorityLevel, string
ParametersList, out string errstring)
```

Python prototype

```
[errstring] TCLScriptExecuteWithPriority (TCLFileName, TaskName, TaskPriorityLevel, ParametersList)
```

Parameters

Input parameters

(string) TCLFileName: TCLFileName

(string) TaskName: TaskName

(string) TaskPriorityLevel: TaskPriorityLevel

(string) ParametersList: ParametersList

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptExecuteWithPriority command which is used to Execute a TCL script with defined priority. Refer to the XPS Programmer's manual to get the command description.

TCLScriptKill

Syntax

C# prototype

int TCLScriptKill(string TaskName, out string errstring)

Python prototype

[errstring] TCLScriptKill (TaskName)

Parameters

Input parameters

(string) TaskName: TaskName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptKill command which is used to Kill TCL Task. Refer to the XPS Programmer's manual to get the command description.

TCLScriptKillAll

Syntax

C# prototype

int TCLScriptKillAll(out string errstring)

Python prototype

[errstring] TCLScriptKillAll ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptKillAll command which is used to Kill all TCL Tasks. Refer to the XPS Programmer's manual to get the command description.

TCLScriptRunningListGet

Syntax

C# prototype

int TCLScriptRunningListGet(out string TCLTaskList, out string errstring)

Python prototype

[TCLTaskList, errstring] TCLScriptRunningListGet ()

Parameters

Input parameters

None

Output parameters

(string) TCLTaskList: TCLTaskList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptRunningListGet command which is used to TCL task running list . Refer to the XPS Programmer's manual to get the command description.

TimerGet

Syntax

C# prototype

```
int TimerGet(string TimerName, out Int32 FrequencyTicks, out string errstring)
```

Python prototype

```
[FrequencyTicks, errstring] TimerGet (TimerName)
```

Parameters

Input parameters

(string) TimerName: TimerName

Output parameters

(Int32_i) FrequencyTicks: FrequencyTicks

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TimerGet command which is used to Get a timer. Refer to the XPS Programmer's manual to get the command description.

TimerSet

Syntax

C# prototype

```
int TimerSet(string TimerName, Int32 FrequencyTicks, out string errstring)
```

Python prototype

```
[errstring] TimerSet (TimerName, FrequencyTicks)
```

Parameters

Input parameters

(string) TimerName: TimerName

(Int32) FrequencyTicks: FrequencyTicks

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TimerSet command which is used to Set a timer. Refer to the XPS Programmer's manual to get the command description.

CleanTmpFolder

Syntax

C# prototype

```
int CleanTmpFolder( out string errstring)
```

Python prototype

```
[errstring] CleanTmpFolder ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CleanTmpFolder command which is used to Clean Tmp folder. Refer to the XPS Programmer's manual to get the command description.

CleanCoreDumpFolder

Syntax

C# prototype

```
int CleanCoreDumpFolder( out string errstring)
```

Python prototype

```
[errstring] CleanCoreDumpFolder ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CleanCoreDumpFolder command which is used to Remove core files existing in /Admin/Public/CoreDump folder. Refer to the XPS Programmer's manual to get the command description.

GlobalArrayGet

Syntax

C# prototype

```
int GlobalArrayGet(Int32 Number, out string ValueString, out string errstring)
```

Python prototype

```
[ValueString, errstring] GlobalArrayGet (Number)
```

Parameters

Input parameters

(Int32) Number: Number

Output parameters

(string) ValueString: ValueString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GlobalArrayGet command which is used to Get global array value. Refer to the XPS Programmer's manual to get the command description.

GlobalArraySet

Syntax

C# prototype

```
int GlobalArraySet(Int32 Number, string ValueString, out string errstring)
```

Python prototype

```
[errstring] GlobalArraySet (Number, ValueString)
```

Parameters

Input parameters

(Int32) Number: Number

(string) ValueString: ValueString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GlobalArraySet command which is used to Set global array value. Refer to the XPS Programmer's manual to get the command description.

DoubleGlobalArrayGet

Syntax

C# prototype

```
int DoubleGlobalArrayGet(Int32 Number, out double DoubleValue, out string errstring)
```

Python prototype

```
[DoubleValue, errstring] DoubleGlobalArrayGet (Number)
```

Parameters

Input parameters

(Int32) Number: Number

Output parameters

(double) DoubleValue: DoubleValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DoubleGlobalArrayGet command which is used to Get double global array value. Refer to the XPS Programmer's manual to get the command description.

DoubleGlobalArraySet

Syntax

C# prototype

```
int DoubleGlobalArraySet(Int32 Number, double DoubleValue, out string errstring)
```

Python prototype

```
[errstring] DoubleGlobalArraySet (Number, DoubleValue)
```

Parameters

Input parameters

(Int32) Number: Number

(double) DoubleValue: DoubleValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DoubleGlobalArraySet command which is used to Set double global array value. Refer to the XPS Programmer's manual to get the command description.

PositionerMagneticTrackPositionAtHomeGet

Syntax

C# prototype

```
int PositionerMagneticTrackPositionAtHomeGet(string PositionerName, out double MagneticTrackPosition, out string errstring)
```

Python prototype

```
[MagneticTrackPosition, errstring] PositionerMagneticTrackPositionAtHomeGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MagneticTrackPosition: MagneticTrackPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMagneticTrackPositionAtHomeGet command which is used to Get magnetic track position at home in units. Refer to the XPS Programmer's manual to get the command description.

PositionerMagneticTrackPositionAtHomeSet

Syntax

C# prototype

```
int PositionerMagneticTrackPositionAtHomeSet(string PositionerName, double MagneticTrackPosition, out string errstring)
```

Python prototype

```
[errstring] PositionerMagneticTrackPositionAtHomeSet (PositionerName, MagneticTrackPosition)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) MagneticTrackPosition: MagneticTrackPosition

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMagneticTrackPositionAtHomeSet command which is used to Set magnetic track position at home in units. Refer to the XPS Programmer's manual to get the command description.

PositionerMotorDualSinForceBalanceGet

Syntax

C# prototype

```
int PositionerMotorDualSinForceBalanceGet(string PositionerName, out double FirstMotorForceBalance, out double SecondMotorForceBalance, out string errstring)
```

Python prototype

```
[FirstMotorForceBalance, SecondMotorForceBalance, errstring] PositionerMotorDualSinForceBalanceGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) FirstMotorForceBalance: FirstMotorForceBalance

(double) SecondMotorForceBalance: SecondMotorForceBalance

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotorDualSinForceBalanceGet command which is used to Get dual sin motor force balance parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerMotorDualSinForceBalanceSet

Syntax

C# prototype

```
int PositionerMotorDualSinForceBalanceSet(string PositionerName, double FirstMotorForceBalance, double SecondMotorForceBalance, out string errstring)
```

Python prototype

```
[errstring] PositionerMotorDualSinForceBalanceSet (PositionerName, FirstMotorForceBalance, SecondMotorForceBalance)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) FirstMotorForceBalance: FirstMotorForceBalance

(double) SecondMotorForceBalance: SecondMotorForceBalance

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotorDualSinForceBalanceSet command which is used to Set dual sin motor force balance parameters. Refer to the XPS Programmer's manual to get the command description.

ZygoConnectToServer

Syntax

C# prototype

```
int ZygoConnectToServer( out string errstring)
```

Python prototype

```
[errstring] ZygoConnectToServer ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoConnectToServer command which is used to Connect to Zygo TCP server. Refer to the XPS Programmer's manual to get the command description.

ZygoReset

Syntax

C# prototype

```
int ZygoReset( out string errstring)
```

Python prototype

```
[errstring] ZygoReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoReset command which is used to Reset all Zygo axes. Refer to the XPS Programmer's manual to get the command description.

ZygoReadWord

Syntax

C# prototype

```
int ZygoReadWord(string AxisNum, string Register, out string Response, out string errstring)
```

Python prototype

```
[Response, errstring] ZygoReadWord (AxisNum, Register)
```

Parameters

Input parameters

(string) AxisNum: AxisNum

(string) Register: Register

Output parameters

(string) Response: Response

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoReadWord command which is used to Read word size from Zygo register, enter AXIS1|2|3|4 and register offset. Refer to the XPS Programmer's manual to get the command description.

ZygoReadLong

Syntax

C# prototype

```
int ZygoReadLong(string AxisNum, string Register, out string Response, out string errstring)
```

Python prototype

```
[Response, errstring] ZygoReadLong (AxisNum, Register)
```

Parameters

Input parameters

(string) AxisNum: AxisNum

(string) Register: Register

Output parameters

(string) Response: Response

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoReadLong command which is used to Read long size from Zygo register, enter AXIS1|2|3|4 and register offset. Refer to the XPS Programmer's manual to get the command description.

ZygoWriteWord

Syntax

C# prototype

```
int ZygoWriteWord(string AxisNum, string Register, string Data, out string Response, out string errstring)
```

Python prototype

```
[Response, errstring] ZygoWriteWord (AxisNum, Register, Data)
```

Parameters

Input parameters

(string) AxisNum: AxisNum

(string) Register: Register

(string) Data: Data

Output parameters

(string) Response: Response

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoWriteWord command which is used to Read word size from Zygo register, enter AXIS1|2|3|4, register offset and data value. Refer to the XPS Programmer's manual to get the command description.

ZygoWriteLong

Syntax

C# prototype

```
int ZygoWriteLong(string AxisNum, string Register, string Data, out string Response, out string errstring)
```

Python prototype

```
[Response, errstring] ZygoWriteLong (AxisNum, Register, Data)
```

Parameters

Input parameters

(string) AxisNum: AxisNum

(string) Register: Register

(string) Data: Data

Output parameters

(string) Response: Response

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoWriteLong command which is used to Read long size from Zygo register,enter AXIS1|2|3|4, register offset and data value. Refer to the XPS Programmer's manual to get the command description.

ZygoSendAndReceive

Syntax

C# prototype

int ZygoSendAndReceive(string Command, out string Response, out string errstring)

Python prototype

[Response, errstring] ZygoSendAndReceive (Command)

Parameters

Input parameters

(string) Command: Command

Output parameters

(string) Response: Response

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoSendAndReceive command which is used to Send command to Zygo board and receive response. Refer to the XPS Programmer's manual to get the command description.

ZygoDisconnectFromServer

Syntax

C# prototype

```
int ZygoDisconnectFromServer( out string errstring)
```

Python prototype

```
[errstring] ZygoDisconnectFromServer ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoDisconnectFromServer command which is used to Disconnect from Zygo server. Refer to the XPS Programmer's manual to get the command description.

ZygoEthernetCommunicationStatusGet

Syntax

C# prototype

int ZygoEthernetCommunicationStatusGet(out Int32 EthernetCommunicationStatus, out string errstring)

Python prototype

[EthernetCommunicationStatus, errstring] ZygoEthernetCommunicationStatusGet ()

Parameters

Input parameters

None

Output parameters

(Int32_i) EthernetCommunicationStatus: EthernetCommunicationStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoEthernetCommunicationStatusGet command which is used to Get the Ethernet Communication Status . Refer to the XPS Programmer's manual to get the command description.

ZygoInterferometerStatusGet

Syntax

C# prototype

```
int ZygoInterferometerStatusGet(out Int32 EthernetCommunicationStatus, out Int32 ZygoAxis1MeasureSignal, out
Int32 ZygoAxis2MeasureSignal, out Int32 ZygoReferenceSignalStatus, out Int32 ZygoP2BoardStatus, out string
errstring)
```

Python prototype

```
[EthernetCommunicationStatus, ZygoAxis1MeasureSignal, ZygoAxis2MeasureSignal, ZygoReferenceSignalStatus,
ZygoP2BoardStatus, errstring] ZygoInterferometerStatusGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) EthernetCommunicationStatus: EthernetCommunicationStatus

(Int32_i) ZygoAxis1MeasureSignal: ZygoAxis1MeasureSignal

(Int32_i) ZygoAxis2MeasureSignal: ZygoAxis2MeasureSignal

(Int32_i) ZygoReferenceSignalStatus: ZygoReferenceSignalStatus

(Int32_i) ZygoP2BoardStatus: ZygoP2BoardStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoInterferometerStatusGet command which is used to Get the ZYGO interferometer status. Refer to the XPS Programmer's manual to get the command description.

ZygoStartInterferometer

Syntax

C# prototype

```
int ZygoStartInterferometer( out string errstring)
```

Python prototype

```
[errstring] ZygoStartInterferometer ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoStartInterferometer command which is used to Start ZYGO interferometer. Can take several minutes.. Refer to the XPS Programmer's manual to get the command description.

ZygoStartBoardP2

Syntax

C# prototype

```
int ZygoStartBoardP2(Int32 Kv, Int32 Kp, bool ReverseDirectionSenseChannel1, bool  
ReverseDirectionSenseChannel2, Int32 DataAgeAdjust, out string errstring)
```

Python prototype

```
[errstring] ZygoStartBoardP2 (Kv, Kp, ReverseDirectionSenseChannel1, ReverseDirectionSenseChannel2,  
DataAgeAdjust)
```

Parameters

Input parameters

(Int32) Kv: Kv

(Int32) Kp: Kp

(bool) ReverseDirectionSenseChannel1: ReverseDirectionSenseChannel1

(bool) ReverseDirectionSenseChannel2: ReverseDirectionSenseChannel2

(Int32) DataAgeAdjust: DataAgeAdjust

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoStartBoardP2 command which is used to Get the Zygo Board P2 up. Refer to the XPS Programmer's manual to get the command description.

ZygoErrorStatusStringGet

Syntax

C# prototype

int ZygoErrorStatusStringGet(Int32 axis, out string ErrorStatus, out string errstring)

Python prototype

[ErrorStatus, errstring] ZygoErrorStatusStringGet (axis)

Parameters

Input parameters

(Int32) axis: axis

Output parameters

(string) ErrorStatus: ErrorStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoErrorStatusStringGet command which is used to Return Zygo Interferometer Error Status description. Refer to the XPS Programmer's manual to get the command description.

ZygoErrorStatusGet

Syntax

C# prototype

int ZygoErrorStatusGet(Int32 axis, out string ErrorStatus, out string errstring)

Python prototype

[ErrorStatus, errstring] ZygoErrorStatusGet (axis)

Parameters

Input parameters

(Int32) axis: axis

Output parameters

(string) ErrorStatus: ErrorStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoErrorStatusGet command which is used to Return Zygo Interferometer Error Status code. Refer to the XPS Programmer's manual to get the command description.

ZygoStatusStringGet

Syntax

C# prototype

int ZygoStatusStringGet(Int32 axis, out string ErrorStatus, out string errstring)

Python prototype

[ErrorStatus, errstring] ZygoStatusStringGet (axis)

Parameters

Input parameters

(Int32) axis: axis

Output parameters

(string) ErrorStatus: ErrorStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoStatusStringGet command which is used to Return Zygo Interferometer Status description. Refer to the XPS Programmer's manual to get the command description.

ZygoStatusGet

Syntax

C# prototype

int ZygoStatusGet(Int32 axis, out string ErrorStatus, out string errstring)

Python prototype

[ErrorStatus, errstring] ZygoStatusGet (axis)

Parameters

Input parameters

(Int32) axis: axis

Output parameters

(string) ErrorStatus: ErrorStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoStatusGet command which is used to Return Zygo Interferometer Status code. Refer to the XPS Programmer's manual to get the command description.

ZygoRegisterSet

Syntax

C# prototype

int ZygoRegisterSet(string PositionerName, Int32 Register, Int32 Value, out string errstring)

Python prototype

[errstring] ZygoRegisterSet (PositionerName, Register, Value)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) Register: Register

(Int32) Value: Value

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoRegisterSet command which is used to . Refer to the XPS Programmer's manual to get the command description.

ZygoRegisterGet

Syntax

C# prototype

int ZygoRegisterGet(string PositionerName, Int32 Register, out Int32 Value, out string errstring)

Python prototype

[Value, errstring] ZygoRegisterGet (PositionerName, Register)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) Register: Register

Output parameters

(Int32_i) Value: Value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ZygoRegisterGet command which is used to . Refer to the XPS Programmer's manual to get the command description.

EventAdd

Syntax

C# prototype

```
int EventAdd(string PositionerName, string EventName, string EventParameter, string ActionName, string
ActionParameter1, string ActionParameter2, string ActionParameter3, out string errstring)
```

Python prototype

```
[errstring] EventAdd (PositionerName, EventName, EventParameter, ActionName, ActionParameter1,
ActionParameter2, ActionParameter3)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EventName: EventName

(string) EventParameter: EventParameter

(string) ActionName: ActionName

(string) ActionParameter1: ActionParameter1

(string) ActionParameter2: ActionParameter2

(string) ActionParameter3: ActionParameter3

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventAdd command which is used to **** OBSOLETE **** Add an event. Refer to the XPS Programmer's manual to get the command description.

EventGet

Syntax

C# prototype

```
int EventGet(string PositionerName, out string EventsAndActionsList, out string errstring)
```

Python prototype

```
[EventsAndActionsList, errstring] EventGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) EventsAndActionsList: EventsAndActionsList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventGet command which is used to **** OBSOLETE **** Read events and actions list. Refer to the XPS Programmer's manual to get the command description.

EventRemove

Syntax

C# prototype

```
int EventRemove(string PositionerName, string EventName, string EventParameter, out string errstring)
```

Python prototype

```
[errstring] EventRemove (PositionerName, EventName, EventParameter)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EventName: EventName

(string) EventParameter: EventParameter

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventRemove command which is used to **** OBSOLETE **** Delete an event. Refer to the XPS Programmer's manual to get the command description.

EventWait

Syntax

C# prototype

```
int EventWait(string PositionerName, string EventName, string EventParameter, out string errstring)
```

Python prototype

```
[errstring] EventWait (PositionerName, EventName, EventParameter)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EventName: EventName

(string) EventParameter: EventParameter

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventWait command which is used to **** OBSOLETE **** Wait an event. Refer to the XPS Programmer's manual to get the command description.

EventActionSetAndStart

Syntax

C# prototype

```
int EventActionSetAndStart(string ExtendedEventName, string EventParameter1, string EventParameter2, string
EventParameter3, string EventParameter4, string ExtendedActionName, string ActionParameter1, string
ActionParameter2, string ActionParameter3, string ActionParameter4, out Int32 ID, out string errstring)
```

Python prototype

```
[ID, errstring] EventActionSetAndStart (ExtendedEventName, EventParameter1, EventParameter2, EventParameter3,
EventParameter4, ExtendedActionName, ActionParameter1, ActionParameter2, ActionParameter3,
ActionParameter4)
```

Parameters

Input parameters

(string) ExtendedEventName: ExtendedEventName

(string) EventParameter1: EventParameter1

(string) EventParameter2: EventParameter2

(string) EventParameter3: EventParameter3

(string) EventParameter4: EventParameter4

(string) ExtendedActionName: ExtendedActionName

(string) ActionParameter1: ActionParameter1

(string) ActionParameter2: ActionParameter2

(string) ActionParameter3: ActionParameter3

(string) ActionParameter4: ActionParameter4

Output parameters

(Int32_i) ID: ID

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventActionSetAndStart command which is used to Set one event and one action, Launch the event and action configuration and return an ID. Refer to the XPS Programmer's manual to get the command description.

EventExtendedConfigurationTriggerSet

Syntax

C# prototype

```
int EventExtendedConfigurationTriggerSet(string[] ExtendedEventName, string[] EventParameter1, string[]  
EventParameter2, string[] EventParameter3, string[] EventParameter4, out string errstring)
```

Python prototype

```
[errstring] EventExtendedConfigurationTriggerSet (ExtendedEventName, EventParameter1, EventParameter2,  
EventParameter3, EventParameter4)
```

Parameters

Input parameters

(string[]) ExtendedEventName: ExtendedEventName

(string[]) EventParameter1: EventParameter1

(string[]) EventParameter2: EventParameter2

(string[]) EventParameter3: EventParameter3

(string[]) EventParameter4: EventParameter4

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedConfigurationTriggerSet command which is used to Configure one or several events. Refer to the XPS Programmer's manual to get the command description.

EventExtendedConfigurationTriggerGet

Syntax

C# prototype

```
int EventExtendedConfigurationTriggerGet(out string EventTriggerConfiguration, out string errstring)
```

Python prototype

```
[EventTriggerConfiguration, errstring] EventExtendedConfigurationTriggerGet ()
```

Parameters

Input parameters

None

Output parameters

(string) EventTriggerConfiguration: EventTriggerConfiguration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedConfigurationTriggerGet command which is used to Read the event configuration. Refer to the XPS Programmer's manual to get the command description.

EventExtendedConfigurationActionSet

Syntax

C# prototype

```
int EventExtendedConfigurationActionSet(string[] ExtendedActionName, string[] ActionParameter1, string[]
ActionParameter2, string[] ActionParameter3, string[] ActionParameter4, out string errstring)
```

Python prototype

```
[errstring] EventExtendedConfigurationActionSet (ExtendedActionName, ActionParameter1, ActionParameter2,
ActionParameter3, ActionParameter4)
```

Parameters

Input parameters

(string[]) ExtendedActionName: ExtendedActionName

(string[]) ActionParameter1: ActionParameter1

(string[]) ActionParameter2: ActionParameter2

(string[]) ActionParameter3: ActionParameter3

(string[]) ActionParameter4: ActionParameter4

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedConfigurationActionSet command which is used to Configure one or several actions. Refer to the XPS Programmer's manual to get the command description.

EventExtendedConfigurationActionGet

Syntax

C# prototype

```
int EventExtendedConfigurationActionGet(out string ActionConfiguration, out string errstring)
```

Python prototype

```
[ActionConfiguration, errstring] EventExtendedConfigurationActionGet ()
```

Parameters

Input parameters

None

Output parameters

(string) ActionConfiguration: ActionConfiguration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedConfigurationActionGet command which is used to Read the action configuration. Refer to the XPS Programmer's manual to get the command description.

EventExtendedStart

Syntax

C# prototype

```
int EventExtendedStart(out Int32 ID, out string errstring)
```

Python prototype

```
[ID, errstring] EventExtendedStart ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) ID: ID

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedStart command which is used to Launch the last event and action configuration and return an ID. Refer to the XPS Programmer's manual to get the command description.

EventExtendedAllGet

Syntax

C# prototype

```
int EventExtendedAllGet(out string EventActionConfigurations, out string errstring)
```

Python prototype

```
[EventActionConfigurations, errstring] EventExtendedAllGet ()
```

Parameters

Input parameters

None

Output parameters

(string) EventActionConfigurations: EventActionConfigurations

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedAllGet command which is used to Read all event and action configurations. Refer to the XPS Programmer's manual to get the command description.

EventExtendedGet

Syntax

C# prototype

```
int EventExtendedGet(Int32 ID, out string EventTriggerConfiguration, out string ActionConfiguration, out string errstring)
```

Python prototype

```
[EventTriggerConfiguration, ActionConfiguration, errstring] EventExtendedGet (ID)
```

Parameters

Input parameters

(Int32) ID: ID

Output parameters

(string) EventTriggerConfiguration: EventTriggerConfiguration

(string) ActionConfiguration: ActionConfiguration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedGet command which is used to Read the event and action configuration defined by ID. Refer to the XPS Programmer's manual to get the command description.

EventExtendedRemove

Syntax

C# prototype

int EventExtendedRemove(Int32 ID, out string errstring)

Python prototype

[errstring] EventExtendedRemove (ID)

Parameters

Input parameters

(Int32) ID: ID

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedRemove command which is used to Remove the event and action configuration defined by ID. Refer to the XPS Programmer's manual to get the command description.

EventExtendedWait

Syntax

C# prototype

```
int EventExtendedWait( out string errstring)
```

Python prototype

```
[errstring] EventExtendedWait ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventExtendedWait command which is used to Wait events from the last event configuration. Refer to the XPS Programmer's manual to get the command description.

GatheringConfigurationGet

Syntax

C# prototype

```
int GatheringConfigurationGet(out string Type, out string errstring)
```

Python prototype

```
[Type, errstring] GatheringConfigurationGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Type: Type

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringConfigurationGet command which is used to Read different mnemonic type. Refer to the XPS Programmer's manual to get the command description.

GatheringConfigurationSet

Syntax

C# prototype

```
int GatheringConfigurationSet(string[] Type, out string errstring)
```

Python prototype

```
[errstring] GatheringConfigurationSet (Type)
```

Parameters

Input parameters

(string[]) Type: Type

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringConfigurationSet command which is used to Configuration acquisition. Refer to the XPS Programmer's manual to get the command description.

GatheringCurrentNumberGet

Syntax

C# prototype

```
int GatheringCurrentNumberGet(out Int32 CurrentNumber, out Int32 MaximumSamplesNumber, out string errstring)
```

Python prototype

```
[CurrentNumber, MaximumSamplesNumber, errstring] GatheringCurrentNumberGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) CurrentNumber: CurrentNumber

(Int32_i) MaximumSamplesNumber: MaximumSamplesNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringCurrentNumberGet command which is used to Maximum number of samples and current number during acquisition. Refer to the XPS Programmer's manual to get the command description.

GatheringCurrentIndexGet

Syntax

C# prototype

```
int GatheringCurrentIndexGet(out Int32 CurrentIndex, out Int32 CurrentNumber, out string errstring)
```

Python prototype

```
[CurrentIndex, CurrentNumber, errstring] GatheringCurrentIndexGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) CurrentIndex: CurrentIndex

(Int32_i) CurrentNumber: CurrentNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringCurrentIndexGet command which is used to Current index of the last acquisition and current number of acquisitions. Refer to the XPS Programmer's manual to get the command description.

GatheringStopAndSave

Syntax

C# prototype

```
int GatheringStopAndSave( out string errstring)
```

Python prototype

```
[errstring] GatheringStopAndSave ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringStopAndSave command which is used to Stop acquisition and save data. Refer to the XPS Programmer's manual to get the command description.

GatheringDataAcquire

Syntax

C# prototype

```
int GatheringDataAcquire( out string errstring)
```

Python prototype

```
[errstring] GatheringDataAcquire ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringDataAcquire command which is used to Acquire a configured data. Refer to the XPS Programmer's manual to get the command description.

GatheringDataGet

Syntax

C# prototype

```
int GatheringDataGet(Int32 IndexPoint, out string DataBufferLine, out string errstring)
```

Python prototype

```
[DataBufferLine, errstring] GatheringDataGet (IndexPoint)
```

Parameters

Input parameters

(Int32) IndexPoint: IndexPoint

Output parameters

(string) DataBufferLine: DataBufferLine

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringDataGet command which is used to Get a data line from gathering buffer. Refer to the XPS Programmer's manual to get the command description.

GatheringDataMultipleLinesGet

Syntax

C# prototype

```
int GatheringDataMultipleLinesGet(Int32 IndexPoint, Int32 NumberOfLines, out string DataBufferLine, out string errstring)
```

Python prototype

```
[DataBufferLine, errstring] GatheringDataMultipleLinesGet (IndexPoint, NumberOfLines)
```

Parameters

Input parameters

(Int32) IndexPoint: IndexPoint

(Int32) NumberOfLines: NumberOfLines

Output parameters

(string) DataBufferLine: DataBufferLine

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringDataMultipleLinesGet command which is used to Get multiple data lines from gathering buffer. Refer to the XPS Programmer's manual to get the command description.

GatheringReset

Syntax

C# prototype

```
int GatheringReset( out string errstring)
```

Python prototype

```
[errstring] GatheringReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringReset command which is used to Empty the gathered data in memory to start new gathering from scratch. Refer to the XPS Programmer's manual to get the command description.

GatheringRun

Syntax

C# prototype

```
int GatheringRun(Int32 DataNumber, Int32 Divisor, out string errstring)
```

Python prototype

```
[errstring] GatheringRun (DataNumber, Divisor)
```

Parameters

Input parameters

(Int32) DataNumber: DataNumber

(Int32) Divisor: Divisor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringRun command which is used to Start a new gathering. Refer to the XPS Programmer's manual to get the command description.

GatheringRunAppend

Syntax

C# prototype

```
int GatheringRunAppend( out string errstring)
```

Python prototype

```
[errstring] GatheringRunAppend ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringRunAppend command which is used to Re-start the stopped gathering to add new data. Refer to the XPS Programmer's manual to get the command description.

GatheringStop

Syntax

C# prototype

```
int GatheringStop( out string errstring)
```

Python prototype

```
[errstring] GatheringStop ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringStop command which is used to Stop the data gathering . Refer to the XPS Programmer's manual to get the command description.

GatheringExternalConfigurationSet

Syntax

C# prototype

```
int GatheringExternalConfigurationSet(string[] Type, out string errstring)
```

Python prototype

```
[errstring] GatheringExternalConfigurationSet (Type)
```

Parameters

Input parameters

(string[]) Type: Type

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExternalConfigurationSet command which is used to Configuration acquisition. Refer to the XPS Programmer's manual to get the command description.

GatheringExternalConfigurationGet

Syntax

C# prototype

```
int GatheringExternalConfigurationGet(out string Type, out string errstring)
```

Python prototype

```
[Type, errstring] GatheringExternalConfigurationGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Type: Type

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExternalConfigurationGet command which is used to Read different mnemonic type. Refer to the XPS Programmer's manual to get the command description.

GatheringExternalCurrentNumberGet

Syntax

C# prototype

```
int GatheringExternalCurrentNumberGet(out Int32 CurrentNumber, out Int32 MaximumSamplesNumber, out string errstring)
```

Python prototype

```
[CurrentNumber, MaximumSamplesNumber, errstring] GatheringExternalCurrentNumberGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) CurrentNumber: CurrentNumber

(Int32_i) MaximumSamplesNumber: MaximumSamplesNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExternalCurrentNumberGet command which is used to Maximum number of samples and current number during acquisition. Refer to the XPS Programmer's manual to get the command description.

GatheringExternalDataGet

Syntax

C# prototype

```
int GatheringExternalDataGet(Int32 IndexPoint, out string DataBufferLine, out string errstring)
```

Python prototype

```
[DataBufferLine, errstring] GatheringExternalDataGet (IndexPoint)
```

Parameters

Input parameters

(Int32) IndexPoint: IndexPoint

Output parameters

(string) DataBufferLine: DataBufferLine

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExternalDataGet command which is used to Get a data line from external gathering buffer. Refer to the XPS Programmer's manual to get the command description.

GatheringExternalStopAndSave

Syntax

C# prototype

```
int GatheringExternalStopAndSave( out string errstring)
```

Python prototype

```
[errstring] GatheringExternalStopAndSave ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExternalStopAndSave command which is used to Stop acquisition and save data. Refer to the XPS Programmer's manual to get the command description.

GPIOAnalogGet

Syntax

C# prototype

```
int GPIOAnalogGet(string[] GPIOName, out double[] AnalogValue, out string errstring)
```

Python prototype

```
[AnalogValue, errstring] GPIOAnalogGet (GPIOName)
```

Parameters

Input parameters

(string[]) GPIOName: GPIOName

Output parameters

(double[]) AnalogValue: AnalogValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIOAnalogGet command which is used to Read analog input or analog output for one or few input. Refer to the XPS Programmer's manual to get the command description.

GPIOAnalogSet

Syntax

C# prototype

```
int GPIOAnalogSet(string[] GPIOName, double[] AnalogOutputValue, out string errstring)
```

Python prototype

```
[errstring] GPIOAnalogSet (GPIOName, AnalogOutputValue)
```

Parameters

Input parameters

(string[]) GPIOName: GPIOName

(double[]) AnalogOutputValue: AnalogOutputValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIOAnalogSet command which is used to Set analog output for one or few output. Refer to the XPS Programmer's manual to get the command description.

GPIOAnalogGainGet

Syntax

C# prototype

```
int GPIOAnalogGainGet(string[] GPIOName, out Int32[] AnalogInputGainValue, out string errstring)
```

Python prototype

```
[AnalogInputGainValue, errstring] GPIOAnalogGainGet (GPIOName)
```

Parameters

Input parameters

(string[]) GPIOName: GPIOName

Output parameters

(Int32_i[]) AnalogInputGainValue: AnalogInputGainValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIOAnalogGainGet command which is used to Read user analog input gain 1, 2, 4 or 8, for one or few input. Refer to the XPS Programmer's manual to get the command description.

GPIOAnalogGainSet

Syntax

C# prototype

```
int GPIOAnalogGainSet(string[] GPIOName, Int32[] AnalogInputGainValue, out string errstring)
```

Python prototype

```
[errstring] GPIOAnalogGainSet (GPIOName, AnalogInputGainValue)
```

Parameters

Input parameters

(string[]) GPIOName: GPIOName

(Int32) AnalogInputGainValue: AnalogInputGainValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIOAnalogGainSet command which is used to Set user analog input gain 1, 2, 4 or 8, for one or few input. Refer to the XPS Programmer's manual to get the command description.

GPIOAnalogRangeConfigurationGet

Syntax

C# prototype

```
int GPIOAnalogRangeConfigurationGet(string GPIOName, out double DACRange, out string errstring)
```

Python prototype

```
[DACRange, errstring] GPIOAnalogRangeConfigurationGet (GPIOName)
```

Parameters

Input parameters

(string) GPIOName: GPIOName

Output parameters

(double) DACRange: DACRange

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIOAnalogRangeConfigurationGet command which is used to Get GPIO DAC range configuration for DAC . Refer to the XPS Programmer's manual to get the command description.

GPIOAnalogRangeConfigurationSet

Syntax

C# prototype

int GPIOAnalogRangeConfigurationSet(string GPIOName, double DACRange, out string errstring)

Python prototype

[errstring] GPIOAnalogRangeConfigurationSet (GPIOName, DACRange)

Parameters

Input parameters

(string) GPIOName: GPIOName

(double) DACRange: DACRange

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIOAnalogRangeConfigurationSet command which is used to Set GPIO DAC range configuration for DAC . Refer to the XPS Programmer's manual to get the command description.

GPIODigitalGet

Syntax

C# prototype

```
int GPIODigitalGet(string GPIOName, out UInt16 DigitalValue, out string errstring)
```

Python prototype

```
[DigitalValue, errstring] GPIODigitalGet (GPIOName)
```

Parameters

Input parameters

(string) GPIOName: GPIOName

Output parameters

(UInt16) DigitalValue: DigitalValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIODigitalGet command which is used to Read digital output or digital input . Refer to the XPS Programmer's manual to get the command description.

GPIODigitalSet

Syntax

C# prototype

```
int GPIODigitalSet(string GPIOName, UInt16 Mask, UInt16 DigitalOutputValue, out string errstring)
```

Python prototype

```
[errstring] GPIODigitalSet (GPIOName, Mask, DigitalOutputValue)
```

Parameters

Input parameters

(string) GPIOName: GPIOName

(UInt16) Mask: Mask

(UInt16) DigitalOutputValue: DigitalOutputValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIODigitalSet command which is used to Set Digital Output for one or few output TTL. Refer to the XPS Programmer's manual to get the command description.

GPIODigitalPulseWidthGet

Syntax

C# prototype

int GPIODigitalPulseWidthGet(string GPIOName, out double PulseWidth, out string errstring)

Python prototype

[PulseWidth, errstring] GPIODigitalPulseWidthGet (GPIOName)

Parameters

Input parameters

(string) GPIOName: GPIOName

Output parameters

(double) PulseWidth: PulseWidth

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIODigitalPulseWidthGet command which is used to Read current GPIO pulse width . Refer to the XPS Programmer's manual to get the command description.

GPIODigitalPulseWidthSet

Syntax

C# prototype

```
int GPIODigitalPulseWidthSet(string GPIOName, double PulseWidth, out string errstring)
```

Python prototype

```
[errstring] GPIODigitalPulseWidthSet (GPIOName, PulseWidth)
```

Parameters

Input parameters

(string) GPIOName: GPIOName

(double) PulseWidth: PulseWidth

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GPIODigitalPulseWidthSet command which is used to Set GPIO pulse width . Refer to the XPS Programmer's manual to get the command description.

GroupAccelerationCurrentGet

Syntax

C# prototype

```
int GroupAccelerationCurrentGet(string GroupName, out double[] CurrentAcceleration, Int32 nbltems, out string errstring)
```

Python prototype

```
[CurrentAcceleration, errstring] GroupAccelerationCurrentGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) CurrentAcceleration: CurrentAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupAccelerationCurrentGet command which is used to Return current acceleration. Refer to the XPS Programmer's manual to get the command description.

GroupAccelerationSetpointGet

Syntax

C# prototype

```
int GroupAccelerationSetpointGet(string GroupName, out double[] SetpointAcceleration, Int32 nbltems, out string errstring)
```

Python prototype

```
[SetpointAcceleration, errstring] GroupAccelerationSetpointGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) SetpointAcceleration: SetpointAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupAccelerationSetpointGet command which is used to Return setpoint acceleration. Refer to the XPS Programmer's manual to get the command description.

GroupAnalogTrackingModeEnable

Syntax

C# prototype

```
int GroupAnalogTrackingModeEnable(string GroupName, string Type, out string errstring)
```

Python prototype

```
[errstring] GroupAnalogTrackingModeEnable (GroupName, Type)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) Type: Type

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupAnalogTrackingModeEnable command which is used to Enable analog tracking mode. Refer to the XPS Programmer's manual to get the command description.

GroupAnalogTrackingModeDisable

Syntax

C# prototype

```
int GroupAnalogTrackingModeDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupAnalogTrackingModeDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupAnalogTrackingModeDisable command which is used to Disable analog tracking mode. Refer to the XPS Programmer's manual to get the command description.

GroupCorrectorOutputGet

Syntax

C# prototype

```
int GroupCorrectorOutputGet(string GroupName, out double[] CorrectorOutput, Int32 nbItems, out string errstring)
```

Python prototype

```
[CorrectorOutput, errstring] GroupCorrectorOutputGet (GroupName, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) CorrectorOutput: CorrectorOutput

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupCorrectorOutputGet command which is used to Return corrector output. Refer to the XPS Programmer's manual to get the command description.

GroupCurrentFollowingErrorGet

Syntax

C# prototype

```
int GroupCurrentFollowingErrorGet(string GroupName, out double[] CurrentFollowingError, Int32 nbltems, out string errstring)
```

Python prototype

```
[CurrentFollowingError, errstring] GroupCurrentFollowingErrorGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) CurrentFollowingError: CurrentFollowingError

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupCurrentFollowingErrorGet command which is used to Return current following error. Refer to the XPS Programmer's manual to get the command description.

GroupHomeSearch

Syntax

C# prototype

```
int GroupHomeSearch(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupHomeSearch (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupHomeSearch command which is used to Start home search sequence. Refer to the XPS Programmer's manual to get the command description.

GroupHomeSearchAndRelativeMove

Syntax

C# prototype

```
int GroupHomeSearchAndRelativeMove(string GroupName, double[] TargetDisplacement, Int32 nbItems, out
string errstring)
```

Python prototype

```
[errstring] GroupHomeSearchAndRelativeMove (GroupName, TargetDisplacement, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetDisplacement: TargetDisplacement

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupHomeSearchAndRelativeMove command which is used to Start home search sequence and execute a displacement. Refer to the XPS Programmer's manual to get the command description.

GroupInitialize

Syntax

C# prototype

```
int GroupInitialize(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupInitialize (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupInitialize command which is used to Start the initialization. Refer to the XPS Programmer's manual to get the command description.

GroupInitializeNoEncoderReset

Syntax

C# prototype

```
int GroupInitializeNoEncoderReset(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupInitializeNoEncoderReset (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupInitializeNoEncoderReset command which is used to Initialize group without encoder reset. Refer to the XPS Programmer's manual to get the command description.

GroupInitializeWithEncoderCalibration

Syntax

C# prototype

```
int GroupInitializeWithEncoderCalibration(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupInitializeWithEncoderCalibration (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupInitializeWithEncoderCalibration command which is used to Initialize group with encoder calibration. Refer to the XPS Programmer's manual to get the command description.

GroupGantryModeGet

Syntax

C# prototype

```
int GroupGantryModeGet(string GroupName, out string Option, out string errstring)
```

Python prototype

```
[Option, errstring] GroupGantryModeGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) Option: Option

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupGantryModeGet command which is used to Return current group gantry mode . Refer to the XPS Programmer's manual to get the command description.

GroupGantryModeSet

Syntax

C# prototype

```
int GroupGantryModeSet(string GroupName, string Option, out string errstring)
```

Python prototype

```
[errstring] GroupGantryModeSet (GroupName, Option)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) Option: Option

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupGantryModeSet command which is used to Set group gantry mode . Refer to the XPS Programmer's manual to get the command description.

GroupInterlockDisable

Syntax

C# prototype

```
int GroupInterlockDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupInterlockDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupInterlockDisable command which is used to Disable group interlock. Refer to the XPS Programmer's manual to get the command description.

GroupInterlockEnable

Syntax

C# prototype

```
int GroupInterlockEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupInterlockEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupInterlockEnable command which is used to Enable group interlock. Refer to the XPS Programmer's manual to get the command description.

GroupJogParametersSet

Syntax

C# prototype

```
int GroupJogParametersSet(string GroupName, double[] Velocity, double[] Acceleration, Int32 nbItems, out string errstring)
```

Python prototype

```
[errstring] GroupJogParametersSet (GroupName, Velocity, Acceleration, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) Velocity: Velocity

(double[]) Acceleration: Acceleration

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupJogParametersSet command which is used to Change jog parameters and activate the modifications. Refer to the XPS Programmer's manual to get the command description.

GroupJogParametersGet

Syntax

C# prototype

```
int GroupJogParametersGet(string GroupName, out double[] Velocity, out double[] Acceleration, Int32 nbItems,  
out string errstring)
```

Python prototype

```
[Velocity, Acceleration, errstring] GroupJogParametersGet (GroupName, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) Velocity: Velocity

(double[]) Acceleration: Acceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupJogParametersGet command which is used to Get jog parameters. Refer to the XPS Programmer's manual to get the command description.

GroupJogCurrentGet

Syntax

C# prototype

```
int GroupJogCurrentGet(string GroupName, out double[] Velocity, out double[] Acceleration, Int32 nbltems, out string errstring)
```

Python prototype

```
[Velocity, Acceleration, errstring] GroupJogCurrentGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) Velocity: Velocity

(double[]) Acceleration: Acceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupJogCurrentGet command which is used to Get current jog parameters. Refer to the XPS Programmer's manual to get the command description.

GroupJogModeEnable

Syntax

C# prototype

```
int GroupJogModeEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupJogModeEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupJogModeEnable command which is used to Enable jog mode. Refer to the XPS Programmer's manual to get the command description.

GroupJogModeDisable

Syntax

C# prototype

```
int GroupJogModeDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupJogModeDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupJogModeDisable command which is used to Disable jog mode. Refer to the XPS Programmer's manual to get the command description.

GroupKill

Syntax

C# prototype

```
int GroupKill(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupKill (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupKill command which is used to Kill the group. Refer to the XPS Programmer's manual to get the command description.

GroupMotionDisable

Syntax

C# prototype

```
int GroupMotionDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupMotionDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMotionDisable command which is used to Disable motion. Refer to the XPS Programmer's manual to get the command description.

GroupMotionEnable

Syntax

C# prototype

```
int GroupMotionEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupMotionEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMotionEnable command which is used to Enable motion. Refer to the XPS Programmer's manual to get the command description.

GroupMotionStatusGet

Syntax

C# prototype

```
int GroupMotionStatusGet(string GroupName, out Int32[] Status, Int32 nbItems, out string errstring)
```

Python prototype

```
[Status, errstring] GroupMotionStatusGet (GroupName, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(Int32_i[]) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMotionStatusGet command which is used to Get group or positioner status. Refer to the XPS Programmer's manual to get the command description.

GroupMoveAbort

Syntax

C# prototype

```
int GroupMoveAbort(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupMoveAbort (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMoveAbort command which is used to Abort move. Refer to the XPS Programmer's manual to get the command description.

GroupExternalProfilerEnable

Syntax

C# prototype

```
int GroupExternalProfilerEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupExternalProfilerEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupExternalProfilerEnable command which is used to Enable external profiler. Refer to the XPS Programmer's manual to get the command description.

GroupExternalProfilerDisable

Syntax

C# prototype

```
int GroupExternalProfilerDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupExternalProfilerDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupExternalProfilerDisable command which is used to Disable external profiler. Refer to the XPS Programmer's manual to get the command description.

GroupMoveAbortFast

Syntax

C# prototype

```
int GroupMoveAbortFast(string GroupName, Int32 AccelerationMultiplier, out string errstring)
```

Python prototype

```
[errstring] GroupMoveAbortFast (GroupName, AccelerationMultiplier)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) AccelerationMultiplier: AccelerationMultiplier

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMoveAbortFast command which is used to Abort quickly a move. Refer to the XPS Programmer's manual to get the command description.

GroupGoToNotReferenced

Syntax

C# prototype

```
int GroupGoToNotReferenced(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupGoToNotReferenced (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupGoToNotReferenced command which is used to Go to Not Referenced state. Refer to the XPS Programmer's manual to get the command description.

GroupMoveAbsolute

Syntax

C# prototype

```
int GroupMoveAbsolute(string GroupName, double[] TargetPosition, Int32 nbItems, out string errstring)
```

Python prototype

```
[errstring] GroupMoveAbsolute (GroupName, TargetPosition, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetPosition: TargetPosition

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMoveAbsolute command which is used to Do an absolute move. Refer to the XPS Programmer's manual to get the command description.

GroupMoveEndWait

Syntax

C# prototype

```
int GroupMoveEndWait(string GroupName, double TimeOutMs, double XPosition, double YPosition, out string errstring)
```

Python prototype

```
[errstring] GroupMoveEndWait (GroupName, TimeOutMs, XPosition, YPosition)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) TimeOutMs: TimeOutMs

(double) XPosition: XPosition

(double) YPosition: YPosition

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMoveEndWait command which is used to Wait the end of motion. Refer to the XPS Programmer's manual to get the command description.

GroupMoveRelative

Syntax

C# prototype

```
int GroupMoveRelative(string GroupName, double[] TargetDisplacement, Int32 nbItems, out string errstring)
```

Python prototype

```
[errstring] GroupMoveRelative (GroupName, TargetDisplacement, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetDisplacement: TargetDisplacement

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMoveRelative command which is used to Do a relative move. Refer to the XPS Programmer's manual to get the command description.

GroupMoveRelativeSimulated

Syntax

C# prototype

```
int GroupMoveRelativeSimulated(string GroupName, double[] TargetDisplacement, Int32 nbltems, out string errstring)
```

Python prototype

```
[errstring] GroupMoveRelativeSimulated (GroupName, TargetDisplacement, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) TargetDisplacement: TargetDisplacement

(Int32) nbltems: nbltems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMoveRelativeSimulated command which is used to Do a simulated relative move. Refer to the XPS Programmer's manual to get the command description.

GroupPositionCurrentGet

Syntax

C# prototype

```
int GroupPositionCurrentGet(string GroupName, out double[] CurrentEncoderPosition, Int32 nblItems, out string errstring)
```

Python prototype

```
[CurrentEncoderPosition, errstring] GroupPositionCurrentGet (GroupName, nblItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nblItems: nblItems

Output parameters

(double[]) CurrentEncoderPosition: CurrentEncoderPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupPositionCurrentGet command which is used to Return current position. Refer to the XPS Programmer's manual to get the command description.

GroupPositionSetpointGet

Syntax

C# prototype

```
int GroupPositionSetpointGet(string GroupName, out double[] SetPointPosition, Int32 nbltems, out string errstring)
```

Python prototype

```
[SetPointPosition, errstring] GroupPositionSetpointGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) SetPointPosition: SetPointPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupPositionSetpointGet command which is used to Return setpoint position. Refer to the XPS Programmer's manual to get the command description.

GroupPositionTargetGet

Syntax

C# prototype

```
int GroupPositionTargetGet(string GroupName, out double[] TargetPosition, Int32 nbltems, out string errstring)
```

Python prototype

```
[TargetPosition, errstring] GroupPositionTargetGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) TargetPosition: TargetPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupPositionTargetGet command which is used to Return target position. Refer to the XPS Programmer's manual to get the command description.

GroupReferencingActionExecute

Syntax

C# prototype

```
int GroupReferencingActionExecute(string PositionerName, string ReferencingAction, string ReferencingSensor,  
double ReferencingParameter, out string errstring)
```

Python prototype

```
[errstring] GroupReferencingActionExecute (PositionerName, ReferencingAction, ReferencingSensor,  
ReferencingParameter)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) ReferencingAction: ReferencingAction

(string) ReferencingSensor: ReferencingSensor

(double) ReferencingParameter: ReferencingParameter

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupReferencingActionExecute command which is used to Execute action in referencing mode. Refer to the XPS Programmer's manual to get the command description.

GroupReferencingStart

Syntax

C# prototype

```
int GroupReferencingStart(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupReferencingStart (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupReferencingStart command which is used to Enter referencing mode. Refer to the XPS Programmer's manual to get the command description.

GroupReferencingStop

Syntax

C# prototype

```
int GroupReferencingStop(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupReferencingStop (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupReferencingStop command which is used to Exit referencing mode. Refer to the XPS Programmer's manual to get the command description.

GroupStatusGet

Syntax

C# prototype

```
int GroupStatusGet(string GroupName, out Int32 Status, out string errstring)
```

Python prototype

```
[Status, errstring] GroupStatusGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupStatusGet command which is used to Return group status. Refer to the XPS Programmer's manual to get the command description.

GroupStatusStringGet

Syntax

C# prototype

```
int GroupStatusStringGet(Int32 GroupStatusCode, out string GroupStatusString, out string errstring)
```

Python prototype

```
[GroupStatusString, errstring] GroupStatusStringGet (GroupStatusCode)
```

Parameters

Input parameters

(Int32) GroupStatusCode: GroupStatusCode

Output parameters

(string) GroupStatusString: GroupStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupStatusStringGet command which is used to Return group status string corresponding to the group status code. Refer to the XPS Programmer's manual to get the command description.

GroupVelocityCurrentGet

Syntax

C# prototype

```
int GroupVelocityCurrentGet(string GroupName, out double[] CurrentVelocity, Int32 nbItems, out string errstring)
```

Python prototype

```
[CurrentVelocity, errstring] GroupVelocityCurrentGet (GroupName, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) CurrentVelocity: CurrentVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupVelocityCurrentGet command which is used to Return current velocity. Refer to the XPS Programmer's manual to get the command description.

GroupVelocitySetpointGet

Syntax

C# prototype

```
int GroupVelocitySetpointGet(string GroupName, out double[] SetpointVelocity, Int32 nbItems, out string errstring)
```

Python prototype

```
[SetpointVelocity, errstring] GroupVelocitySetpointGet (GroupName, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) SetpointVelocity: SetpointVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupVelocitySetpointGet command which is used to Return setpoint velocity. Refer to the XPS Programmer's manual to get the command description.

ExternalInterlockErrorStringGet

Syntax

C# prototype

```
int ExternalInterlockErrorStringGet(Int32 InterlockErrorCode, out string InterlockErrorString, out string errstring)
```

Python prototype

```
[InterlockErrorString, errstring] ExternalInterlockErrorStringGet (InterlockErrorCode)
```

Parameters

Input parameters

(Int32) InterlockErrorCode: InterlockErrorCode

Output parameters

(string) InterlockErrorString: InterlockErrorString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalInterlockErrorStringGet command which is used to Return external interlock errors description. Refer to the XPS Programmer's manual to get the command description.

ExternalInterlockGroupErrorGet

Syntax

C# prototype

```
int ExternalInterlockGroupErrorGet(string GroupName, out Int32 GroupInterlockErrorCode, out string errstring)
```

Python prototype

```
[GroupInterlockErrorCode, errstring] ExternalInterlockGroupErrorGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) GroupInterlockErrorCode: GroupInterlockErrorCode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalInterlockGroupErrorGet command which is used to Return current external group interlock errors. Refer to the XPS Programmer's manual to get the command description.

ExternalInterlockLiftPinErrorGet

Syntax

C# prototype

```
int ExternalInterlockLiftPinErrorGet(string GroupName, out Int32 LiftPinInterlockErrorCode, out string errstring)
```

Python prototype

```
[LiftPinInterlockErrorCode, errstring] ExternalInterlockLiftPinErrorGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) LiftPinInterlockErrorCode: LiftPinInterlockErrorCode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalInterlockLiftPinErrorGet command which is used to Return current external lift pin interlock errors. Refer to the XPS Programmer's manual to get the command description.

KillAll

Syntax

C# prototype

int KillAll(out string errstring)

Python prototype

[errstring] KillAll ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous KillAll command which is used to Put all groups in NOT_INITIALIZED state. Refer to the XPS Programmer's manual to get the command description.

GroupInitializeFromRegistration

Syntax

C# prototype

```
int GroupInitializeFromRegistration(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupInitializeFromRegistration (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupInitializeFromRegistration command which is used to Start the initialization without large moves after GroupKillWithRegistration. Refer to the XPS Programmer's manual to get the command description.

GroupReadyAtPosition

Syntax

C# prototype

```
int GroupReadyAtPosition(string GroupName, double[] Position, Int32 nbltems, out string errstring)
```

Python prototype

```
[errstring] GroupReadyAtPosition (GroupName, Position, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double[]) Position: Position

(Int32) nbltems: nbltems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupReadyAtPosition command which is used to Go to ready without home search sequence. Refer to the XPS Programmer's manual to get the command description.

GroupKillWithRegistration

Syntax

C# prototype

```
int GroupKillWithRegistration(string GroupName, out string errstring)
```

Python prototype

```
[errstring] GroupKillWithRegistration (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupKillWithRegistration command which is used to Kill the group with positions registration. Refer to the XPS Programmer's manual to get the command description.

PositionerDriverStatusGet

Syntax

C# prototype

```
int PositionerDriverStatusGet(string PositionerName, out UInt32 DriverStatus, out string errstring)
```

Python prototype

```
[DriverStatus, errstring] PositionerDriverStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(UInt32_I) DriverStatus: DriverStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerDriverStatusGet command which is used to Read positioner driver status. Refer to the XPS Programmer's manual to get the command description.

PositionerDriverStatusStringGet

Syntax

C# prototype

```
int PositionerDriverStatusStringGet(UInt32 PositionerDriverStatus, out string PositionerDriverStatusString, out string errstring)
```

Python prototype

```
[PositionerDriverStatusString, errstring] PositionerDriverStatusStringGet (PositionerDriverStatus)
```

Parameters

Input parameters

(UInt32) PositionerDriverStatus: PositionerDriverStatus

Output parameters

(string) PositionerDriverStatusString: PositionerDriverStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerDriverStatusStringGet command which is used to Return driver status string from the driver status code. Refer to the XPS Programmer's manual to get the command description.

PositionerErrorGet

Syntax

C# prototype

```
int PositionerErrorGet(string PositionerName, out Int32 ErrorCode, out string errstring)
```

Python prototype

```
[ErrorCode, errstring] PositionerErrorGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) ErrorCode: ErrorCode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerErrorGet command which is used to Read and clear positioner error code. Refer to the XPS Programmer's manual to get the command description.

PositionerErrorRead

Syntax

C# prototype

```
int PositionerErrorRead(string PositionerName, out Int32 ErrorCode, out string errstring)
```

Python prototype

```
[ErrorCode, errstring] PositionerErrorRead (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) ErrorCode: ErrorCode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerErrorRead command which is used to Read only positioner error code without clear it. Refer to the XPS Programmer's manual to get the command description.

PositionerErrorStringGet

Syntax

C# prototype

```
int PositionerErrorStringGet(Int32 PositionerErrorCode, out string PositionerErrorString, out string errstring)
```

Python prototype

```
[PositionerErrorString, errstring] PositionerErrorStringGet (PositionerErrorCode)
```

Parameters

Input parameters

(Int32) PositionerErrorCode: PositionerErrorCode

Output parameters

(string) PositionerErrorString: PositionerErrorString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerErrorStringGet command which is used to Return positioner error string from positioner error code. Refer to the XPS Programmer's manual to get the command description.

PositionerHardwareStatusGet

Syntax

C# prototype

```
int PositionerHardwareStatusGet(string PositionerName, out Int32 HardwareStatus, out string errstring)
```

Python prototype

```
[HardwareStatus, errstring] PositionerHardwareStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) HardwareStatus: HardwareStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerHardwareStatusGet command which is used to Read positioner hardware status. Refer to the XPS Programmer's manual to get the command description.

PositionerHardwareStatusStringGet

Syntax

C# prototype

```
int PositionerHardwareStatusStringGet(Int32 PositionerHardwareStatus, out string PositionerHardwareStatusString,  
out string errstring)
```

Python prototype

```
[PositionerHardwareStatusString, errstring] PositionerHardwareStatusStringGet (PositionerHardwareStatus)
```

Parameters

Input parameters

(Int32) PositionerHardwareStatus: PositionerHardwareStatus

Output parameters

(string) PositionerHardwareStatusString: PositionerHardwareStatusString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerHardwareStatusStringGet command which is used to Return hardware status string from hardware status code. Refer to the XPS Programmer's manual to get the command description.

PositionersEncoderIndexDifferenceGet

Syntax

C# prototype

int PositionersEncoderIndexDifferenceGet(string PositionerName, out double distance, out string errstring)

Python prototype

[distance, errstring] PositionersEncoderIndexDifferenceGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) distance: distance

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionersEncoderIndexDifferenceGet command which is used to Return the difference between index of primary axis and secondary axis . Refer to the XPS Programmer's manual to get the command description.

PositionerGantryEndReferencingPositionGet

Syntax

C# prototype

```
int PositionerGantryEndReferencingPositionGet(string PositionerName, out double Position, out string errstring)
```

Python prototype

```
[Position, errstring] PositionerGantryEndReferencingPositionGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Position: Position

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerGantryEndReferencingPositionGet command which is used to Return the secondary axis position at the end of home. Refer to the XPS Programmer's manual to get the command description.

PositionerStageParameterGet

Syntax

C# prototype

```
int PositionerStageParameterGet(string PositionerName, string ParameterName, out string ParameterValue, out string errstring)
```

Python prototype

```
[ParameterValue, errstring] PositionerStageParameterGet (PositionerName, ParameterName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) ParameterName: ParameterName

Output parameters

(string) ParameterValue: ParameterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerStageParameterGet command which is used to Return the stage parameter. Refer to the XPS Programmer's manual to get the command description.

PositionerStageParameterSet

Syntax

C# prototype

```
int PositionerStageParameterSet(string PositionerName, string ParameterName, string ParameterValue, out string errstring)
```

Python prototype

```
[errstring] PositionerStageParameterSet (PositionerName, ParameterName, ParameterValue)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) ParameterName: ParameterName

(string) ParameterValue: ParameterValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerStageParameterSet command which is used to Save the stage parameter. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorTypeGet

Syntax

C# prototype

```
int PositionerCorrectorTypeGet(string PositionerName, out string CorrectorType, out string errstring)
```

Python prototype

```
[CorrectorType, errstring] PositionerCorrectorTypeGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) CorrectorType: CorrectorType

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorTypeGet command which is used to Read corrector type. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorFilterListGet

Syntax

C# prototype

```
int PositionerCorrectorFilterListGet(string PositionerName, out string CorrectorFilterList, out string errstring)
```

Python prototype

```
[CorrectorFilterList, errstring] PositionerCorrectorFilterListGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) CorrectorFilterList: CorrectorFilterList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorFilterListGet command which is used to Get the list of corrector's filters. Refer to the XPS Programmer's manual to get the command description.

PositionerWarningFollowingErrorSet

Syntax

C# prototype

```
int PositionerWarningFollowingErrorSet(string PositionerName, double WarningFollowingError, out string errstring)
```

Python prototype

```
[errstring] PositionerWarningFollowingErrorSet (PositionerName, WarningFollowingError)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) WarningFollowingError: WarningFollowingError

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerWarningFollowingErrorSet command which is used to Set positioner warning following error limit. Refer to the XPS Programmer's manual to get the command description.

PositionerWarningFollowingErrorGet

Syntax

C# prototype

```
int PositionerWarningFollowingErrorGet(string PositionerName, out double WarningFollowingError, out string errstring)
```

Python prototype

```
[WarningFollowingError, errstring] PositionerWarningFollowingErrorGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) WarningFollowingError: WarningFollowingError

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerWarningFollowingErrorGet command which is used to Get positioner warning following error limit. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPositionFilterSet

Syntax

C# prototype

int PositionerCompensationPositionFilterSet(string PositionerName, Int32 FilterNumber, double Frequency, double DampingFactor, out string errstring)

Python prototype

[errstring] PositionerCompensationPositionFilterSet (PositionerName, FilterNumber, Frequency, DampingFactor)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) FilterNumber: FilterNumber

(double) Frequency: Frequency

(double) DampingFactor: DampingFactor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPositionFilterSet command which is used to Set and update position filter parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPositionFilterGet

Syntax

C# prototype

```
int PositionerCompensationPositionFilterGet(string PositionerName, Int32 FilterNumber, out double Frequency, out double DampingFactor, out string errstring)
```

Python prototype

```
[Frequency, DampingFactor, errstring] PositionerCompensationPositionFilterGet (PositionerName, FilterNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) FilterNumber: FilterNumber

Output parameters

(double) Frequency: Frequency

(double) DampingFactor: DampingFactor

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPositionFilterGet command which is used to Get position filter parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPlantFeedForwardDelaySet

Syntax

C# prototype

```
int PositionerCorrectorPlantFeedForwardDelaySet(string PositionerName, Int32 CorrectorISRPeriodNumber, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPlantFeedForwardDelaySet (PositionerName, CorrectorISRPeriodNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) CorrectorISRPeriodNumber: CorrectorISRPeriodNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPlantFeedForwardDelaySet command which is used to Set and update the fifo depth of Plant FeedForward Delay. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPlantFeedForwardDelayGet

Syntax

C# prototype

```
int PositionerCorrectorPlantFeedForwardDelayGet(string PositionerName, out Int32 CorrectorISRPeriodNumber, out string errstring)
```

Python prototype

```
[CorrectorISRPeriodNumber, errstring] PositionerCorrectorPlantFeedForwardDelayGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) CorrectorISRPeriodNumber: CorrectorISRPeriodNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPlantFeedForwardDelayGet command which is used to Get the fifo depth of Plant FeedForward Delay. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDisturbanceDisable

Syntax

C# prototype

```
int PositionerCompensationDisturbanceDisable(string PositionerName, string DisabledDirection, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationDisturbanceDisable (PositionerName, DisabledDirection)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) DisabledDirection: DisabledDirection

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDisturbanceDisable command which is used to Disable disturbance compensation for selected direction . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDisturbanceEnable

Syntax

C# prototype

```
int PositionerCompensationDisturbanceEnable(string PositionerName, string EnabledDirection, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationDisturbanceEnable (PositionerName, EnabledDirection)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) EnabledDirection: EnabledDirection

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDisturbanceEnable command which is used to Enable disturbance compensation for selected direction . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDisturbanceFileLoad

Syntax

C# prototype

```
int PositionerCompensationDisturbanceFileLoad(string PositionerName, string Direction, string FileName, out
string errstring)
```

Python prototype

```
[errstring] PositionerCompensationDisturbanceFileLoad (PositionerName, Direction, FileName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) Direction: Direction

(string) FileName: FileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDisturbanceFileLoad command which is used to Load file to compensate disturbance in requested direction . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDisturbanceStatusGet

Syntax

C# prototype

```
int PositionerCompensationDisturbanceStatusGet(string PositionerName, out Int32
PositiveCompensationEnabledStatus, out Int32 NegativeCompensationEnabledStatus, out string errstring)
```

Python prototype

```
[PositiveCompensationEnabledStatus, NegativeCompensationEnabledStatus, errstring]
PositionerCompensationDisturbanceStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) PositiveCompensationEnabledStatus: PositiveCompensationEnabledStatus

(Int32_i) NegativeCompensationEnabledStatus: NegativeCompensationEnabledStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDisturbanceStatusGet command which is used to Get status of disturbance compensation in both directions . Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorAutoTuning

Syntax

C# prototype

```
int PositionerCorrectorAutoTuning(string PositionerName, Int32 TuningMode, out double KP, out double KI, out double KD, out string errstring)
```

Python prototype

```
[KP, KI, KD, errstring] PositionerCorrectorAutoTuning (PositionerName, TuningMode)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) TuningMode: TuningMode

Output parameters

(double) KP: KP

(double) KI: KI

(double) KD: KD

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorAutoTuning command which is used to Astrom and Hagglund based auto-tuning. Refer to the XPS Programmer's manual to get the command description.

PositionerAccelerationAutoScaling

Syntax

C# prototype

```
int PositionerAccelerationAutoScaling(string PositionerName, out double Scaling, out string errstring)
```

Python prototype

```
[Scaling, errstring] PositionerAccelerationAutoScaling (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Scaling: Scaling

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerAccelerationAutoScaling command which is used to Astrom and Hagglund based auto-scaling. Refer to the XPS Programmer's manual to get the command description.

PositionerExcitationSignalGet

Syntax

C# prototype

```
int PositionerExcitationSignalGet(string PositionerName, out Int32 Mode, out double Frequency, out double Amplitude, out double Time, out string errstring)
```

Python prototype

```
[Mode, Frequency, Amplitude, Time, errstring] PositionerExcitationSignalGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) Mode: Mode

(double) Frequency: Frequency

(double) Amplitude: Amplitude

(double) Time: Time

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerExcitationSignalGet command which is used to Get excitation signal parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerExcitationSignalSet

Syntax

C# prototype

```
int PositionerExcitationSignalSet(string PositionerName, Int32 Mode, double Frequency, double Amplitude, double Time, out string errstring)
```

Python prototype

```
[errstring] PositionerExcitationSignalSet (PositionerName, Mode, Frequency, Amplitude, Time)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) Mode: Mode

(double) Frequency: Frequency

(double) Amplitude: Amplitude

(double) Time: Time

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerExcitationSignalSet command which is used to Set excitation signal mode. Refer to the XPS Programmer's manual to get the command description.

PositionerExcitationSignalCorrectorOutSet

Syntax

C# prototype

```
int PositionerExcitationSignalCorrectorOutSet(string PositionerName, Int32 Mode, double Frequency, double Amplitude, double Time, out string errstring)
```

Python prototype

```
[errstring] PositionerExcitationSignalCorrectorOutSet (PositionerName, Mode, Frequency, Amplitude, Time)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) Mode: Mode

(double) Frequency: Frequency

(double) Amplitude: Amplitude

(double) Time: Time

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerExcitationSignalCorrectorOutSet command which is used to Set excitation signal mode at corrector out. Refer to the XPS Programmer's manual to get the command description.

PositionerCurrentVelocityAccelerationFiltersSet

Syntax

C# prototype

```
int PositionerCurrentVelocityAccelerationFiltersSet(string PositionerName, double CurrentVelocityCutOffFrequency,
double CurrentAccelerationCutOffFrequency, out string errstring)
```

Python prototype

```
[errstring] PositionerCurrentVelocityAccelerationFiltersSet (PositionerName, CurrentVelocityCutOffFrequency,
CurrentAccelerationCutOffFrequency)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) CurrentVelocityCutOffFrequency: CurrentVelocityCutOffFrequency

(double) CurrentAccelerationCutOffFrequency: CurrentAccelerationCutOffFrequency

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCurrentVelocityAccelerationFiltersSet command which is used to Set current velocity and acceleration cut off frequencies. Refer to the XPS Programmer's manual to get the command description.

PositionerCurrentVelocityAccelerationFiltersGet

Syntax

C# prototype

```
int PositionerCurrentVelocityAccelerationFiltersGet(string PositionerName, out double
CurrentVelocityCutOffFrequency, out double CurrentAccelerationCutOffFrequency, out string errstring)
```

Python prototype

```
[CurrentVelocityCutOffFrequency, CurrentAccelerationCutOffFrequency, errstring]
PositionerCurrentVelocityAccelerationFiltersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CurrentVelocityCutOffFrequency: CurrentVelocityCutOffFrequency

(double) CurrentAccelerationCutOffFrequency: CurrentAccelerationCutOffFrequency

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCurrentVelocityAccelerationFiltersGet command which is used to Get current velocity and acceleration cut off frequencies. Refer to the XPS Programmer's manual to get the command description.

PositionerEncoderAmplitudeValuesGet

Syntax

C# prototype

```
int PositionerEncoderAmplitudeValuesGet(string PositionerName, out double CalibrationSinusAmplitude, out double CurrentSinusAmplitude, out double CalibrationCosinusAmplitude, out double CurrentCosinusAmplitude, out string errstring)
```

Python prototype

```
[CalibrationSinusAmplitude, CurrentSinusAmplitude, CalibrationCosinusAmplitude, CurrentCosinusAmplitude, errstring] PositionerEncoderAmplitudeValuesGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CalibrationSinusAmplitude: CalibrationSinusAmplitude

(double) CurrentSinusAmplitude: CurrentSinusAmplitude

(double) CalibrationCosinusAmplitude: CalibrationCosinusAmplitude

(double) CurrentCosinusAmplitude: CurrentCosinusAmplitude

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerEncoderAmplitudeValuesGet command which is used to Read analog interpolated encoder amplitude values. Refer to the XPS Programmer's manual to get the command description.

PositionerEncoderCalibrationParametersGet

Syntax

C# prototype

```
int PositionerEncoderCalibrationParametersGet(string PositionerName, out double SinusOffset, out double CosinusOffset, out double DifferentialGain, out double PhaseCompensation, out string errstring)
```

Python prototype

```
[SinusOffset, CosinusOffset, DifferentialGain, PhaseCompensation, errstring]  
PositionerEncoderCalibrationParametersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) SinusOffset: SinusOffset

(double) CosinusOffset: CosinusOffset

(double) DifferentialGain: DifferentialGain

(double) PhaseCompensation: PhaseCompensation

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerEncoderCalibrationParametersGet command which is used to Read analog interpolated encoder calibration parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerEncoderLinearCorrectionGet

Syntax

C# prototype

```
int PositionerEncoderLinearCorrectionGet(string PositionerName, out double LinearEncoderCorrection, out string errstring)
```

Python prototype

```
[LinearEncoderCorrection, errstring] PositionerEncoderLinearCorrectionGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) LinearEncoderCorrection: LinearEncoderCorrection

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerEncoderLinearCorrectionGet command which is used to Get linear encoder correction in ppm. Refer to the XPS Programmer's manual to get the command description.

PositionerEncoderLinearCorrectionSet

Syntax

C# prototype

```
int PositionerEncoderLinearCorrectionSet(string PositionerName, double LinearEncoderCorrection, out string errstring)
```

Python prototype

```
[errstring] PositionerEncoderLinearCorrectionSet (PositionerName, LinearEncoderCorrection)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) LinearEncoderCorrection: LinearEncoderCorrection

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerEncoderLinearCorrectionSet command which is used to Set linear encoder correction in ppm. Refer to the XPS Programmer's manual to get the command description.

PositionerRawEncoderPositionGet

Syntax

C# prototype

```
int PositionerRawEncoderPositionGet(string PositionerName, double UserEncoderPosition, out double RawEncoderPosition, out string errstring)
```

Python prototype

```
[RawEncoderPosition, errstring] PositionerRawEncoderPositionGet (PositionerName, UserEncoderPosition)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) UserEncoderPosition: UserEncoderPosition

Output parameters

(double) RawEncoderPosition: RawEncoderPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerRawEncoderPositionGet command which is used to Get the raw encoder position. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDFFAccelerationSet

Syntax

C# prototype

```
int PositionerCorrectorPIDFFAccelerationSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI,
double KD, double KS, double IntegrationTime, double DerivativeFilterCutOffFrequency, double GKP, double GKI,
double GKD, double KForm, double KFeedForwardAcceleration, double KFeedForwardJerk, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPIDFFAccelerationSet (PositionerName, ClosedLoopStatus, KP, KI, KD, KS,
IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, KFeedForwardAcceleration,
KFeedForwardJerk)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) KFeedForwardAcceleration: KFeedForwardAcceleration

(double) KFeedForwardJerk: KFeedForwardJerk

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDFFAccelerationSet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDFFAccelerationGet

Syntax

C# prototype

```
int PositionerCorrectorPIDFFAccelerationGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double KD, out double KS, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double GKP, out double GKI, out double GKD, out double KForm, out double KFeedForwardAcceleration, out double KFeedForwardJerk, out string errstring)
```

Python prototype

```
[ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, KFeedForwardAcceleration, KFeedForwardJerk, errstring] PositionerCorrectorPIDFFAccelerationGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) KFeedForwardAcceleration: KFeedForwardAcceleration

(double) KFeedForwardJerk: KFeedForwardJerk

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDFFAccelerationGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDAccelerationFilterSet

Syntax

C# prototype

```
int PositionerCorrectorPIDAccelerationFilterSet(string PositionerName, bool FilterControlStatus, double KD, double  
DerivativeFilterCutOffFrequency, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPIDAccelerationFilterSet (PositionerName, FilterControlStatus, KD,  
DerivativeFilterCutOffFrequency)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) FilterControlStatus: FilterControlStatus

(double) KD: KD

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDAccelerationFilterSet command which is used to Update PID acceleration corrector filter parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDAccelerationFilterGet

Syntax

C# prototype

```
int PositionerCorrectorPIDAccelerationFilterGet(string PositionerName, out bool FilterControlStatus, out double KD, out double DerivativeFilterCutOffFrequency, out string errstring)
```

Python prototype

```
[FilterControlStatus, KD, DerivativeFilterCutOffFrequency, errstring] PositionerCorrectorPIDAccelerationFilterGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) FilterControlStatus: FilterControlStatus

(double) KD: KD

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDAccelerationFilterGet command which is used to Read PID acceleration corrector filter parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorDamperFilterSet

Syntax

C# prototype

```
int PositionerCorrectorDamperFilterSet(string PositionerName, double CutOffFrequency, double DampingFactor, double Gain, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorDamperFilterSet (PositionerName, CutOffFrequency, DampingFactor, Gain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) CutOffFrequency: CutOffFrequency

(double) DampingFactor: DampingFactor

(double) Gain: Gain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorDamperFilterSet command which is used to Update corrector damper filter parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorDamperFilterGet

Syntax

C# prototype

```
int PositionerCorrectorDamperFilterGet(string PositionerName, out double CutOffFrequency, out double DampingFactor, out double Gain, out string errstring)
```

Python prototype

```
[CutOffFrequency, DampingFactor, Gain, errstring] PositionerCorrectorDamperFilterGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CutOffFrequency: CutOffFrequency

(double) DampingFactor: DampingFactor

(double) Gain: Gain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorDamperFilterGet command which is used to Read corrector damper filter parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPostFFSet

Syntax

C# prototype

```
int PositionerCorrectorPostFFSet(string PositionerName, double PostKFeedForwardAcceleration, double PostKFeedForwardJerk, double PostKFeedForwardSlope, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPostFFSet (PositionerName, PostKFeedForwardAcceleration, PostKFeedForwardJerk, PostKFeedForwardSlope)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PostKFeedForwardAcceleration: PostKFeedForwardAcceleration

(double) PostKFeedForwardJerk: PostKFeedForwardJerk

(double) PostKFeedForwardSlope: PostKFeedForwardSlope

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPostFFSet command which is used to Update post feedforward parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPostFFGet

Syntax

C# prototype

```
int PositionerCorrectorPostFFGet(string PositionerName, out double PostKFeedForwardAcceleration, out double PostKFeedForwardJerk, out double PostKFeedForwardSlope, out string errstring)
```

Python prototype

```
[PostKFeedForwardAcceleration, PostKFeedForwardJerk, PostKFeedForwardSlope, errstring]  
PositionerCorrectorPostFFGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PostKFeedForwardAcceleration: PostKFeedForwardAcceleration

(double) PostKFeedForwardJerk: PostKFeedForwardJerk

(double) PostKFeedForwardSlope: PostKFeedForwardSlope

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPostFFGet command which is used to Read post feedforward parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorExcitationSignalGainSet

Syntax

C# prototype

```
int PositionerCorrectorExcitationSignalGainSet(string PositionerName, double ExcitationSignalGain, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorExcitationSignalGainSet (PositionerName, ExcitationSignalGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) ExcitationSignalGain: ExcitationSignalGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorExcitationSignalGainSet command which is used to Update the excitation signal gain [-1:1]. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorExcitationSignalGainGet

Syntax

C# prototype

```
int PositionerCorrectorExcitationSignalGainGet(string PositionerName, out double ExcitationSignalGain, out string errstring)
```

Python prototype

```
[ExcitationSignalGain, errstring] PositionerCorrectorExcitationSignalGainGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) ExcitationSignalGain: ExcitationSignalGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorExcitationSignalGainGet command which is used to Read excitation signal gain. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDBaseSet

Syntax

C# prototype

```
int PositionerCorrectorPIDBaseSet(string PositionerName, double MovingMass, double StaticMass, double Viscosity, double Stiffness, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPIDBaseSet (PositionerName, MovingMass, StaticMass, Viscosity, Stiffness)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) MovingMass: MovingMass

(double) StaticMass: StaticMass

(double) Viscosity: Viscosity

(double) Stiffness: Stiffness

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDBaseSet command which is used to Update PIDBase parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDBaseGet

Syntax

C# prototype

```
int PositionerCorrectorPIDBaseGet(string PositionerName, out double MovingMass, out double StaticMass, out double Viscosity, out double Stiffness, out string errstring)
```

Python prototype

```
[MovingMass, StaticMass, Viscosity, Stiffness, errstring] PositionerCorrectorPIDBaseGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MovingMass: MovingMass

(double) StaticMass: StaticMass

(double) Viscosity: Viscosity

(double) Stiffness: Stiffness

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDBaseGet command which is used to Read PIDBase parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareScanAccelerationLimitGet

Syntax

C# prototype

```
int PositionerPositionCompareScanAccelerationLimitGet(string PositionerName, out double ScanAccelerationLimit,
out string errstring)
```

Python prototype

```
[ScanAccelerationLimit, errstring] PositionerPositionCompareScanAccelerationLimitGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) ScanAccelerationLimit: ScanAccelerationLimit

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareScanAccelerationLimitGet command which is used to Get position compare scan acceleration limit. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareScanAccelerationLimitSet

Syntax

C# prototype

```
int PositionerPositionCompareScanAccelerationLimitSet(string PositionerName, double ScanAccelerationLimit, out
string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareScanAccelerationLimitSet (PositionerName, ScanAccelerationLimit)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) ScanAccelerationLimit: ScanAccelerationLimit

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareScanAccelerationLimitSet command which is used to Set position compare scan acceleration limit. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDDualFFVoltageSet

Syntax

C# prototype

```
int PositionerCorrectorPIDDualFFVoltageSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI,
double KD, double KS, double IntegrationTime, double DerivativeFilterCutOffFrequency, double GKP, double GKI,
double GKD, double KForm, double KFeedForwardVelocity, double KFeedForwardAcceleration, double Friction, out
string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPIDDualFFVoltageSet (PositionerName, ClosedLoopStatus, KP, KI, KD, KS,
IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, KFeedForwardVelocity,
KFeedForwardAcceleration, Friction)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) KFeedForwardVelocity: KFeedForwardVelocity

(double) KFeedForwardAcceleration: KFeedForwardAcceleration

(double) Friction: Friction

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDDualFFVoltageSet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDDualFFVoltageGet

Syntax

C# prototype

```
int PositionerCorrectorPIDDualFFVoltageGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double KD, out double KS, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double GKP, out double GKI, out double GKD, out double KForm, out double KFeedForwardVelocity, out double KFeedForwardAcceleration, out double Friction, out string errstring)
```

Python prototype

```
[ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, KFeedForwardVelocity, KFeedForwardAcceleration, Friction, errstring] PositionerCorrectorPIDDualFFVoltageGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) KFeedForwardVelocity: KFeedForwardVelocity

(double) KFeedForwardAcceleration: KFeedForwardAcceleration

(double) Friction: Friction

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDDualFFVoltageGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDFFVelocitySet

Syntax

C# prototype

```
int PositionerCorrectorPIDFFVelocitySet(string PositionerName, bool ClosedLoopStatus, double KP, double KI,
double KD, double KS, double IntegrationTime, double DerivativeFilterCutOffFrequency, double GKP, double GKI,
double GKD, double KForm, double KFeedForwardVelocity, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPIDFFVelocitySet (PositionerName, ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime,
DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, KFeedForwardVelocity)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) KFeedForwardVelocity: KFeedForwardVelocity

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDFFVelocitySet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIDFFVelocityGet

Syntax

C# prototype

```
int PositionerCorrectorPIDFFVelocityGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double KD, out double KS, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double GKP, out double GKI, out double GKD, out double KForm, out double KFeedForwardVelocity, out string errstring)
```

Python prototype

```
[ClosedLoopStatus, KP, KI, KD, KS, IntegrationTime, DerivativeFilterCutOffFrequency, GKP, GKI, GKD, KForm, KFeedForwardVelocity, errstring] PositionerCorrectorPIDFFVelocityGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) KS: KS

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) GKP: GKP

(double) GKI: GKI

(double) GKD: GKD

(double) KForm: KForm

(double) KFeedForwardVelocity: KFeedForwardVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIDFFVelocityGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIPositionSet

Syntax

C# prototype

```
int PositionerCorrectorPIPositionSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI, double IntegrationTime, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorPIPositionSet (PositionerName, ClosedLoopStatus, KP, KI, IntegrationTime)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) IntegrationTime: IntegrationTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIPositionSet command which is used to Update corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorPIPositionGet

Syntax

C# prototype

```
int PositionerCorrectorPIPositionGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI, out double IntegrationTime, out string errstring)
```

Python prototype

```
[ClosedLoopStatus, KP, KI, IntegrationTime, errstring] PositionerCorrectorPIPositionGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) IntegrationTime: IntegrationTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorPIPositionGet command which is used to Read corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorDualSet

Syntax

C# prototype

```
int PositionerCorrectorDualSet(string PositionerName, bool ClosedLoopStatus, double KP, double KI, double KD,
double IntegrationTime, double DerivativeFilterCutOffFrequency, double KFeedForwardAcceleration, double
KFeedForwardJerk, double AntiWindUpTime, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorDualSet (PositionerName, ClosedLoopStatus, KP, KI, KD, IntegrationTime,
DerivativeFilterCutOffFrequency, KFeedForwardAcceleration, KFeedForwardJerk, AntiWindUpTime)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) KFeedForwardAcceleration: KFeedForwardAcceleration

(double) KFeedForwardJerk: KFeedForwardJerk

(double) AntiWindUpTime: AntiWindUpTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorDualSet command which is used to Update dual corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorDualGet

Syntax

C# prototype

```
int PositionerCorrectorDualGet(string PositionerName, out bool ClosedLoopStatus, out double KP, out double KI,
out double KD, out double IntegrationTime, out double DerivativeFilterCutOffFrequency, out double
KFeedForwardAcceleration, out double KFeedForwardJerk, out double AntiWindUpTime, out string errstring)
```

Python prototype

```
[ClosedLoopStatus, KP, KI, KD, IntegrationTime, DerivativeFilterCutOffFrequency, KFeedForwardAcceleration,
KFeedForwardJerk, AntiWindUpTime, errstring] PositionerCorrectorDualGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(bool) ClosedLoopStatus: ClosedLoopStatus

(double) KP: KP

(double) KI: KI

(double) KD: KD

(double) IntegrationTime: IntegrationTime

(double) DerivativeFilterCutOffFrequency: DerivativeFilterCutOffFrequency

(double) KFeedForwardAcceleration: KFeedForwardAcceleration

(double) KFeedForwardJerk: KFeedForwardJerk

(double) AntiWindUpTime: AntiWindUpTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorDualGet command which is used to Read dual corrector parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDualLoopNotchFilterSet

Syntax

C# prototype

```
int PositionerCompensationDualLoopNotchFilterSet(string PositionerName, Int32 NotchNumber, double
NotchFrequency, double NotchBandwidth, double NotchGain, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationDualLoopNotchFilterSet (PositionerName, NotchNumber, NotchFrequency,
NotchBandwidth, NotchGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchNumber: NotchNumber

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDualLoopNotchFilterSet command which is used to Update Notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDualLoopNotchFilterGet

Syntax

C# prototype

```
int PositionerCompensationDualLoopNotchFilterGet(string PositionerName, Int32 NotchNumber, out double NotchFrequency, out double NotchBandwidth, out double NotchGain, out string errstring)
```

Python prototype

```
[NotchFrequency, NotchBandwidth, NotchGain, errstring] PositionerCompensationDualLoopNotchFilterGet(PositionerName, NotchNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchNumber: NotchNumber

Output parameters

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDualLoopNotchFilterGet command which is used to Read Notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDualLoopPhaseCorrectionFilterSet

Syntax

C# prototype

```
int PositionerCompensationDualLoopPhaseCorrectionFilterSet(string PositionerName, Int32
PhaseCorrectionFilterNumber, double PhaseCorrectionFn, double PhaseCorrectionFd, double PhaseCorrectionGain,
out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationDualLoopPhaseCorrectionFilterSet (PositionerName,
PhaseCorrectionFilterNumber, PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDualLoopPhaseCorrectionFilterSet command which is used to Update one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationDualLoopPhaseCorrectionFilterGet

Syntax

C# prototype

```
int PositionerCompensationDualLoopPhaseCorrectionFilterGet(string PositionerName, Int32
PhaseCorrectionFilterNumber, out double PhaseCorrectionFn, out double PhaseCorrectionFd, out double
PhaseCorrectionGain, out string errstring)
```

Python prototype

```
[PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain, errstring]
PositionerCompensationDualLoopPhaseCorrectionFilterGet (PositionerName, PhaseCorrectionFilterNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

Output parameters

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationDualLoopPhaseCorrectionFilterGet command which is used to Read one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerDriverFiltersGet

Syntax

C# prototype

```
int PositionerDriverFiltersGet(string PositionerName, out double KI, out double NotchFrequency, out double NotchBandwidth, out double NotchGain, out double LowpassFrequency, out string errstring)
```

Python prototype

```
[KI, NotchFrequency, NotchBandwidth, NotchGain, LowpassFrequency, errstring] PositionerDriverFiltersGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) KI: KI

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(double) LowpassFrequency: LowpassFrequency

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerDriverFiltersGet command which is used to Get driver filters parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerDriverFiltersSet

Syntax

C# prototype

```
int PositionerDriverFiltersSet(string PositionerName, double KI, double NotchFrequency, double NotchBandwidth,
double NotchGain, double LowpassFrequency, out string errstring)
```

Python prototype

```
[errstring] PositionerDriverFiltersSet (PositionerName, KI, NotchFrequency, NotchBandwidth, NotchGain,
LowpassFrequency)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) KI: KI

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(double) LowpassFrequency: LowpassFrequency

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerDriverFiltersSet command which is used to Set driver filters parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerDriverPositionOffsetsGet

Syntax

C# prototype

```
int PositionerDriverPositionOffsetsGet(string PositionerName, out double StagePositionOffset, out double
GagePositionOffset, out string errstring)
```

Python prototype

```
[StagePositionOffset, GagePositionOffset, errstring] PositionerDriverPositionOffsetsGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) StagePositionOffset: StagePositionOffset

(double) GagePositionOffset: GagePositionOffset

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerDriverPositionOffsetsGet command which is used to Get driver stage and gage position offset. Refer to the XPS Programmer's manual to get the command description.

PositionerPreCorrectorExcitationSignalGet

Syntax

C# prototype

```
int PositionerPreCorrectorExcitationSignalGet(string PositionerName, out double Frequency, out double Amplitude,
out double Time, out string errstring)
```

Python prototype

```
[Frequency, Amplitude, Time, errstring] PositionerPreCorrectorExcitationSignalGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Frequency: Frequency

(double) Amplitude: Amplitude

(double) Time: Time

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPreCorrectorExcitationSignalGet command which is used to Get pre-corrector excitation signal mode. Refer to the XPS Programmer's manual to get the command description.

PositionerPreCorrectorExcitationSignalSet

Syntax

C# prototype

```
int PositionerPreCorrectorExcitationSignalSet(string PositionerName, double Frequency, double Amplitude, double Time, out string errstring)
```

Python prototype

```
[errstring] PositionerPreCorrectorExcitationSignalSet (PositionerName, Frequency, Amplitude, Time)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Frequency: Frequency

(double) Amplitude: Amplitude

(double) Time: Time

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPreCorrectorExcitationSignalSet command which is used to Set pre-corrector excitation signal mode. Refer to the XPS Programmer's manual to get the command description.

PositionerBacklashSet

Syntax

C# prototype

```
int PositionerBacklashSet(string PositionerName, double BacklashValue, out string errstring)
```

Python prototype

```
[errstring] PositionerBacklashSet (PositionerName, BacklashValue)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) BacklashValue: BacklashValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerBacklashSet command which is used to Set backlash value. Refer to the XPS Programmer's manual to get the command description.

PositionerBacklashGet

Syntax

C# prototype

```
int PositionerBacklashGet(string PositionerName, out double BacklashValue, out string BacklashStatus, out string errstring)
```

Python prototype

```
[BacklashValue, BacklashStatus, errstring] PositionerBacklashGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) BacklashValue: BacklashValue

(string) BacklashStatus: BacklashStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerBacklashGet command which is used to Read backlash value and status. Refer to the XPS Programmer's manual to get the command description.

PositionerBacklashEnable

Syntax

C# prototype

```
int PositionerBacklashEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerBacklashEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerBacklashEnable command which is used to Enable the backlash. Refer to the XPS Programmer's manual to get the command description.

PositionerBacklashDisable

Syntax

C# prototype

```
int PositionerBacklashDisable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerBacklashDisable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerBacklashDisable command which is used to Disable the backlash. Refer to the XPS Programmer's manual to get the command description.

PositionerMotionDoneGet

Syntax

C# prototype

```
int PositionerMotionDoneGet(string PositionerName, out double PositionWindow, out double VelocityWindow, out double CheckingTime, out double MeanPeriod, out double TimeOut, out string errstring)
```

Python prototype

```
[PositionWindow, VelocityWindow, CheckingTime, MeanPeriod, TimeOut, errstring] PositionerMotionDoneGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PositionWindow: PositionWindow

(double) VelocityWindow: VelocityWindow

(double) CheckingTime: CheckingTime

(double) MeanPeriod: MeanPeriod

(double) TimeOut: TimeOut

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotionDoneGet command which is used to Read motion done parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerMotionDoneSet

Syntax

C# prototype

```
int PositionerMotionDoneSet(string PositionerName, double PositionWindow, double VelocityWindow, double CheckingTime, double MeanPeriod, double TimeOut, out string errstring)
```

Python prototype

```
[errstring] PositionerMotionDoneSet (PositionerName, PositionWindow, VelocityWindow, CheckingTime, MeanPeriod, TimeOut)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PositionWindow: PositionWindow

(double) VelocityWindow: VelocityWindow

(double) CheckingTime: CheckingTime

(double) MeanPeriod: MeanPeriod

(double) TimeOut: TimeOut

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotionDoneSet command which is used to Update motion done parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerHardInterpolatorFactorGet

Syntax

C# prototype

```
int PositionerHardInterpolatorFactorGet(string PositionerName, out Int32 InterpolationFactor, out string errstring)
```

Python prototype

```
[InterpolationFactor, errstring] PositionerHardInterpolatorFactorGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) InterpolationFactor: InterpolationFactor

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerHardInterpolatorFactorGet command which is used to Get hard interpolator parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerHardInterpolatorFactorSet

Syntax

C# prototype

```
int PositionerHardInterpolatorFactorSet(string PositionerName, Int32 InterpolationFactor, out string errstring)
```

Python prototype

```
[errstring] PositionerHardInterpolatorFactorSet (PositionerName, InterpolationFactor)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) InterpolationFactor: InterpolationFactor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerHardInterpolatorFactorSet command which is used to Set hard interpolator parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerHardInterpolatorPositionGet

Syntax

C# prototype

```
int PositionerHardInterpolatorPositionGet(string PositionerName, out double Position, out string errstring)
```

Python prototype

```
[Position, errstring] PositionerHardInterpolatorPositionGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Position: Position

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerHardInterpolatorPositionGet command which is used to Read external latch position. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareGet

Syntax

C# prototype

```
int PositionerPositionCompareGet(string PositionerName, out double MinimumPosition, out double MaximumPosition, out double PositionStep, out bool EnableState, out string errstring)
```

Python prototype

```
[MinimumPosition, MaximumPosition, PositionStep, EnableState, errstring] PositionerPositionCompareGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) PositionStep: PositionStep

(bool) EnableState: EnableState

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareGet command which is used to Read position compare parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareSet

Syntax

C# prototype

```
int PositionerPositionCompareSet(string PositionerName, double MinimumPosition, double MaximumPosition,
double PositionStep, out string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareSet (PositionerName, MinimumPosition, MaximumPosition, PositionStep)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) PositionStep: PositionStep

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareSet command which is used to Set position compare parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareEnable

Syntax

C# prototype

```
int PositionerPositionCompareEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareEnable command which is used to Enable position compare. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareDisable

Syntax

C# prototype

```
int PositionerPositionCompareDisable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareDisable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareDisable command which is used to Disable position compare. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareAquadBAAlwaysEnable

Syntax

C# prototype

```
int PositionerPositionCompareAquadBAAlwaysEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareAquadBAAlwaysEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareAquadBAAlwaysEnable command which is used to Enable AquadB signal in always mode. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareAquadBWindowedGet

Syntax

C# prototype

```
int PositionerPositionCompareAquadBWindowedGet(string PositionerName, out double MinimumPosition, out double MaximumPosition, out bool EnableState, out string errstring)
```

Python prototype

```
[MinimumPosition, MaximumPosition, EnableState, errstring] PositionerPositionCompareAquadBWindowedGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(bool) EnableState: EnableState

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareAquadBWindowedGet command which is used to Read position compare AquadB windowed parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareAquadBWindowedSet

Syntax

C# prototype

```
int PositionerPositionCompareAquadBWindowedSet(string PositionerName, double MinimumPosition, double
MaximumPosition, out string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareAquadBWindowedSet (PositionerName, MinimumPosition, MaximumPosition)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareAquadBWindowedSet command which is used to Set position compare AquadB windowed parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerTimeFlasherGet

Syntax

C# prototype

```
int PositionerTimeFlasherGet(string PositionerName, out double MinimumPosition, out double MaximumPosition, out double PositionStep, out bool EnableState, out string errstring)
```

Python prototype

```
[MinimumPosition, MaximumPosition, PositionStep, EnableState, errstring] PositionerTimeFlasherGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) PositionStep: PositionStep

(bool) EnableState: EnableState

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerTimeFlasherGet command which is used to Read time flasher parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerTimeFlasherSet

Syntax

C# prototype

```
int PositionerTimeFlasherSet(string PositionerName, double MinimumPosition, double MaximumPosition, double
TimeInterval, out string errstring)
```

Python prototype

```
[errstring] PositionerTimeFlasherSet (PositionerName, MinimumPosition, MaximumPosition, TimeInterval)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerTimeFlasherSet command which is used to Set time flasher parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerTimeFlasherEnable

Syntax

C# prototype

```
int PositionerTimeFlasherEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerTimeFlasherEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerTimeFlasherEnable command which is used to Enable time flasher. Refer to the XPS Programmer's manual to get the command description.

PositionerTimeFlasherDisable

Syntax

C# prototype

```
int PositionerTimeFlasherDisable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerTimeFlasherDisable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerTimeFlasherDisable command which is used to Disable time flasher. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionComparePulseParametersGet

Syntax

C# prototype

```
int PositionerPositionComparePulseParametersGet(string PositionerName, out double PCOPulseWidth, out double EncoderSettlingTimeISAOnly, out string errstring)
```

Python prototype

```
[PCOPulseWidth, EncoderSettlingTimeISAOnly, errstring] PositionerPositionComparePulseParametersGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PCOPulseWidth: PCOPulseWidth

(double) EncoderSettlingTimeISAOnly: EncoderSettlingTimeISAOnly

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionComparePulseParametersGet command which is used to Get position compare PCO pulse parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionComparePulseParametersSet

Syntax

C# prototype

```
int PositionerPositionComparePulseParametersSet(string PositionerName, double PCOPulseWidth, double EncoderSettlingTimeISAOnly, out string errstring)
```

Python prototype

```
[errstring] PositionerPositionComparePulseParametersSet (PositionerName, PCOPulseWidth, EncoderSettlingTimeISAOnly)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PCOPulseWidth: PCOPulseWidth

(double) EncoderSettlingTimeISAOnly: EncoderSettlingTimeISAOnly

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionComparePulseParametersSet command which is used to Set position compare PCO pulse parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareAquadBPrescalerSet

Syntax

C# prototype

```
int PositionerPositionCompareAquadBPrescalerSet(string PositionerName, double PCOInterpolationFactor, out
string errstring)
```

Python prototype

```
[errstring] PositionerPositionCompareAquadBPrescalerSet (PositionerName, PCOInterpolationFactor)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PCOInterpolationFactor: PCOInterpolationFactor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareAquadBPrescalerSet command which is used to Set PCO AquadB Interpolation Factor . Refer to the XPS Programmer's manual to get the command description.

PositionerPositionCompareAquadBPrescalerGet

Syntax

C# prototype

```
int PositionerPositionCompareAquadBPrescalerGet(string PositionerName, out double PCOInterpolationFactor, out string errstring)
```

Python prototype

```
[PCOInterpolationFactor, errstring] PositionerPositionCompareAquadBPrescalerGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PCOInterpolationFactor: PCOInterpolationFactor

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerPositionCompareAquadBPrescalerGet command which is used to Get current PCO Interpolation Factor . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOAbort

Syntax

C# prototype

```
int PositionerCompensatedPCOAbort(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOAbort (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOAbort command which is used to Abort CIE08 compensated PCO mode. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOCurrentStatusGet

Syntax

C# prototype

```
int PositionerCompensatedPCOCurrentStatusGet(string PositionerName, out Int32 Status, out string errstring)
```

Python prototype

```
[Status, errstring] PositionerCompensatedPCOCurrentStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOCurrentStatusGet command which is used to Get current status of CIE08 compensated PCO mode. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOEnable

Syntax

C# prototype

```
int PositionerCompensatedPCOEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOEnable command which is used to Enable CIE08 compensated PCO mode execution. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOFromFile

Syntax

C# prototype

```
int PositionerCompensatedPCOFromFile(string PositionerName, string DataFileName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOFromFile (PositionerName, DataFileName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) DataFileName: DataFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOFromFile command which is used to Load file to CIE08 compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOLoadToMemory

Syntax

C# prototype

```
int PositionerCompensatedPCOLoadToMemory(string PositionerName, string DataLines, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOLoadToMemory (PositionerName, DataLines)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) DataLines: DataLines

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOLoadToMemory command which is used to Load data lines to CIE08 compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOMemoryReset

Syntax

C# prototype

```
int PositionerCompensatedPCOMemoryReset(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOMemoryReset (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOMemoryReset command which is used to Reset CIE08 compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOPrepare

Syntax

C# prototype

```
int PositionerCompensatedPCOPrepare(string PositionerName, Int32 ScanDirection, double[] StartPosition, Int32 nbltems, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOPrepare (PositionerName, ScanDirection, StartPosition, nbltems)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) ScanDirection: ScanDirection

(double[]) StartPosition: StartPosition

(Int32) nbltems: nbltems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOPrepare command which is used to Prepare data for CIE08 compensated PCO mode. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedPCOSet

Syntax

C# prototype

```
int PositionerCompensatedPCOSet(string PositionerName, double Start, double Stop, double Distance, double Width, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedPCOSet (PositionerName, Start, Stop, Distance, Width)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Start: Start

(double) Stop: Stop

(double) Distance: Distance

(double) Width: Width

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedPCOSet command which is used to Set data to CIE08 compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOAbort

Syntax

C# prototype

```
int PositionerCompensatedFastPCOAbort(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOAbort (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOAbort command which is used to Abort fast compensated PCO mode. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOCurrentStatusGet

Syntax

C# prototype

```
int PositionerCompensatedFastPCOCurrentStatusGet(string PositionerName, out Int32 Status, out string errstring)
```

Python prototype

```
[Status, errstring] PositionerCompensatedFastPCOCurrentStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(Int32_i) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOCurrentStatusGet command which is used to Get current status of fast compensated PCO mode. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOEnable

Syntax

C# prototype

```
int PositionerCompensatedFastPCOEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOEnable command which is used to Enable fast compensated PCO mode execution. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOFromFile

Syntax

C# prototype

```
int PositionerCompensatedFastPCOFromFile(string PositionerName, string DataFileName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOFromFile (PositionerName, DataFileName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) DataFileName: DataFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOFromFile command which is used to Load file to fast compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOLoadToMemory

Syntax

C# prototype

```
int PositionerCompensatedFastPCOLoadToMemory(string PositionerName, string DataLines, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOLoadToMemory (PositionerName, DataLines)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) DataLines: DataLines

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOLoadToMemory command which is used to Load data lines to fast compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOMemoryReset

Syntax

C# prototype

```
int PositionerCompensatedFastPCOMemoryReset(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOMemoryReset (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOMemoryReset command which is used to Reset fast compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOPrepare

Syntax

C# prototype

```
int PositionerCompensatedFastPCOPrepare(string PositionerName, Int32 ScanDirection, double[] StartPosition,
Int32 nbItems, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOPrepare (PositionerName, ScanDirection, StartPosition, nbItems)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) ScanDirection: ScanDirection

(double[]) StartPosition: StartPosition

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOPrepare command which is used to Prepare data for fast compensated PCO mode. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOSet

Syntax

C# prototype

```
int PositionerCompensatedFastPCOSet(string PositionerName, double Start, double Stop, double Step, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOSet (PositionerName, Start, Stop, Step)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Start: Start

(double) Stop: Stop

(double) Step: Step

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOSet command which is used to Set data to compensated PCO data buffer. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOPulseParametersGet

Syntax

C# prototype

```
int PositionerCompensatedFastPCOPulseParametersGet(string PositionerName, out double PulseWidth, out Int32 PulsePolarity, out bool PulseToggle, out string errstring)
```

Python prototype

```
[PulseWidth, PulsePolarity, PulseToggle, errstring] PositionerCompensatedFastPCOPulseParametersGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PulseWidth: PulseWidth

(Int32_i) PulsePolarity: PulsePolarity

(bool) PulseToggle: PulseToggle

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOPulseParametersGet command which is used to Get pulse configuration to compensated PCO. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensatedFastPCOPulseParametersSet

Syntax

C# prototype

```
int PositionerCompensatedFastPCOPulseParametersSet(string PositionerName, double PulseWidth, Int32
PulsePolarity, bool PulseToggle, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensatedFastPCOPulseParametersSet (PositionerName, PulseWidth, PulsePolarity,
PulseToggle)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PulseWidth: PulseWidth

(Int32) PulsePolarity: PulsePolarity

(bool) PulseToggle: PulseToggle

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensatedFastPCOPulseParametersSet command which is used to Set pulse configuration to compensated PCO. Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationEncoderNotchFilterSet

Syntax

C# prototype

```
int PositionerCompensationEncoderNotchFilterSet(string PositionerName, Int32 NotchNumber, double
NotchFrequency, double NotchBandwidth, double NotchGain, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationEncoderNotchFilterSet (PositionerName, NotchNumber, NotchFrequency,
NotchBandwidth, NotchGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchNumber: NotchNumber

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationEncoderNotchFilterSet command which is used to Update Notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationEncoderNotchFilterGet

Syntax

C# prototype

```
int PositionerCompensationEncoderNotchFilterGet(string PositionerName, Int32 NotchNumber, out double
NotchFrequency, out double NotchBandwidth, out double NotchGain, out string errstring)
```

Python prototype

```
[NotchFrequency, NotchBandwidth, NotchGain, errstring] PositionerCompensationEncoderNotchFilterGet
(PositionerName, NotchNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchNumber: NotchNumber

Output parameters

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationEncoderNotchFilterGet command which is used to Read Notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationNotchFilterSet

Syntax

C# prototype

```
int PositionerCompensationNotchFilterSet(string PositionerName, Int32 NotchNumber, double NotchFrequency, double NotchBandwidth, double NotchGain, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationNotchFilterSet (PositionerName, NotchNumber, NotchFrequency, NotchBandwidth, NotchGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchNumber: NotchNumber

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationNotchFilterSet command which is used to Update Notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationNotchFilterGet

Syntax

C# prototype

```
int PositionerCompensationNotchFilterGet(string PositionerName, Int32 NotchNumber, out double
NotchFrequency, out double NotchBandwidth, out double NotchGain, out string errstring)
```

Python prototype

```
[NotchFrequency, NotchBandwidth, NotchGain, errstring] PositionerCompensationNotchFilterGet (PositionerName,
NotchNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchNumber: NotchNumber

Output parameters

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationNotchFilterGet command which is used to Read Notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPhaseCorrectionFilterSet

Syntax

C# prototype

```
int PositionerCompensationPhaseCorrectionFilterSet(string PositionerName, Int32 PhaseCorrectionFilterNumber,
double PhaseCorrectionFn, double PhaseCorrectionFd, double PhaseCorrectionGain, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPhaseCorrectionFilterSet (PositionerName, PhaseCorrectionFilterNumber,
PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPhaseCorrectionFilterSet command which is used to Update one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPhaseCorrectionFilterGet

Syntax

C# prototype

```
int PositionerCompensationPhaseCorrectionFilterGet(string PositionerName, Int32 PhaseCorrectionFilterNumber,
out double PhaseCorrectionFn, out double PhaseCorrectionFd, out double PhaseCorrectionGain, out string
errstring)
```

Python prototype

```
[PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain, errstring]
PositionerCompensationPhaseCorrectionFilterGet (PositionerName, PhaseCorrectionFilterNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

Output parameters

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPhaseCorrectionFilterGet command which is used to Read one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorNotchFiltersSet

Syntax

C# prototype

```
int PositionerCorrectorNotchFiltersSet(string PositionerName, double NotchFrequency1, double NotchBandwidth1,
double NotchGain1, double NotchFrequency2, double NotchBandwidth2, double NotchGain2, out string errstring)
```

Python prototype

```
[errstring] PositionerCorrectorNotchFiltersSet (PositionerName, NotchFrequency1, NotchBandwidth1, NotchGain1,
NotchFrequency2, NotchBandwidth2, NotchGain2)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) NotchFrequency1: NotchFrequency1

(double) NotchBandwidth1: NotchBandwidth1

(double) NotchGain1: NotchGain1

(double) NotchFrequency2: NotchFrequency2

(double) NotchBandwidth2: NotchBandwidth2

(double) NotchGain2: NotchGain2

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorNotchFiltersSet command which is used to Update filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCorrectorNotchFiltersGet

Syntax

C# prototype

```
int PositionerCorrectorNotchFiltersGet(string PositionerName, out double NotchFrequency1, out double
NotchBandwidth1, out double NotchGain1, out double NotchFrequency2, out double NotchBandwidth2, out
double NotchGain2, out string errstring)
```

Python prototype

```
[NotchFrequency1, NotchBandwidth1, NotchGain1, NotchFrequency2, NotchBandwidth2, NotchGain2, errstring]
PositionerCorrectorNotchFiltersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) NotchFrequency1: NotchFrequency1

(double) NotchBandwidth1: NotchBandwidth1

(double) NotchGain1: NotchGain1

(double) NotchFrequency2: NotchFrequency2

(double) NotchBandwidth2: NotchBandwidth2

(double) NotchGain2: NotchGain2

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCorrectorNotchFiltersGet command which is used to Read filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPreFeedForwardFrequencyNotchFilterGet

Syntax

C# prototype

```
int PositionerCompensationPreFeedForwardFrequencyNotchFilterGet(string PositionerName, Int32
NotchFrequencyNumber, out double NotchFrequency, out double NotchBandwidth, out double NotchGain, out
string errstring)
```

Python prototype

```
[NotchFrequency, NotchBandwidth, NotchGain, errstring]
PositionerCompensationPreFeedForwardFrequencyNotchFilterGet (PositionerName, NotchFrequencyNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchFrequencyNumber: NotchFrequencyNumber

Output parameters

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPreFeedForwardFrequencyNotchFilterGet command which is used to Read one frequency compensation notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPreFeedForwardFrequencyNotchFilterSet

Syntax

C# prototype

```
int PositionerCompensationPreFeedForwardFrequencyNotchFilterSet(string PositionerName, Int32
NotchFrequencyNumber, double NotchFrequency, double NotchBandwidth, double NotchGain, out string
errstring)
```

Python prototype

```
[errstring] PositionerCompensationPreFeedForwardFrequencyNotchFilterSet (PositionerName,
NotchFrequencyNumber, NotchFrequency, NotchBandwidth, NotchGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchFrequencyNumber: NotchFrequencyNumber

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPreFeedForwardFrequencyNotchFilterSet command which is used to Update one frequency compensation notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPreFeedForwardSpatialNotchFilterGet

Syntax

C# prototype

```
int PositionerCompensationPreFeedForwardSpatialNotchFilterGet(string PositionerName, Int32
SpatialNotchNumber, out double SpatialNotchStep, out double SpatialNotchBandwidth, out double
SpatialNotchGain, out string errstring)
```

Python prototype

```
[SpatialNotchStep, SpatialNotchBandwidth, SpatialNotchGain, errstring]
PositionerCompensationPreFeedForwardSpatialNotchFilterGet (PositionerName, SpatialNotchNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) SpatialNotchNumber: SpatialNotchNumber

Output parameters

(double) SpatialNotchStep: SpatialNotchStep

(double) SpatialNotchBandwidth: SpatialNotchBandwidth

(double) SpatialNotchGain: SpatialNotchGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPreFeedForwardSpatialNotchFilterGet command which is used to Read one spatial periodic compensation notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPreFeedForwardSpatialNotchFilterSet

Syntax

C# prototype

```
int PositionerCompensationPreFeedForwardSpatialNotchFilterSet(string PositionerName, Int32
SpatialNotchNumber, double SpatialNotchStep, double SpatialNotchBandwidth, double SpatialNotchGain, out
string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPreFeedForwardSpatialNotchFilterSet (PositionerName, SpatialNotchNumber,
SpatialNotchStep, SpatialNotchBandwidth, SpatialNotchGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) SpatialNotchNumber: SpatialNotchNumber

(double) SpatialNotchStep: SpatialNotchStep

(double) SpatialNotchBandwidth: SpatialNotchBandwidth

(double) SpatialNotchGain: SpatialNotchGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPreFeedForwardSpatialNotchFilterSet command which is used to Update one spatial periodic compensation notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPreFeedForwardPhaseCorrectionFilterGet

Syntax

C# prototype

```
int PositionerCompensationPreFeedForwardPhaseCorrectionFilterGet(string PositionerName, Int32
PhaseCorrectionFilterNumber, out double PhaseCorrectionFn, out double PhaseCorrectionFd, out double
PhaseCorrectionGain, out string errstring)
```

Python prototype

```
[PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain, errstring]
PositionerCompensationPreFeedForwardPhaseCorrectionFilterGet (PositionerName, PhaseCorrectionFilterNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

Output parameters

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPreFeedForwardPhaseCorrectionFilterGet command which is used to Read one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPreFeedForwardPhaseCorrectionFilterSet

Syntax

C# prototype

```
int PositionerCompensationPreFeedForwardPhaseCorrectionFilterSet(string PositionerName, Int32
PhaseCorrectionFilterNumber, double PhaseCorrectionFn, double PhaseCorrectionFd, double PhaseCorrectionGain,
out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPreFeedForwardPhaseCorrectionFilterSet (PositionerName,
PhaseCorrectionFilterNumber, PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPreFeedForwardPhaseCorrectionFilterSet command which is used to Update one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationFrequencyNotchsGet

Syntax

C# prototype

```
int PositionerCompensationFrequencyNotchsGet(string PositionerName, out double NotchFrequency1, out double NotchBandwidth1, out double NotchGain1, out double NotchFrequency2, out double NotchBandwidth2, out double NotchGain2, out double NotchFrequency3, out double NotchBandwidth3, out double NotchGain3, out string errstring)
```

Python prototype

```
[NotchFrequency1, NotchBandwidth1, NotchGain1, NotchFrequency2, NotchBandwidth2, NotchGain2, NotchFrequency3, NotchBandwidth3, NotchGain3, errstring] PositionerCompensationFrequencyNotchsGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) NotchFrequency1: NotchFrequency1

(double) NotchBandwidth1: NotchBandwidth1

(double) NotchGain1: NotchGain1

(double) NotchFrequency2: NotchFrequency2

(double) NotchBandwidth2: NotchBandwidth2

(double) NotchGain2: NotchGain2

(double) NotchFrequency3: NotchFrequency3

(double) NotchBandwidth3: NotchBandwidth3

(double) NotchGain3: NotchGain3

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationFrequencyNotchsGet command which is used to Read frequency compensation notch filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationFrequencyNotchsSet

Syntax

C# prototype

```
int PositionerCompensationFrequencyNotchsSet(string PositionerName, double NotchFrequency1, double
NotchBandwidth1, double NotchGain1, double NotchFrequency2, double NotchBandwidth2, double NotchGain2,
double NotchFrequency3, double NotchBandwidth3, double NotchGain3, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationFrequencyNotchsSet (PositionerName, NotchFrequency1, NotchBandwidth1,
NotchGain1, NotchFrequency2, NotchBandwidth2, NotchGain2, NotchFrequency3, NotchBandwidth3, NotchGain3)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) NotchFrequency1: NotchFrequency1

(double) NotchBandwidth1: NotchBandwidth1

(double) NotchGain1: NotchGain1

(double) NotchFrequency2: NotchFrequency2

(double) NotchBandwidth2: NotchBandwidth2

(double) NotchGain2: NotchGain2

(double) NotchFrequency3: NotchFrequency3

(double) NotchBandwidth3: NotchBandwidth3

(double) NotchGain3: NotchGain3

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationFrequencyNotchsSet command which is used to Update frequency compensation notch filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationSpatialPeriodicNotchsGet

Syntax

C# prototype

```
int PositionerCompensationSpatialPeriodicNotchsGet(string PositionerName, out double SpatialNotchStep1, out double SpatialNotchBandwidth1, out double SpatialNotchGain1, out double SpatialNotchStep2, out double SpatialNotchBandwidth2, out double SpatialNotchGain2, out double SpatialNotchStep3, out double SpatialNotchBandwidth3, out double SpatialNotchGain3, out string errstring)
```

Python prototype

```
[SpatialNotchStep1, SpatialNotchBandwidth1, SpatialNotchGain1, SpatialNotchStep2, SpatialNotchBandwidth2, SpatialNotchGain2, SpatialNotchStep3, SpatialNotchBandwidth3, SpatialNotchGain3, errstring]  
PositionerCompensationSpatialPeriodicNotchsGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) SpatialNotchStep1: SpatialNotchStep1

(double) SpatialNotchBandwidth1: SpatialNotchBandwidth1

(double) SpatialNotchGain1: SpatialNotchGain1

(double) SpatialNotchStep2: SpatialNotchStep2

(double) SpatialNotchBandwidth2: SpatialNotchBandwidth2

(double) SpatialNotchGain2: SpatialNotchGain2

(double) SpatialNotchStep3: SpatialNotchStep3

(double) SpatialNotchBandwidth3: SpatialNotchBandwidth3

(double) SpatialNotchGain3: SpatialNotchGain3

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationSpatialPeriodicNotchsGet command which is used to Read spatial compensation notch filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationSpatialPeriodicNotchsSet

Syntax

C# prototype

```
int PositionerCompensationSpatialPeriodicNotchsSet(string PositionerName, double SpatialNotchStep1, double
SpatialNotchBandwidth1, double SpatialNotchGain1, double SpatialNotchStep2, double SpatialNotchBandwidth2,
double SpatialNotchGain2, double SpatialNotchStep3, double SpatialNotchBandwidth3, double SpatialNotchGain3,
out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationSpatialPeriodicNotchsSet (PositionerName, SpatialNotchStep1,
SpatialNotchBandwidth1, SpatialNotchGain1, SpatialNotchStep2, SpatialNotchBandwidth2, SpatialNotchGain2,
SpatialNotchStep3, SpatialNotchBandwidth3, SpatialNotchGain3)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) SpatialNotchStep1: SpatialNotchStep1

(double) SpatialNotchBandwidth1: SpatialNotchBandwidth1

(double) SpatialNotchGain1: SpatialNotchGain1

(double) SpatialNotchStep2: SpatialNotchStep2

(double) SpatialNotchBandwidth2: SpatialNotchBandwidth2

(double) SpatialNotchGain2: SpatialNotchGain2

(double) SpatialNotchStep3: SpatialNotchStep3

(double) SpatialNotchBandwidth3: SpatialNotchBandwidth3

(double) SpatialNotchGain3: SpatialNotchGain3

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationSpatialPeriodicNotchsSet command which is used to Update spatial compensation notch filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationLowPassFilterGet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationLowPassFilterGet(string PositionerName, out double CutOffFrequency,  
out string errstring)
```

Python prototype

```
[CutOffFrequency, errstring] PositionerCompensationPostExcitationLowPassFilterGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CutOffFrequency: CutOffFrequency

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationLowPassFilterGet command which is used to Read second order low-pass filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationLowPassFilterSet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationLowPassFilterSet(string PositionerName, double CutOffFrequency, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPostExcitationLowPassFilterSet (PositionerName, CutOffFrequency)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) CutOffFrequency: CutOffFrequency

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationLowPassFilterSet command which is used to Update second order low-pass filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationFrequencyNotchFilterGet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationFrequencyNotchFilterGet(string PositionerName, Int32
NotchFrequencyNumber, out double NotchFrequency, out double NotchBandwidth, out double NotchGain, out
string errstring)
```

Python prototype

```
[NotchFrequency, NotchBandwidth, NotchGain, errstring]
PositionerCompensationPostExcitationFrequencyNotchFilterGet (PositionerName, NotchFrequencyNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchFrequencyNumber: NotchFrequencyNumber

Output parameters

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationFrequencyNotchFilterGet command which is used to Read one frequency compensation notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationFrequencyNotchFilterSet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationFrequencyNotchFilterSet(string PositionerName, Int32
NotchFrequencyNumber, double NotchFrequency, double NotchBandwidth, double NotchGain, out string
errstring)
```

Python prototype

```
[errstring] PositionerCompensationPostExcitationFrequencyNotchFilterSet (PositionerName,
NotchFrequencyNumber, NotchFrequency, NotchBandwidth, NotchGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchFrequencyNumber: NotchFrequencyNumber

(double) NotchFrequency: NotchFrequency

(double) NotchBandwidth: NotchBandwidth

(double) NotchGain: NotchGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationFrequencyNotchFilterSet command which is used to Update one frequency compensation notch filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationNotchModeFilterGet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationNotchModeFilterGet(string PositionerName, Int32 NotchModeNumber,
out double NotchModeFr, out double NotchModeFa, out double NotchModeZr, out double NotchModeZa, out
string errstring)
```

Python prototype

```
[NotchModeFr, NotchModeFa, NotchModeZr, NotchModeZa, errstring]
PositionerCompensationPostExcitationNotchModeFilterGet (PositionerName, NotchModeNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchModeNumber: NotchModeNumber

Output parameters

(double) NotchModeFr: NotchModeFr

(double) NotchModeFa: NotchModeFa

(double) NotchModeZr: NotchModeZr

(double) NotchModeZa: NotchModeZa

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationNotchModeFilterGet command which is used to Read a notch mode filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationNotchModeFilterSet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationNotchModeFilterSet(string PositionerName, Int32 NotchModeNumber,
double NotchModeFr, double NotchModeFa, double NotchModeZr, double NotchModeZa, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPostExcitationNotchModeFilterSet (PositionerName, NotchModeNumber,
NotchModeFr, NotchModeFa, NotchModeZr, NotchModeZa)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) NotchModeNumber: NotchModeNumber

(double) NotchModeFr: NotchModeFr

(double) NotchModeFa: NotchModeFa

(double) NotchModeZr: NotchModeZr

(double) NotchModeZa: NotchModeZa

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationNotchModeFilterSet command which is used to Update a notch mode filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationPhaseCorrectionFilterGet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationPhaseCorrectionFilterGet(string PositionerName, Int32
PhaseCorrectionFilterNumber, out double PhaseCorrectionFn, out double PhaseCorrectionFd, out double
PhaseCorrectionGain, out string errstring)
```

Python prototype

```
[PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain, errstring]
PositionerCompensationPostExcitationPhaseCorrectionFilterGet (PositionerName, PhaseCorrectionFilterNumber)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

Output parameters

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationPhaseCorrectionFilterGet command which is used to Read one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPostExcitationPhaseCorrectionFilterSet

Syntax

C# prototype

```
int PositionerCompensationPostExcitationPhaseCorrectionFilterSet(string PositionerName, Int32
PhaseCorrectionFilterNumber, double PhaseCorrectionFn, double PhaseCorrectionFd, double PhaseCorrectionGain,
out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPostExcitationPhaseCorrectionFilterSet (PositionerName,
PhaseCorrectionFilterNumber, PhaseCorrectionFn, PhaseCorrectionFd, PhaseCorrectionGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) PhaseCorrectionFilterNumber: PhaseCorrectionFilterNumber

(double) PhaseCorrectionFn: PhaseCorrectionFn

(double) PhaseCorrectionFd: PhaseCorrectionFd

(double) PhaseCorrectionGain: PhaseCorrectionGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPostExcitationPhaseCorrectionFilterSet command which is used to Update one phase correction filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationLowPassTwoFilterGet

Syntax

C# prototype

```
int PositionerCompensationLowPassTwoFilterGet(string PositionerName, out double CutOffFrequency, out string errstring)
```

Python prototype

```
[CutOffFrequency, errstring] PositionerCompensationLowPassTwoFilterGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) CutOffFrequency: CutOffFrequency

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationLowPassTwoFilterGet command which is used to Read second order low-pass filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationLowPassTwoFilterSet

Syntax

C# prototype

```
int PositionerCompensationLowPassTwoFilterSet(string PositionerName, double CutOffFrequency, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationLowPassTwoFilterSet (PositionerName, CutOffFrequency)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) CutOffFrequency: CutOffFrequency

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationLowPassTwoFilterSet command which is used to Update second order low-pass filter parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationNotchModeFiltersGet

Syntax

C# prototype

```
int PositionerCompensationNotchModeFiltersGet(string PositionerName, out double NotchModeFr1, out double NotchModeFa1, out double NotchModeZr1, out double NotchModeZa1, out double NotchModeFr2, out double NotchModeFa2, out double NotchModeZr2, out double NotchModeZa2, out string errstring)
```

Python prototype

```
[NotchModeFr1, NotchModeFa1, NotchModeZr1, NotchModeZa1, NotchModeFr2, NotchModeFa2, NotchModeZr2, NotchModeZa2, errstring] PositionerCompensationNotchModeFiltersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) NotchModeFr1: NotchModeFr1

(double) NotchModeFa1: NotchModeFa1

(double) NotchModeZr1: NotchModeZr1

(double) NotchModeZa1: NotchModeZa1

(double) NotchModeFr2: NotchModeFr2

(double) NotchModeFa2: NotchModeFa2

(double) NotchModeZr2: NotchModeZr2

(double) NotchModeZa2: NotchModeZa2

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationNotchModeFiltersGet command which is used to Read notch mode filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationNotchModeFiltersSet

Syntax

C# prototype

```
int PositionerCompensationNotchModeFiltersSet(string PositionerName, double NotchModeFr1, double NotchModeFa1, double NotchModeZr1, double NotchModeZa1, double NotchModeFr2, double NotchModeFa2, double NotchModeZr2, double NotchModeZa2, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationNotchModeFiltersSet (PositionerName, NotchModeFr1, NotchModeFa1, NotchModeZr1, NotchModeZa1, NotchModeFr2, NotchModeFa2, NotchModeZr2, NotchModeZa2)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) NotchModeFr1: NotchModeFr1

(double) NotchModeFa1: NotchModeFa1

(double) NotchModeZr1: NotchModeZr1

(double) NotchModeZa1: NotchModeZa1

(double) NotchModeFr2: NotchModeFr2

(double) NotchModeFa2: NotchModeFa2

(double) NotchModeZr2: NotchModeZr2

(double) NotchModeZa2: NotchModeZa2

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationNotchModeFiltersSet command which is used to Update notch mode filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPhaseCorrectionFiltersGet

Syntax

C# prototype

```
int PositionerCompensationPhaseCorrectionFiltersGet(string PositionerName, out double PhaseCorrectionFn1, out double PhaseCorrectionFd1, out double PhaseCorrectionGain1, out double PhaseCorrectionFn2, out double PhaseCorrectionFd2, out double PhaseCorrectionGain2, out string errstring)
```

Python prototype

```
[PhaseCorrectionFn1, PhaseCorrectionFd1, PhaseCorrectionGain1, PhaseCorrectionFn2, PhaseCorrectionFd2, PhaseCorrectionGain2, errstring] PositionerCompensationPhaseCorrectionFiltersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PhaseCorrectionFn1: PhaseCorrectionFn1

(double) PhaseCorrectionFd1: PhaseCorrectionFd1

(double) PhaseCorrectionGain1: PhaseCorrectionGain1

(double) PhaseCorrectionFn2: PhaseCorrectionFn2

(double) PhaseCorrectionFd2: PhaseCorrectionFd2

(double) PhaseCorrectionGain2: PhaseCorrectionGain2

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPhaseCorrectionFiltersGet command which is used to Read phase correction filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerCompensationPhaseCorrectionFiltersSet

Syntax

C# prototype

```
int PositionerCompensationPhaseCorrectionFiltersSet(string PositionerName, double PhaseCorrectionFn1, double
PhaseCorrectionFd1, double PhaseCorrectionGain1, double PhaseCorrectionFn2, double PhaseCorrectionFd2,
double PhaseCorrectionGain2, out string errstring)
```

Python prototype

```
[errstring] PositionerCompensationPhaseCorrectionFiltersSet (PositionerName, PhaseCorrectionFn1,
PhaseCorrectionFd1, PhaseCorrectionGain1, PhaseCorrectionFn2, PhaseCorrectionFd2, PhaseCorrectionGain2)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PhaseCorrectionFn1: PhaseCorrectionFn1

(double) PhaseCorrectionFd1: PhaseCorrectionFd1

(double) PhaseCorrectionGain1: PhaseCorrectionGain1

(double) PhaseCorrectionFn2: PhaseCorrectionFn2

(double) PhaseCorrectionFd2: PhaseCorrectionFd2

(double) PhaseCorrectionGain2: PhaseCorrectionGain2

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerCompensationPhaseCorrectionFiltersSet command which is used to Update phase correction filters parameters . Refer to the XPS Programmer's manual to get the command description.

PositionerAnalogTrackingPositionParametersGet

Syntax

C# prototype

```
int PositionerAnalogTrackingPositionParametersGet(string PositionerName, out string GPIOName, out double Offset, out double Scale, out double Velocity, out double Acceleration, out string errstring)
```

Python prototype

```
[GPIOName, Offset, Scale, Velocity, Acceleration, errstring] PositionerAnalogTrackingPositionParametersGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) GPIOName: GPIOName

(double) Offset: Offset

(double) Scale: Scale

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerAnalogTrackingPositionParametersGet command which is used to Read dynamic parameters for one axe of a group for a future analog tracking position. Refer to the XPS Programmer's manual to get the command description.

PositionerAnalogTrackingPositionParametersSet

Syntax

C# prototype

```
int PositionerAnalogTrackingPositionParametersSet(string PositionerName, string GPIOName, double Offset,
double Scale, double Velocity, double Acceleration, out string errstring)
```

Python prototype

```
[errstring] PositionerAnalogTrackingPositionParametersSet (PositionerName, GPIOName, Offset, Scale, Velocity,
Acceleration)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) GPIOName: GPIOName

(double) Offset: Offset

(double) Scale: Scale

(double) Velocity: Velocity

(double) Acceleration: Acceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerAnalogTrackingPositionParametersSet command which is used to Update dynamic parameters for one axe of a group for a future analog tracking position. Refer to the XPS Programmer's manual to get the command description.

PositionerAnalogTrackingVelocityParametersGet

Syntax

C# prototype

```
int PositionerAnalogTrackingVelocityParametersGet(string PositionerName, out string GPIOName, out double
Offset, out double Scale, out double DeadBandThreshold, out Int32 Order, out double Velocity, out double
Acceleration, out string errstring)
```

Python prototype

```
[GPIOName, Offset, Scale, DeadBandThreshold, Order, Velocity, Acceleration, errstring]
PositionerAnalogTrackingVelocityParametersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) GPIOName: GPIOName

(double) Offset: Offset

(double) Scale: Scale

(double) DeadBandThreshold: DeadBandThreshold

(Int32_i) Order: Order

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerAnalogTrackingVelocityParametersGet command which is used to Read dynamic parameters for one axe of a group for a future analog tracking velocity. Refer to the XPS Programmer's manual to get the command description.

PositionerAnalogTrackingVelocityParametersSet

Syntax

C# prototype

```
int PositionerAnalogTrackingVelocityParametersSet(string PositionerName, string GPIOName, double Offset,
double Scale, double DeadBandThreshold, Int32 Order, double Velocity, double Acceleration, out string errstring)
```

Python prototype

```
[errstring] PositionerAnalogTrackingVelocityParametersSet (PositionerName, GPIOName, Offset, Scale,
DeadBandThreshold, Order, Velocity, Acceleration)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) GPIOName: GPIOName

(double) Offset: Offset

(double) Scale: Scale

(double) DeadBandThreshold: DeadBandThreshold

(Int32) Order: Order

(double) Velocity: Velocity

(double) Acceleration: Acceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerAnalogTrackingVelocityParametersSet command which is used to Update dynamic parameters for one axe of a group for a future analog tracking velocity. Refer to the XPS Programmer's manual to get the command description.

PositionerJogMaximumVelocityAndAccelerationGet

Syntax

C# prototype

```
int PositionerJogMaximumVelocityAndAccelerationGet(string PositionerName, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)
```

Python prototype

```
[MaximumVelocity, MaximumAcceleration, errstring] PositionerJogMaximumVelocityAndAccelerationGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerJogMaximumVelocityAndAccelerationGet command which is used to Return jog maximum velocity and acceleration of the positioner. Refer to the XPS Programmer's manual to get the command description.

PositionerMaximumVelocityAndAccelerationGet

Syntax

C# prototype

```
int PositionerMaximumVelocityAndAccelerationGet(string PositionerName, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)
```

Python prototype

```
[MaximumVelocity, MaximumAcceleration, errstring] PositionerMaximumVelocityAndAccelerationGet(PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMaximumVelocityAndAccelerationGet command which is used to Return maximum velocity and acceleration of the positioner. Refer to the XPS Programmer's manual to get the command description.

PositionerUserTravelLimitsGet

Syntax

C# prototype

```
int PositionerUserTravelLimitsGet(string PositionerName, out double UserMinimumTarget, out double
UserMaximumTarget, out string errstring)
```

Python prototype

```
[UserMinimumTarget, UserMaximumTarget, errstring] PositionerUserTravelLimitsGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) UserMinimumTarget: UserMinimumTarget

(double) UserMaximumTarget: UserMaximumTarget

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerUserTravelLimitsGet command which is used to Read UserMinimumTarget and UserMaximumTarget. Refer to the XPS Programmer's manual to get the command description.

PositionerUserTravelLimitsSet

Syntax

C# prototype

```
int PositionerUserTravelLimitsSet(string PositionerName, double UserMinimumTarget, double UserMaximumTarget,  
out string errstring)
```

Python prototype

```
[errstring] PositionerUserTravelLimitsSet (PositionerName, UserMinimumTarget, UserMaximumTarget)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) UserMinimumTarget: UserMinimumTarget

(double) UserMaximumTarget: UserMaximumTarget

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerUserTravelLimitsSet command which is used to Update UserMinimumTarget and UserMaximumTarget. Refer to the XPS Programmer's manual to get the command description.

PositionerSGammaExactVelocityAjustedDisplacementGet

Syntax

C# prototype

```
int PositionerSGammaExactVelocityAjustedDisplacementGet(string PositionerName, double DesiredDisplacement, out double AdjustedDisplacement, out string errstring)
```

Python prototype

```
[AdjustedDisplacement, errstring] PositionerSGammaExactVelocityAjustedDisplacementGet (PositionerName, DesiredDisplacement)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) DesiredDisplacement: DesiredDisplacement

Output parameters

(double) AdjustedDisplacement: AdjustedDisplacement

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerSGammaExactVelocityAjustedDisplacementGet command which is used to Calculate SGamma adjusted displacement to get exact velocity. Refer to the XPS Programmer's manual to get the command description.

PositionerSGammaMoveResultGet

Syntax

C# prototype

```
int PositionerSGammaMoveResultGet(string PositionerName, out double TimeToAccelerate, out double RealAcceleration, out double DistanceToAccelerate, out double TimeAtConstantSpeed, out double RealVelocity, out double DistanceAtConstantSpeed, out double TotalTimeToMove, out double DistanceToDecelerate, out double ExecutedDisplacement, out string errstring)
```

Python prototype

```
[TimeToAccelerate, RealAcceleration, DistanceToAccelerate, TimeAtConstantSpeed, RealVelocity, DistanceAtConstantSpeed, TotalTimeToMove, DistanceToDecelerate, ExecutedDisplacement, errstring]  
PositionerSGammaMoveResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) TimeToAccelerate: TimeToAccelerate

(double) RealAcceleration: RealAcceleration

(double) DistanceToAccelerate: DistanceToAccelerate

(double) TimeAtConstantSpeed: TimeAtConstantSpeed

(double) RealVelocity: RealVelocity

(double) DistanceAtConstantSpeed: DistanceAtConstantSpeed

(double) TotalTimeToMove: TotalTimeToMove

(double) DistanceToDecelerate: DistanceToDecelerate

(double) ExecutedDisplacement: ExecutedDisplacement

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerSGammaMoveResultGet command which is used to Get last SGamma move result. Refer to the XPS Programmer's manual to get the command description.

PositionerSGammaParametersGet

Syntax

C# prototype

```
int PositionerSGammaParametersGet(string PositionerName, out double Velocity, out double Acceleration, out double MinimumTjerkTime, out double MaximumTjerkTime, out string errstring)
```

Python prototype

```
[Velocity, Acceleration, MinimumTjerkTime, MaximumTjerkTime, errstring] PositionerSGammaParametersGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinimumTjerkTime: MinimumTjerkTime

(double) MaximumTjerkTime: MaximumTjerkTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerSGammaParametersGet command which is used to Get SGamma moving parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerSGammaParametersSet

Syntax

C# prototype

```
int PositionerSGammaParametersSet(string PositionerName, double Velocity, double Acceleration, double MinimumTjerkTime, double MaximumTjerkTime, out string errstring)
```

Python prototype

```
[errstring] PositionerSGammaParametersSet (PositionerName, Velocity, Acceleration, MinimumTjerkTime, MaximumTjerkTime)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinimumTjerkTime: MinimumTjerkTime

(double) MaximumTjerkTime: MaximumTjerkTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerSGammaParametersSet command which is used to Set SGamma moving parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerSGammaVelocityAndAccelerationSet

Syntax

C# prototype

```
int PositionerSGammaVelocityAndAccelerationSet(string PositionerName, double Velocity, double Acceleration,  
out string errstring)
```

Python prototype

```
[errstring] PositionerSGammaVelocityAndAccelerationSet (PositionerName, Velocity, Acceleration)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerSGammaVelocityAndAccelerationSet command which is used to Set SGamma velocity and acceleration. Refer to the XPS Programmer's manual to get the command description.

PositionerSGammaPreviousMotionTimesGet

Syntax

C# prototype

```
int PositionerSGammaPreviousMotionTimesGet(string PositionerName, out double SettingTime, out double SettlingTime, out string errstring)
```

Python prototype

```
[SettingTime, SettlingTime, errstring] PositionerSGammaPreviousMotionTimesGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) SettingTime: SettingTime

(double) SettlingTime: SettlingTime

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerSGammaPreviousMotionTimesGet command which is used to Get SGamma SettingTime and SettlingTime. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTVerification

Syntax

C# prototype

```
int MultipleAxesPTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTVerification command which is used to Multiple axes PT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTVerificationResultGet

Syntax

C# prototype

```
int MultipleAxesPTVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition,
out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string
errstring)
```

Python prototype

```
[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
MultipleAxesPTVerificationResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTVerificationResultGet command which is used to Multiple axes PT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTExecution

Syntax

C# prototype

```
int MultipleAxesPTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPTExecution (GroupName, TrajectoryFileName, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTExecution command which is used to Multiple axes PT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTParametersGet

Syntax

C# prototype

```
int MultipleAxesPTParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] MultipleAxesPTParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTParametersGet command which is used to Multiple axes PT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTPulseOutputSet

Syntax

C# prototype

```
int MultipleAxesPTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval,
out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTPulseOutputGet

Syntax

C# prototype

```
int MultipleAxesPTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double  
TimeInterval, out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] MultipleAxesPTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTLoadToMemory

Syntax

C# prototype

```
int MultipleAxesPTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTLoadToMemory command which is used to Multiple Axes Load PT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPTResetInMemory

Syntax

C# prototype

```
int MultipleAxesPTResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPTResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPTResetInMemory command which is used to Multiple Axes PT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

XYPTVerification

Syntax

C# prototype

```
int XYPTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] XYPTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTVerification command which is used to XY PT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

XYPTVerificationResultGet

Syntax

C# prototype

int XYPTVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)

Python prototype

[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
XYPTVerificationResultGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTVerificationResultGet command which is used to XY PT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

XYPTExecution

Syntax

C# prototype

int XYPTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)

Python prototype

[errstring] XYPTExecution (GroupName, TrajectoryFileName, ExecutionNumber)

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTExecution command which is used to XY PT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

XYPTParametersGet

Syntax

C# prototype

```
int XYPTParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] XYPTParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTParametersGet command which is used to XY PT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

XYPTPulseOutputSet

Syntax

C# prototype

```
int XYPTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval, out string errstring)
```

Python prototype

```
[errstring] XYPTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

XYTPulseOutputGet

Syntax

C# prototype

int XYTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double TimeInterval, out string errstring)

Python prototype

[StartElement, EndElement, TimeInterval, errstring] XYTPulseOutputGet (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

XYPTLoadToMemory

Syntax

C# prototype

```
int XYPTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] XYPTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTLoadToMemory command which is used to XY Load PT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

XYPTRResetInMemory

Syntax

C# prototype

```
int XYPTRResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYPTRResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPTRResetInMemory command which is used to XY PT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

TZPTVerification

Syntax

C# prototype

```
int TZPTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] TZPTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTVerification command which is used to TZ PT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

TZPTVerificationResultGet

Syntax

C# prototype

int TZPTVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)

Python prototype

[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
TZPTVerificationResultGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTVerificationResultGet command which is used to TZ PT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

TZPTExecution

Syntax

C# prototype

```
int TZPTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] TZPTExecution (GroupName, TrajectoryFileName, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTExecution command which is used to TZ PT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

TZPTParametersGet

Syntax

C# prototype

```
int TZPTParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] TZPTParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTParametersGet command which is used to TZ PT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

TZPTPulseOutputSet

Syntax

C# prototype

```
int TZPTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval, out string errstring)
```

Python prototype

```
[errstring] TZPTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

TZPTPulseOutputGet

Syntax

C# prototype

```
int TZPTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double TimeInterval,
out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] TZPTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

TZPTLoadToMemory

Syntax

C# prototype

```
int TZPTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] TZPTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTLoadToMemory command which is used to TZ Load PT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

TZPTResetInMemory

Syntax

C# prototype

```
int TZPTResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] TZPTResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPTResetInMemory command which is used to TZ PT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTVerification

Syntax

C# prototype

```
int HexapodLegsPTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTVerification command which is used to Hexapod legs PT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTVerificationResultGet

Syntax

C# prototype

```
int HexapodLegsPTVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition,
out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string
errstring)
```

Python prototype

```
[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
HexapodLegsPTVerificationResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTVerificationResultGet command which is used to Hexapod legs PT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTExecution

Syntax

C# prototype

```
int HexapodLegsPTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPTExecution (GroupName, TrajectoryFileName, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTExecution command which is used to Hexapod legs PT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTParametersGet

Syntax

C# prototype

```
int HexapodLegsPTParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] HexapodLegsPTParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTParametersGet command which is used to Hexapod legs PT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTPulseOutputSet

Syntax

C# prototype

```
int HexapodLegsPTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval,
out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTPulseOutputGet

Syntax

C# prototype

```
int HexapodLegsPTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double TimeInterval, out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] HexapodLegsPTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTSave

Syntax

C# prototype

```
int HexapodLegsPTSave(string GroupName, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPTSave (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTSave command which is used to Save Hexapod legs PT trajectory from memory in a text file. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTLoadToMemory

Syntax

C# prototype

```
int HexapodLegsPTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTLoadToMemory command which is used to Load Hexapod legs PT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPTResetInMemory

Syntax

C# prototype

```
int HexapodLegsPTResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPTResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPTResetInMemory command which is used to Hexapod legs PT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTVerification

Syntax

C# prototype

```
int MultipleAxesPVTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPVTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTVerification command which is used to Multiple axes PVT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTVerificationResultGet

Syntax

C# prototype

```
int MultipleAxesPVTVerificationResultGet(string PositionerName, out string FileName, out double
MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration,
out string errstring)
```

Python prototype

```
[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
MultipleAxesPVTVerificationResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTVerificationResultGet command which is used to Multiple axes PVT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTExecution

Syntax

C# prototype

```
int MultipleAxesPVTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPVTExecution (GroupName, TrajectoryFileName, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTExecution command which is used to Multiple axes PVT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTPParametersGet

Syntax

C# prototype

```
int MultipleAxesPVTPParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] MultipleAxesPVTPParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTPParametersGet command which is used to Multiple axes PVT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTPulseOutputSet

Syntax

C# prototype

```
int MultipleAxesPVTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval,
out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPVTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTPulseOutputGet

Syntax

C# prototype

```
int MultipleAxesPVTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double
TimeInterval, out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] MultipleAxesPVTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTLoadToMemory

Syntax

C# prototype

```
int MultipleAxesPVTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPVTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTLoadToMemory command which is used to Multiple Axes Load PVT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesPVTRresetInMemory

Syntax

C# prototype

```
int MultipleAxesPVTRresetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesPVTRresetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesPVTRresetInMemory command which is used to Multiple Axes PVT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

XYPVTVerification

Syntax

C# prototype

```
int XYPVTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] XYPVTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTVerification command which is used to XY PVT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

XYPVTVerificationResultGet

Syntax

C# prototype

int XYPVTVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)

Python prototype

[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
XYPVTVerificationResultGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTVerificationResultGet command which is used to XY PVT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

XYPVTExecution

Syntax

C# prototype

int XYPVTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)

Python prototype

[errstring] XYPVTExecution (GroupName, TrajectoryFileName, ExecutionNumber)

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTExecution command which is used to XY PVT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

XYPVTPParametersGet

Syntax

C# prototype

```
int XYPVTPParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] XYPVTPParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTPParametersGet command which is used to XY PVT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

XYPVTPulseOutputSet

Syntax

C# prototype

```
int XYPVTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval, out string errstring)
```

Python prototype

```
[errstring] XYPVTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

XYPVTPulseOutputGet

Syntax

C# prototype

```
int XYPVTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double
TimeInterval, out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] XYPVTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

XYPVTLoadToMemory

Syntax

C# prototype

```
int XYPVTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] XYPVTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTLoadToMemory command which is used to XY Load PVT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

XYPVTRResetInMemory

Syntax

C# prototype

```
int XYPVTRResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYPVTRResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYPVTRResetInMemory command which is used to XY PVT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

TZPVTVerification

Syntax

C# prototype

```
int TZPVTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] TZPVTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTVerification command which is used to TZ PVT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

TZPVTVerificationResultGet

Syntax

C# prototype

int TZPVTVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)

Python prototype

[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
TZPVTVerificationResultGet (PositionerName)

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTVerificationResultGet command which is used to TZ PVT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

TZPVTExecution

Syntax

C# prototype

```
int TZPVTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] TZPVTExecution (GroupName, TrajectoryFileName, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTExecution command which is used to TZ PVT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

TZPVTParametersGet

Syntax

C# prototype

```
int TZPVTParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] TZPVTParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTParametersGet command which is used to TZ PVT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

TZPVTPulseOutputSet

Syntax

C# prototype

```
int TZPVTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval, out string errstring)
```

Python prototype

```
[errstring] TZPVTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

TZPVTPulseOutputGet

Syntax

C# prototype

```
int TZPVTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double
TimeInterval, out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] TZPVTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

TZPVTLoadToMemory

Syntax

C# prototype

```
int TZPVTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] TZPVTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTLoadToMemory command which is used to TZ Load PVT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

TZPVTRResetInMemory

Syntax

C# prototype

```
int TZPVTRResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] TZPVTRResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZPVTRResetInMemory command which is used to TZ PVT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTVerification

Syntax

C# prototype

```
int HexapodLegsPVTVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPVTVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTVerification command which is used to Hexapod legs PVT trajectory verification. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTVerificationResultGet

Syntax

C# prototype

```
int HexapodLegsPVTVerificationResultGet(string PositionerName, out string FileName, out double
MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration,
out string errstring)
```

Python prototype

```
[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]
HexapodLegsPVTVerificationResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTVerificationResultGet command which is used to Hexapod legs PVT trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTExecution

Syntax

C# prototype

```
int HexapodLegsPVTExecution(string GroupName, string TrajectoryFileName, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPVTExecution (GroupName, TrajectoryFileName, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTExecution command which is used to Hexapod legs PVT trajectory execution. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTPParametersGet

Syntax

C# prototype

```
int HexapodLegsPVTPParametersGet(string GroupName, out string FileName, out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, CurrentElementNumber, errstring] HexapodLegsPVTPParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTPParametersGet command which is used to Hexapod legs PVT trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTPulseOutputSet

Syntax

C# prototype

```
int HexapodLegsPVTPulseOutputSet(string GroupName, Int32 StartElement, Int32 EndElement, double TimeInterval,
out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPVTPulseOutputSet (GroupName, StartElement, EndElement, TimeInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) StartElement: StartElement

(Int32) EndElement: EndElement

(double) TimeInterval: TimeInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTPulseOutputSet command which is used to Configure pulse output on legs trajectory. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTPulseOutputGet

Syntax

C# prototype

```
int HexapodLegsPVTPulseOutputGet(string GroupName, out Int32 StartElement, out Int32 EndElement, out double TimeInterval, out string errstring)
```

Python prototype

```
[StartElement, EndElement, TimeInterval, errstring] HexapodLegsPVTPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) StartElement: StartElement

(Int32_i) EndElement: EndElement

(double) TimeInterval: TimeInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTPulseOutputGet command which is used to Get pulse output on legs trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTSave

Syntax

C# prototype

```
int HexapodLegsPVTSave(string GroupName, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPVTSave (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTSave command which is used to Save Hexapod legs PVT trajectory from memory in a text file. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTLoadToMemory

Syntax

C# prototype

```
int HexapodLegsPVTLoadToMemory(string GroupName, string TrajectoryPart, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPVTLoadToMemory (GroupName, TrajectoryPart)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryPart: TrajectoryPart

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTLoadToMemory command which is used to Load Hexapod legs PVT trajectory through function. Refer to the XPS Programmer's manual to get the command description.

HexapodLegsPVTResetInMemory

Syntax

C# prototype

```
int HexapodLegsPVTResetInMemory(string GroupName, out string errstring)
```

Python prototype

```
[errstring] HexapodLegsPVTResetInMemory (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodLegsPVTResetInMemory command which is used to Hexapod legs PVT trajectory reset in memory. Refer to the XPS Programmer's manual to get the command description.

XYLineArcVerification

Syntax

C# prototype

```
int XYLineArcVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] XYLineArcVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYLineArcVerification command which is used to XY trajectory verification. Refer to the XPS Programmer's manual to get the command description.

XYLineArcVerificationResultGet

Syntax

C# prototype

```
int XYLineArcVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)
```

Python prototype

```
[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]  
XYLineArcVerificationResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYLineArcVerificationResultGet command which is used to XY trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

XYLineArcExecution

Syntax

C# prototype

```
int XYLineArcExecution(string GroupName, string TrajectoryFileName, double Velocity, double Acceleration, Int32 ExecutionNumber, out string errstring)
```

Python prototype

```
[errstring] XYLineArcExecution (GroupName, TrajectoryFileName, Velocity, Acceleration, ExecutionNumber)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYLineArcExecution command which is used to XY trajectory execution. Refer to the XPS Programmer's manual to get the command description.

XYLineArcParametersGet

Syntax

C# prototype

```
int XYLineArcParametersGet(string GroupName, out string FileName, out double Velocity, out double Acceleration,  
out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, Velocity, Acceleration, CurrentElementNumber, errstring] XYLineArcParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYLineArcParametersGet command which is used to XY trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

XYLineArcPulseOutputSet

Syntax

C# prototype

```
int XYLineArcPulseOutputSet(string GroupName, double StartLength, double EndLength, double PathLengthInterval, out string errstring)
```

Python prototype

```
[errstring] XYLineArcPulseOutputSet (GroupName, StartLength, EndLength, PathLengthInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) StartLength: StartLength

(double) EndLength: EndLength

(double) PathLengthInterval: PathLengthInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYLineArcPulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

XYLineArcPulseOutputGet

Syntax

C# prototype

```
int XYLineArcPulseOutputGet(string GroupName, out double StartLength, out double EndLength, out double PathLengthInterval, out string errstring)
```

Python prototype

```
[StartLength, EndLength, PathLengthInterval, errstring] XYLineArcPulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) StartLength: StartLength

(double) EndLength: EndLength

(double) PathLengthInterval: PathLengthInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYLineArcPulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

XYZSplineVerification

Syntax

C# prototype

```
int XYZSplineVerification(string GroupName, string TrajectoryFileName, out string errstring)
```

Python prototype

```
[errstring] XYZSplineVerification (GroupName, TrajectoryFileName)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZSplineVerification command which is used to XYZ trajectory verification. Refer to the XPS Programmer's manual to get the command description.

XYZSplineVerificationResultGet

Syntax

C# prototype

```
int XYZSplineVerificationResultGet(string PositionerName, out string FileName, out double MinimumPosition, out double MaximumPosition, out double MaximumVelocity, out double MaximumAcceleration, out string errstring)
```

Python prototype

```
[FileName, MinimumPosition, MaximumPosition, MaximumVelocity, MaximumAcceleration, errstring]  
XYZSplineVerificationResultGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) FileName: FileName

(double) MinimumPosition: MinimumPosition

(double) MaximumPosition: MaximumPosition

(double) MaximumVelocity: MaximumVelocity

(double) MaximumAcceleration: MaximumAcceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZSplineVerificationResultGet command which is used to XYZ trajectory verification result get. Refer to the XPS Programmer's manual to get the command description.

XYZSplineExecution

Syntax

C# prototype

```
int XYZSplineExecution(string GroupName, string TrajectoryFileName, double Velocity, double Acceleration, out
string errstring)
```

Python prototype

```
[errstring] XYZSplineExecution (GroupName, TrajectoryFileName, Velocity, Acceleration)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TrajectoryFileName: TrajectoryFileName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZSplineExecution command which is used to XYZ trajectory execution. Refer to the XPS Programmer's manual to get the command description.

XYZSplineParametersGet

Syntax

C# prototype

```
int XYZSplineParametersGet(string GroupName, out string FileName, out double Velocity, out double Acceleration,  
out Int32 CurrentElementNumber, out string errstring)
```

Python prototype

```
[FileName, Velocity, Acceleration, CurrentElementNumber, errstring] XYZSplineParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) FileName: FileName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(Int32_i) CurrentElementNumber: CurrentElementNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZSplineParametersGet command which is used to XYZ trajectory get parameters. Refer to the XPS Programmer's manual to get the command description.

XYZSplinePulseOutputSet

Syntax

C# prototype

```
int XYZSplinePulseOutputSet(string GroupName, double StartLength, double EndLength, double
PathLengthInterval, out string errstring)
```

Python prototype

```
[errstring] XYZSplinePulseOutputSet (GroupName, StartLength, EndLength, PathLengthInterval)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) StartLength: StartLength

(double) EndLength: EndLength

(double) PathLengthInterval: PathLengthInterval

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZSplinePulseOutputSet command which is used to Configure pulse output on trajectory. Refer to the XPS Programmer's manual to get the command description.

XYZSplinePulseOutputGet

Syntax

C# prototype

```
int XYZSplinePulseOutputGet(string GroupName, out double StartLength, out double EndLength, out double PathLengthInterval, out string errstring)
```

Python prototype

```
[StartLength, EndLength, PathLengthInterval, errstring] XYZSplinePulseOutputGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) StartLength: StartLength

(double) EndLength: EndLength

(double) PathLengthInterval: PathLengthInterval

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZSplinePulseOutputGet command which is used to Get pulse output on trajectory configuration. Refer to the XPS Programmer's manual to get the command description.

SingleAxisSlaveModeEnable

Syntax

C# prototype

```
int SingleAxisSlaveModeEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SingleAxisSlaveModeEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisSlaveModeEnable command which is used to Enable the slave mode. Refer to the XPS Programmer's manual to get the command description.

SingleAxisSlaveModeDisable

Syntax

C# prototype

```
int SingleAxisSlaveModeDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SingleAxisSlaveModeDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisSlaveModeDisable command which is used to Disable the slave mode. Refer to the XPS Programmer's manual to get the command description.

SingleAxisSlaveParametersSet

Syntax

C# prototype

```
int SingleAxisSlaveParametersSet(string GroupName, string PositionerName, double Ratio, out string errstring)
```

Python prototype

```
[errstring] SingleAxisSlaveParametersSet (GroupName, PositionerName, Ratio)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) PositionerName: PositionerName

(double) Ratio: Ratio

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisSlaveParametersSet command which is used to Set slave parameters. Refer to the XPS Programmer's manual to get the command description.

SingleAxisSlaveParametersGet

Syntax

C# prototype

```
int SingleAxisSlaveParametersGet(string GroupName, out string PositionerName, out double Ratio, out string errstring)
```

Python prototype

```
[PositionerName, Ratio, errstring] SingleAxisSlaveParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) PositionerName: PositionerName

(double) Ratio: Ratio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisSlaveParametersGet command which is used to Get slave parameters. Refer to the XPS Programmer's manual to get the command description.

SpindleSlaveModeEnable

Syntax

C# prototype

```
int SpindleSlaveModeEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SpindleSlaveModeEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SpindleSlaveModeEnable command which is used to Enable the slave mode. Refer to the XPS Programmer's manual to get the command description.

SpindleSlaveModeDisable

Syntax

C# prototype

int SpindleSlaveModeDisable(string GroupName, out string errstring)

Python prototype

[errstring] SpindleSlaveModeDisable (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SpindleSlaveModeDisable command which is used to Disable the slave mode. Refer to the XPS Programmer's manual to get the command description.

SpindleSlaveParametersSet

Syntax

C# prototype

```
int SpindleSlaveParametersSet(string GroupName, string PositionerName, double Ratio, out string errstring)
```

Python prototype

```
[errstring] SpindleSlaveParametersSet (GroupName, PositionerName, Ratio)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) PositionerName: PositionerName

(double) Ratio: Ratio

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SpindleSlaveParametersSet command which is used to Set slave parameters. Refer to the XPS Programmer's manual to get the command description.

SpindleSlaveParametersGet

Syntax

C# prototype

```
int SpindleSlaveParametersGet(string GroupName, out string PositionerName, out double Ratio, out string errstring)
```

Python prototype

```
[PositionerName, Ratio, errstring] SpindleSlaveParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) PositionerName: PositionerName

(double) Ratio: Ratio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SpindleSlaveParametersGet command which is used to Get slave parameters. Refer to the XPS Programmer's manual to get the command description.

GroupSpinParametersSet

Syntax

C# prototype

```
int GroupSpinParametersSet(string GroupName, double Velocity, double Acceleration, out string errstring)
```

Python prototype

```
[errstring] GroupSpinParametersSet (GroupName, Velocity, Acceleration)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) Velocity: Velocity

(double) Acceleration: Acceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupSpinParametersSet command which is used to Modify Spin parameters on selected group and activate the continuous move. Refer to the XPS Programmer's manual to get the command description.

GroupSpinParametersGet

Syntax

C# prototype

```
int GroupSpinParametersGet(string GroupName, out double Velocity, out double Acceleration, out string errstring)
```

Python prototype

```
[Velocity, Acceleration, errstring] GroupSpinParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupSpinParametersGet command which is used to Get Spin parameters on selected group. Refer to the XPS Programmer's manual to get the command description.

GroupSpinCurrentGet

Syntax

C# prototype

```
int GroupSpinCurrentGet(string GroupName, out double Velocity, out double Acceleration, out string errstring)
```

Python prototype

```
[Velocity, Acceleration, errstring] GroupSpinCurrentGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupSpinCurrentGet command which is used to Get Spin current on selected group. Refer to the XPS Programmer's manual to get the command description.

GroupSpinModeStop

Syntax

C# prototype

```
int GroupSpinModeStop(string GroupName, double Acceleration, out string errstring)
```

Python prototype

```
[errstring] GroupSpinModeStop (GroupName, Acceleration)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) Acceleration: Acceleration

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupSpinModeStop command which is used to Stop Spin mode on selected group with specified acceleration. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesDisableShutter

Syntax

C# prototype

```
int MultipleAxesDisableShutter(string GroupName, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesDisableShutter (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesDisableShutter command which is used to Go back to ready state. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesScanPositions

Syntax

C# prototype

```
int MultipleAxesScanPositions(string GroupName, double XStartOpeningPosition, double XStartOpeningVelocity,
double XEndOpeningPosition, double XEndOpeningVelocity, double XStartClosingPosition, double
XStartClosingVelocity, double XEndClosingPosition, double XEndClosingVelocity, double BeamOpeningPercentage,
out string errstring)
```

Python prototype

```
[errstring] MultipleAxesScanPositions (GroupName, XStartOpeningPosition, XStartOpeningVelocity,
XEndOpeningPosition, XEndOpeningVelocity, XStartClosingPosition, XStartClosingVelocity, XEndClosingPosition,
XEndClosingVelocity, BeamOpeningPercentage)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) XStartOpeningPosition: XStartOpeningPosition

(double) XStartOpeningVelocity: XStartOpeningVelocity

(double) XEndOpeningPosition: XEndOpeningPosition

(double) XEndOpeningVelocity: XEndOpeningVelocity

(double) XStartClosingPosition: XStartClosingPosition

(double) XStartClosingVelocity: XStartClosingVelocity

(double) XEndClosingPosition: XEndClosingPosition

(double) XEndClosingVelocity: XEndClosingVelocity

(double) BeamOpeningPercentage: BeamOpeningPercentage

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesScanPositions command which is used to Scan positions. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesGetShutterPositions

Syntax

C# prototype

```
int MultipleAxesGetShutterPositions(string GroupName, out double BeamBlockedMin, out double
BeamBlockedMax, out double ShutterPosition2, out double ShutterPosition3, out double ShutterPosition4, out
double ShutterPosition5, out string errstring)
```

Python prototype

```
[BeamBlockedMin, BeamBlockedMax, ShutterPosition2, ShutterPosition3, ShutterPosition4, ShutterPosition5,
errstring] MultipleAxesGetShutterPositions (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) BeamBlockedMin: BeamBlockedMin

(double) BeamBlockedMax: BeamBlockedMax

(double) ShutterPosition2: ShutterPosition2

(double) ShutterPosition3: ShutterPosition3

(double) ShutterPosition4: ShutterPosition4

(double) ShutterPosition5: ShutterPosition5

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesGetShutterPositions command which is used to Get shutter positions. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesSetShutterPositions

Syntax

C# prototype

```
int MultipleAxesSetShutterPositions(string GroupName, double BeamBlockedMin, double BeamBlockedMax,
double ShutterPosition2, double ShutterPosition3, double ShutterPosition4, double ShutterPosition5, out string
errstring)
```

Python prototype

```
[errstring] MultipleAxesSetShutterPositions (GroupName, BeamBlockedMin, BeamBlockedMax, ShutterPosition2,
ShutterPosition3, ShutterPosition4, ShutterPosition5)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) BeamBlockedMin: BeamBlockedMin

(double) BeamBlockedMax: BeamBlockedMax

(double) ShutterPosition2: ShutterPosition2

(double) ShutterPosition3: ShutterPosition3

(double) ShutterPosition4: ShutterPosition4

(double) ShutterPosition5: ShutterPosition5

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesSetShutterPositions command which is used to Set shutter positions. Refer to the XPS Programmer's manual to get the command description.

MultipleAxesTraceNextScan

Syntax

C# prototype

```
int MultipleAxesTraceNextScan(string GroupName, out string errstring)
```

Python prototype

```
[errstring] MultipleAxesTraceNextScan (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MultipleAxesTraceNextScan command which is used to Write the next PVT trajectory in a file for debugging. Refer to the XPS Programmer's manual to get the command description.

XYClampDisable

Syntax

C# prototype

```
int XYClampDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYClampDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYClampDisable command which is used to Set clamping disable on selected group. Refer to the XPS Programmer's manual to get the command description.

XYClampEnable

Syntax

C# prototype

```
int XYClampEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYClampEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYClampEnable command which is used to Set clamping enable on selected group. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardAccDisable

Syntax

C# prototype

```
int PositionerFeedforwardAccDisable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerFeedforwardAccDisable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardAccDisable command which is used to Disable XY Feed forward Acceleration feature. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardAccEnable

Syntax

C# prototype

```
int PositionerFeedforwardAccEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerFeedforwardAccEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardAccEnable command which is used to Enable XY Feed forward Acceleration feature. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardAccGet

Syntax

C# prototype

```
int PositionerFeedforwardAccGet(string PositionerName, out string OutputName1, out double Scale1, out string OutputName2, out double Scale2, out string errstring)
```

Python prototype

```
[OutputName1, Scale1, OutputName2, Scale2, errstring] PositionerFeedforwardAccGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) OutputName1: OutputName1

(double) Scale1: Scale1

(string) OutputName2: OutputName2

(double) Scale2: Scale2

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardAccGet command which is used to Get XY Feed forward Acceleration parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardAccSet

Syntax

C# prototype

```
int PositionerFeedforwardAccSet(string PositionerName, string OutputName1, double Scale1, string OutputName2, double Scale2, out string errstring)
```

Python prototype

```
[errstring] PositionerFeedforwardAccSet (PositionerName, OutputName1, Scale1, OutputName2, Scale2)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) OutputName1: OutputName1

(double) Scale1: Scale1

(string) OutputName2: OutputName2

(double) Scale2: Scale2

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardAccSet command which is used to Set XY Feed forward Acceleration parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardAccStatusGet

Syntax

C# prototype

```
int PositionerFeedforwardAccStatusGet(string PositionerName, out string Status, out string errstring)
```

Python prototype

```
[Status, errstring] PositionerFeedforwardAccStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardAccStatusGet command which is used to Get current XY Feed forward Acceleration status . Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardPositionDisable

Syntax

C# prototype

```
int PositionerFeedforwardPositionDisable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerFeedforwardPositionDisable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardPositionDisable command which is used to Disable XY Feed forward Position feature. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardPositionEnable

Syntax

C# prototype

```
int PositionerFeedforwardPositionEnable(string PositionerName, out string errstring)
```

Python prototype

```
[errstring] PositionerFeedforwardPositionEnable (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardPositionEnable command which is used to Enable XY Feed forward Position feature. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardPositionGet

Syntax

C# prototype

```
int PositionerFeedforwardPositionGet(string PositionerName, out string OutputName, out double Scale, out double Offset, out string errstring)
```

Python prototype

```
[OutputName, Scale, Offset, errstring] PositionerFeedforwardPositionGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) OutputName: OutputName

(double) Scale: Scale

(double) Offset: Offset

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardPositionGet command which is used to Get XY Feed forward Position parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardPositionSet

Syntax

C# prototype

```
int PositionerFeedforwardPositionSet(string PositionerName, string OutputName, double Scale, double Offset, out
string errstring)
```

Python prototype

```
[errstring] PositionerFeedforwardPositionSet (PositionerName, OutputName, Scale, Offset)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(string) OutputName: OutputName

(double) Scale: Scale

(double) Offset: Offset

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardPositionSet command which is used to Set XY Feed forward Position parameters. Refer to the XPS Programmer's manual to get the command description.

PositionerFeedforwardPositionStatusGet

Syntax

C# prototype

```
int PositionerFeedforwardPositionStatusGet(string PositionerName, out string Status, out string errstring)
```

Python prototype

```
[Status, errstring] PositionerFeedforwardPositionStatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(string) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerFeedforwardPositionStatusGet command which is used to Get current XY Feed forward Position status . Refer to the XPS Programmer's manual to get the command description.

GroupBrakeSet

Syntax

C# prototype

```
int GroupBrakeSet(string GroupName, Int32 Command, out string errstring)
```

Python prototype

```
[errstring] GroupBrakeSet (GroupName, Command)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Command: Command

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupBrakeSet command which is used to Set Brake command . Refer to the XPS Programmer's manual to get the command description.

GroupBrakeStateGet

Syntax

C# prototype

```
int GroupBrakeStateGet(string GroupName, out Int32 CommandState, out string errstring)
```

Python prototype

```
[CommandState, errstring] GroupBrakeStateGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) CommandState: CommandState

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupBrakeStateGet command which is used to Get current Brake State . Refer to the XPS Programmer's manual to get the command description.

XYCrossTalkCompensationMotorDecouplingSet

Syntax

C# prototype

```
int XYCrossTalkCompensationMotorDecouplingSet(string GroupName, Int32 Mode, double YToX1FFAccRatio,
double YToX2FFAccRatio, double X1ToYFFAccRatio, double X2ToYFFAccRatio, out string errstring)
```

Python prototype

```
[errstring] XYCrossTalkCompensationMotorDecouplingSet (GroupName, Mode, YToX1FFAccRatio,
YToX2FFAccRatio, X1ToYFFAccRatio, X2ToYFFAccRatio)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Mode: Mode

(double) YToX1FFAccRatio: YToX1FFAccRatio

(double) YToX2FFAccRatio: YToX2FFAccRatio

(double) X1ToYFFAccRatio: X1ToYFFAccRatio

(double) X2ToYFFAccRatio: X2ToYFFAccRatio

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYCrossTalkCompensationMotorDecouplingSet command which is used to Set XY CrossTalkCompensation MotorDecoupling parameters. Refer to the XPS Programmer's manual to get the command description.

XYCrossTalkCompensationMotorDecouplingGet

Syntax

C# prototype

```
int XYCrossTalkCompensationMotorDecouplingGet(string GroupName, out Int32 Mode, out double
YToX1FFAccRatio, out double YToX2FFAccRatio, out double X1ToYFFAccRatio, out double X2ToYFFAccRatio, out
string errstring)
```

Python prototype

```
[Mode, YToX1FFAccRatio, YToX2FFAccRatio, X1ToYFFAccRatio, X2ToYFFAccRatio, errstring]
XYCrossTalkCompensationMotorDecouplingGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) Mode: Mode

(double) YToX1FFAccRatio: YToX1FFAccRatio

(double) YToX2FFAccRatio: YToX2FFAccRatio

(double) X1ToYFFAccRatio: X1ToYFFAccRatio

(double) X2ToYFFAccRatio: X2ToYFFAccRatio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYCrossTalkCompensationMotorDecouplingGet command which is used to Get XY CrossTalkCompensation MotorDecoupling parameters. Refer to the XPS Programmer's manual to get the command description.

XYGroupPositionPCORawEncoderGet

Syntax

C# prototype

```
int XYGroupPositionPCORawEncoderGet(string GroupName, double PositionX, double PositionY, out double PCORawPositionX, out double PCORawPositionY, out string errstring)
```

Python prototype

```
[PCORawPositionX, PCORawPositionY, errstring] XYGroupPositionPCORawEncoderGet (GroupName, PositionX, PositionY)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) PositionX: PositionX

(double) PositionY: PositionY

Output parameters

(double) PCORawPositionX: PCORawPositionX

(double) PCORawPositionY: PCORawPositionY

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYGroupPositionPCORawEncoderGet command which is used to Return PCO raw encoder positions. Refer to the XPS Programmer's manual to get the command description.

XYGroupPositionCorrectedProfilerGet

Syntax

C# prototype

```
int XYGroupPositionCorrectedProfilerGet(string GroupName, double PositionX, double PositionY, out double  
CorrectedProfilerPositionX, out double CorrectedProfilerPositionY, out string errstring)
```

Python prototype

```
[CorrectedProfilerPositionX, CorrectedProfilerPositionY, errstring] XYGroupPositionCorrectedProfilerGet  
(GroupName, PositionX, PositionY)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) PositionX: PositionX

(double) PositionY: PositionY

Output parameters

(double) CorrectedProfilerPositionX: CorrectedProfilerPositionX

(double) CorrectedProfilerPositionY: CorrectedProfilerPositionY

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYGroupPositionCorrectedProfilerGet command which is used to Return corrected profiler positions. Refer to the XPS Programmer's manual to get the command description.

XYDualCorrectorMappingGet

Syntax

C# prototype

```
int XYDualCorrectorMappingGet(string PositionerName, Int32 LineNumber, out double[] Value, Int32 nbItems, out string errstring)
```

Python prototype

```
[Value, errstring] XYDualCorrectorMappingGet (PositionerName, LineNumber, nbItems)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) LineNumber: LineNumber

(Int32) nbItems: nbItems

Output parameters

(double[]) Value: Value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYDualCorrectorMappingGet command which is used to Get a dual corrector mapping line. Refer to the XPS Programmer's manual to get the command description.

XYDualCorrectorMappingSet

Syntax

C# prototype

int XYDualCorrectorMappingSet(string PositionerName, Int32 LineNumber, double[] Value, Int32 nbItems, out string errstring)

Python prototype

[errstring] XYDualCorrectorMappingSet (PositionerName, LineNumber, Value, nbItems)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) LineNumber: LineNumber

(double[]) Value: Value

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYDualCorrectorMappingSet command which is used to Set a dual corrector mapping line. Refer to the XPS Programmer's manual to get the command description.

XYGantryDestressingMappingGet

Syntax

C# prototype

int XYGantryDestressingMappingGet(string PositionerName, Int32 LineNumber, out double[] Value, Int32 nbltems, out string errstring)

Python prototype

[Value, errstring] XYGantryDestressingMappingGet (PositionerName, LineNumber, nbltems)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) LineNumber: LineNumber

(Int32) nbltems: nbltems

Output parameters

(double[]) Value: Value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYGantryDestressingMappingGet command which is used to Get a gantry destressing mapping line. Refer to the XPS Programmer's manual to get the command description.

XYGantryDestressingMappingSet

Syntax

C# prototype

int XYGantryDestressingMappingSet(string PositionerName, Int32 LineNumber, double[] Value, Int32 nbItems, out string errstring)

Python prototype

[errstring] XYGantryDestressingMappingSet (PositionerName, LineNumber, Value, nbItems)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) LineNumber: LineNumber

(double[]) Value: Value

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYGantryDestressingMappingSet command which is used to Set a gantry destressing mapping line. Refer to the XPS Programmer's manual to get the command description.

XYMappingGet

Syntax

C# prototype

int XYMappingGet(string PositionerName, Int32 MappingNumber, Int32 LineNumber, out double[] Value, Int32 nbItems, out string errstring)

Python prototype

[Value, errstring] XYMappingGet (PositionerName, MappingNumber, LineNumber, nbItems)

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) MappingNumber: MappingNumber

(Int32) LineNumber: LineNumber

(Int32) nbItems: nbItems

Output parameters

(double[]) Value: Value

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYMappingGet command which is used to Get a mapping line value. Refer to the XPS Programmer's manual to get the command description.

XYMappingSet

Syntax

C# prototype

```
int XYMappingSet(string PositionerName, Int32 MappingNumber, Int32 LineNumber, double[] Value, Int32 nbItems,  
out string errstring)
```

Python prototype

```
[errstring] XYMappingSet (PositionerName, MappingNumber, LineNumber, Value, nbItems)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int32) MappingNumber: MappingNumber

(Int32) LineNumber: LineNumber

(double[]) Value: Value

(Int32) nbItems: nbItems

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYMappingSet command which is used to Set a mapping line value. Refer to the XPS Programmer's manual to get the command description.

XYScanNextGet

Syntax

C# prototype

```
int XYScanNextGet(string GroupName, out double MinStartVelocity, out double MaxStartVelocity, out double
StartCheckingTime, out double MinPositionForLaserEnable, out double MaxPositionForLaserEnable, out double
MinPositionForNotchStart, out double MaxPositionForNotchEnd, out string errstring)
```

Python prototype

```
[MinStartVelocity, MaxStartVelocity, StartCheckingTime, MinPositionForLaserEnable, MaxPositionForLaserEnable,
MinPositionForNotchStart, MaxPositionForNotchEnd, errstring] XYScanNextGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) MinStartVelocity: MinStartVelocity

(double) MaxStartVelocity: MaxStartVelocity

(double) StartCheckingTime: StartCheckingTime

(double) MinPositionForLaserEnable: MinPositionForLaserEnable

(double) MaxPositionForLaserEnable: MaxPositionForLaserEnable

(double) MinPositionForNotchStart: MinPositionForNotchStart

(double) MaxPositionForNotchEnd: MaxPositionForNotchEnd

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanNextGet command which is used to Get parameters of next scan move. Refer to the XPS Programmer's manual to get the command description.

XYScanNextSet

Syntax

C# prototype

```
int XYScanNextSet(string GroupName, double MinStartVelocity, double MaxStartVelocity, double
StartCheckingTime, double MinPositionForLaserEnable, double MaxPositionForLaserEnable, double
MinPositionForNotchStart, double MaxPositionForNotchEnd, out string errstring)
```

Python prototype

```
[errstring] XYScanNextSet (GroupName, MinStartVelocity, MaxStartVelocity, StartCheckingTime,
MinPositionForLaserEnable, MaxPositionForLaserEnable, MinPositionForNotchStart, MaxPositionForNotchEnd)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) MinStartVelocity: MinStartVelocity

(double) MaxStartVelocity: MaxStartVelocity

(double) StartCheckingTime: StartCheckingTime

(double) MinPositionForLaserEnable: MinPositionForLaserEnable

(double) MaxPositionForLaserEnable: MaxPositionForLaserEnable

(double) MinPositionForNotchStart: MinPositionForNotchStart

(double) MaxPositionForNotchEnd: MaxPositionForNotchEnd

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanNextSet command which is used to Set parameters for next scan move. Refer to the XPS Programmer's manual to get the command description.

XYScanMoveAbsolute

Syntax

C# prototype

```
int XYScanMoveAbsolute(string GroupName, double TargetPositionX, double TargetPositionY, double StartPosition,
out Int32 ScanErrorCode, out double MinimumRecordedVelocity, out double MaximumRecordedVelocity, out
double AverageScanVelocity, out string errstring)
```

Python prototype

```
[ScanErrorCode, MinimumRecordedVelocity, MaximumRecordedVelocity, AverageScanVelocity, errstring]
XYScanMoveAbsolute (GroupName, TargetPositionX, TargetPositionY, StartPosition)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) TargetPositionX: TargetPositionX

(double) TargetPositionY: TargetPositionY

(double) StartPosition: StartPosition

Output parameters

(Int32_i) ScanErrorCode: ScanErrorCode

(double) MinimumRecordedVelocity: MinimumRecordedVelocity

(double) MaximumRecordedVelocity: MaximumRecordedVelocity

(double) AverageScanVelocity: AverageScanVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanMoveAbsolute command which is used to Make a scan with absolute move. Refer to the XPS Programmer's manual to get the command description.

XYScanMoveRelative

Syntax

C# prototype

```
int XYScanMoveRelative(string GroupName, double TargetDisplacementX, double TargetDisplacementY, double
StartDisplacement, out Int32 ScanErrorCode, out double MinimumRecordedVelocity, out double
MaximumRecordedVelocity, out double AverageScanVelocity, out string errstring)
```

Python prototype

```
[ScanErrorCode, MinimumRecordedVelocity, MaximumRecordedVelocity, AverageScanVelocity, errstring]
XYScanMoveRelative (GroupName, TargetDisplacementX, TargetDisplacementY, StartDisplacement)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) TargetDisplacementX: TargetDisplacementX

(double) TargetDisplacementY: TargetDisplacementY

(double) StartDisplacement: StartDisplacement

Output parameters

(Int32_i) ScanErrorCode: ScanErrorCode

(double) MinimumRecordedVelocity: MinimumRecordedVelocity

(double) MaximumRecordedVelocity: MaximumRecordedVelocity

(double) AverageScanVelocity: AverageScanVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanMoveRelative command which is used to Make a scan with relative move. Refer to the XPS Programmer's manual to get the command description.

XYScanExecutePVT

Syntax

C# prototype

int XYScanExecutePVT(string GroupName, string FileName, Int32 ExecutionNumber, out Int32 ScanErrorCode, out double MinimumRecordedVelocity, out double MaximumRecordedVelocity, out double AverageScanVelocity, out string errstring)

Python prototype

[ScanErrorCode, MinimumRecordedVelocity, MaximumRecordedVelocity, AverageScanVelocity, errstring]
XYScanExecutePVT (GroupName, FileName, ExecutionNumber)

Parameters

Input parameters

(string) GroupName: GroupName

(string) FileName: FileName

(Int32) ExecutionNumber: ExecutionNumber

Output parameters

(Int32_i) ScanErrorCode: ScanErrorCode

(double) MinimumRecordedVelocity: MinimumRecordedVelocity

(double) MaximumRecordedVelocity: MaximumRecordedVelocity

(double) AverageScanVelocity: AverageScanVelocity

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanExecutePVT command which is used to Make a scan with PVT trajectory. Refer to the XPS Programmer's manual to get the command description.

XYScanShutterDisable

Syntax

C# prototype

```
int XYScanShutterDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYScanShutterDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanShutterDisable command which is used to Disable scan shutter. Refer to the XPS Programmer's manual to get the command description.

XYScanShutterEnable

Syntax

C# prototype

```
int XYScanShutterEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYScanShutterEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYScanShutterEnable command which is used to Enable scan shutter. Refer to the XPS Programmer's manual to get the command description.

XYShutterInterlockMonitoringDisable

Syntax

C# prototype

```
int XYShutterInterlockMonitoringDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYShutterInterlockMonitoringDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYShutterInterlockMonitoringDisable command which is used to Disable shutter interlock monitoring. Refer to the XPS Programmer's manual to get the command description.

XYShutterInterlockMonitoringEnable

Syntax

C# prototype

```
int XYShutterInterlockMonitoringEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] XYShutterInterlockMonitoringEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYShutterInterlockMonitoringEnable command which is used to Enable shutter interlock monitoring. Refer to the XPS Programmer's manual to get the command description.

XYShutterInterlockParametersGet

Syntax

C# prototype

```
int XYShutterInterlockParametersGet(string GroupName, out double IRefX, out double IRefY, out double I1, out double I2, out double I3, out double I4, out double r1, out double r2, out double r3, out double r4, out double L1, out double L2, out double L3, out double L4, out double R1, out double R2, out double R3, out double R4, out double WMP, out double WMVMin, out double WMVInit, out double SMP, out double SMVMin, out double SMVInit, out double OMP, out double OMVMin, out double OMVInit, out double mL1, out double mL2, out double mR, out string errstring)
```

Python prototype

```
[IRefX, IRefY, I1, I2, I3, I4, r1, r2, r3, r4, L1, L2, L3, L4, R1, R2, R3, R4, WMP, WMVMin, WMVInit, SMP, SMVMin, SMVInit, OMP, OMVMin, OMVInit, mL1, mL2, mR, errstring] XYShutterInterlockParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) IRefX: IRefX

(double) IRefY: IRefY

(double) I1: I1

(double) I2: I2

(double) I3: I3

(double) I4: I4

(double) r1: r1

(double) r2: r2

(double) r3: r3

(double) r4: r4

(double) L1: L1

(double) L2: L2

(double) L3: L3

(double) L4: L4

(double) R1: R1

(double) R2: R2

(double) R3: R3

(double) R4: R4

(double) WMP: WMP

(double) WMVMin: WMVMin

(double) WMVInit: WMVInit

(double) SMP: SMP

(double) SMVMin: SMVMin

(double) SMVInit: SMVInit

(double) OMP: OMP

(double) OMVMin: OMVMin

(double) OMVInit: OMVInit

(double) mL1: mL1

(double) mL2: mL2

(double) mR: mR

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYShutterInterlockParametersGet command which is used to Get shutter interlock parameters. Refer to the XPS Programmer's manual to get the command description.

XYShutterInterlockParametersSet

Syntax

C# prototype

```
int XYShutterInterlockParametersSet(string GroupName, double IRefX, double IRefY, double I1, double I2, double I3, double I4, double r1, double r2, double r3, double r4, double L1, double L2, double L3, double L4, double R1, double R2, double R3, double R4, double WMP, double WMVMin, double WMVInit, double SMP, double SMVMin, double SMVInit, double OMP, double OMVMin, double OMVInit, double mL1, double mL2, double mR, out string errstring)
```

Python prototype

```
[errstring] XYShutterInterlockParametersSet (GroupName, IRefX, IRefY, I1, I2, I3, I4, r1, r2, r3, r4, L1, L2, L3, L4, R1, R2, R3, R4, WMP, WMVMin, WMVInit, SMP, SMVMin, SMVInit, OMP, OMVMin, OMVInit, mL1, mL2, mR)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) IRefX: IRefX

(double) IRefY: IRefY

(double) I1: I1

(double) I2: I2

(double) I3: I3

(double) I4: I4

(double) r1: r1

(double) r2: r2

(double) r3: r3

(double) r4: r4

(double) L1: L1

(double) L2: L2

(double) L3: L3

(double) L4: L4

(double) R1: R1

(double) R2: R2

(double) R3: R3

(double) R4: R4

(double) WMP: WMP

(double) WMVMin: WMVMin

(double) WMVInit: WMVInit

(double) SMP: SMP

(double) SMVMin: SMVMin

(double) SMVInit: SMVInit

(double) OMP: OMP

(double) OMVMin: OMVMin

(double) OMVInit: OMVInit

(double) mL1: mL1

(double) mL2: mL2

(double) mR: mR

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYShutterInterlockParametersSet command which is used to Set shutter interlock parameters. Refer to the XPS Programmer's manual to get the command description.

XYZGroupPositionCorrectedProfilerGet

Syntax

C# prototype

```
int XYZGroupPositionCorrectedProfilerGet(string GroupName, double PositionX, double PositionY, double PositionZ, out double CorrectedProfilerPositionX, out double CorrectedProfilerPositionY, out double CorrectedProfilerPositionZ, out string errstring)
```

Python prototype

```
[CorrectedProfilerPositionX, CorrectedProfilerPositionY, CorrectedProfilerPositionZ, errstring]  
XYZGroupPositionCorrectedProfilerGet (GroupName, PositionX, PositionY, PositionZ)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) PositionX: PositionX

(double) PositionY: PositionY

(double) PositionZ: PositionZ

Output parameters

(double) CorrectedProfilerPositionX: CorrectedProfilerPositionX

(double) CorrectedProfilerPositionY: CorrectedProfilerPositionY

(double) CorrectedProfilerPositionZ: CorrectedProfilerPositionZ

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZGroupPositionCorrectedProfilerGet command which is used to Return corrected profiler positions. Refer to the XPS Programmer's manual to get the command description.

XYZGroupPositionPCORawEncoderGet

Syntax

C# prototype

```
int XYZGroupPositionPCORawEncoderGet(string GroupName, double PositionX, double PositionY, double PositionZ, out double PCORawPositionX, out double PCORawPositionY, out double PCORawPositionZ, out string errstring)
```

Python prototype

```
[PCORawPositionX, PCORawPositionY, PCORawPositionZ, errstring] XYZGroupPositionPCORawEncoderGet (GroupName, PositionX, PositionY, PositionZ)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) PositionX: PositionX

(double) PositionY: PositionY

(double) PositionZ: PositionZ

Output parameters

(double) PCORawPositionX: PCORawPositionX

(double) PCORawPositionY: PCORawPositionY

(double) PCORawPositionZ: PCORawPositionZ

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous XYZGroupPositionPCORawEncoderGet command which is used to Return PCO raw encoder positions. Refer to the XPS Programmer's manual to get the command description.

HexapodPositionCurrentGet

Syntax

C# prototype

```
int HexapodPositionCurrentGet(string GroupName, out double[] CurrentPosition, Int32 nbItems, out string errstring)
```

Python prototype

```
[CurrentPosition, errstring] HexapodPositionCurrentGet (GroupName, nbItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbItems: nbItems

Output parameters

(double[]) CurrentPosition: CurrentPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodPositionCurrentGet command which is used to Return current hexapod position tool in work. Refer to the XPS Programmer's manual to get the command description.

HexapodPositionSetpointGet

Syntax

C# prototype

```
int HexapodPositionSetpointGet(string GroupName, out double[] SetPointPosition, Int32 nblItems, out string errstring)
```

Python prototype

```
[SetPointPosition, errstring] HexapodPositionSetpointGet (GroupName, nblItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nblItems: nblItems

Output parameters

(double[]) SetPointPosition: SetPointPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodPositionSetpointGet command which is used to Return setpoint hexapod position tool in work. Refer to the XPS Programmer's manual to get the command description.

HexapodPositionTargetGet

Syntax

C# prototype

```
int HexapodPositionTargetGet(string GroupName, out double[] TargetPosition, Int32 nbltems, out string errstring)
```

Python prototype

```
[TargetPosition, errstring] HexapodPositionTargetGet (GroupName, nbltems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nbltems: nbltems

Output parameters

(double[]) TargetPosition: TargetPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodPositionTargetGet command which is used to Return target hexapod position tool in work. Refer to the XPS Programmer's manual to get the command description.

HexapodMoveAbsolute

Syntax

C# prototype

```
int HexapodMoveAbsolute(string GroupName, string CoordinateSystem, double X, double Y, double Z, double U, double V, double W, out string errstring)
```

Python prototype

```
[errstring] HexapodMoveAbsolute (GroupName, CoordinateSystem, X, Y, Z, U, V, W)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(double) X: X

(double) Y: Y

(double) Z: Z

(double) U: U

(double) V: V

(double) W: W

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodMoveAbsolute command which is used to Hexapod absolute move in a specific coordinate system. Refer to the XPS Programmer's manual to get the command description.

HexapodMoveIncremental

Syntax

C# prototype

```
int HexapodMoveIncremental(string GroupName, string CoordinateSystem, double dX, double dY, double dZ,
double dU, double dV, double dW, out string errstring)
```

Python prototype

```
[errstring] HexapodMoveIncremental (GroupName, CoordinateSystem, dX, dY, dZ, dU, dV, dW)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

(double) dU: dU

(double) dV: dV

(double) dW: dW

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodMoveIncremental command which is used to Hexapod incremental move in a specific coordinate system. Refer to the XPS Programmer's manual to get the command description.

HexapodCoordinatesGet

Syntax

C# prototype

```
int HexapodCoordinatesGet(string GroupName, string CoordinateSystemIn, string CoordinateSystemOut, double
Xin, double Yin, double Zin, double Uin, double Vin, double Win, out double Xout, out double Yout, out double
Zout, out double Uout, out double Vout, out double Wout, out string errstring)
```

Python prototype

```
[Xout, Yout, Zout, Uout, Vout, Wout, errstring] HexapodCoordinatesGet (GroupName, CoordinateSystemIn,
CoordinateSystemOut, Xin, Yin, Zin, Uin, Vin, Win)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystemIn: CoordinateSystemIn

(string) CoordinateSystemOut: CoordinateSystemOut

(double) Xin: Xin

(double) Yin: Yin

(double) Zin: Zin

(double) Uin: Uin

(double) Vin: Vin

(double) Win: Win

Output parameters

(double) Xout: Xout

(double) Yout: Yout

(double) Zout: Zout

(double) Uout: Uout

(double) Vout: Vout

(double) Wout: Wout

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodCoordinatesGet command which is used to Get coordinates in a specific coordinate system of a point specified in another coordinate system. Refer to the XPS Programmer's manual to get the command description.

HexapodCoordinateSystemSet

Syntax

C# prototype

```
int HexapodCoordinateSystemSet(string GroupName, string CoordinateSystem, double X, double Y, double Z, double U, double V, double W, out string errstring)
```

Python prototype

```
[errstring] HexapodCoordinateSystemSet (GroupName, CoordinateSystem, X, Y, Z, U, V, W)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(double) X: X

(double) Y: Y

(double) Z: Z

(double) U: U

(double) V: V

(double) W: W

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodCoordinateSystemSet command which is used to Modify the position of a coordinate system. Refer to the XPS Programmer's manual to get the command description.

HexapodCoordinateSystemGet

Syntax

C# prototype

```
int HexapodCoordinateSystemGet(string GroupName, string CoordinateSystem, out double X, out double Y, out double Z, out double U, out double V, out double W, out string errstring)
```

Python prototype

```
[X, Y, Z, U, V, W, errstring] HexapodCoordinateSystemGet (GroupName, CoordinateSystem)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

Output parameters

(double) X: X

(double) Y: Y

(double) Z: Z

(double) U: U

(double) V: V

(double) W: W

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodCoordinateSystemGet command which is used to Get the position of a coordinate system. Refer to the XPS Programmer's manual to get the command description.

HexapodMoveIncrementalControl

Syntax

C# prototype

```
int HexapodMoveIncrementalControl(string GroupName, string CoordinateSystem, string HexapodTrajectoryType,
double dX, double dY, double dZ, out string errstring)
```

Python prototype

```
[errstring] HexapodMoveIncrementalControl (GroupName, CoordinateSystem, HexapodTrajectoryType, dX, dY, dZ)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(string) HexapodTrajectoryType: HexapodTrajectoryType

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodMoveIncrementalControl command which is used to Hexapod trajectory . Refer to the XPS Programmer's manual to get the command description.

HexapodMoveIncrementalControlWithTargetVelocity

Syntax

C# prototype

```
int HexapodMoveIncrementalControlWithTargetVelocity(string GroupName, string CoordinateSystem, string
HexapodTrajectoryType, double dX, double dY, double dZ, double Velocity, out string errstring)
```

Python prototype

```
[errstring] HexapodMoveIncrementalControlWithTargetVelocity (GroupName, CoordinateSystem,
HexapodTrajectoryType, dX, dY, dZ, Velocity)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(string) HexapodTrajectoryType: HexapodTrajectoryType

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

(double) Velocity: Velocity

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodMoveIncrementalControlWithTargetVelocity command which is used to Hexapod trajectory . Refer to the XPS Programmer's manual to get the command description.

HexapodMoveIncrementalControlPulseAndGatheringSet

Syntax

C# prototype

```
int HexapodMoveIncrementalControlPulseAndGatheringSet(string GroupName, Int32 Divisor, out string errstring)
```

Python prototype

```
[errstring] HexapodMoveIncrementalControlPulseAndGatheringSet (GroupName, Divisor)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Divisor: Divisor

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodMoveIncrementalControlPulseAndGatheringSet command which is used to Configure gathering with pulses : gathered data are X, Y, Z, U, V, W and pulses will be generated during only constant velocity. Refer to the XPS Programmer's manual to get the command description.

HexapodMoveIncrementalControlLimitGet

Syntax

C# prototype

```
int HexapodMoveIncrementalControlLimitGet(string GroupName, string CoordinateSystem, string
HexapodTrajectoryType, double dX, double dY, double dZ, out double MaximumVelocityCarriage, out double
TrajectoryPercent, out string errstring)
```

Python prototype

```
[MaximumVelocityCarriage, TrajectoryPercent, errstring] HexapodMoveIncrementalControlLimitGet (GroupName,
CoordinateSystem, HexapodTrajectoryType, dX, dY, dZ)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) CoordinateSystem: CoordinateSystem

(string) HexapodTrajectoryType: HexapodTrajectoryType

(double) dX: dX

(double) dY: dY

(double) dZ: dZ

Output parameters

(double) MaximumVelocityCarriage: MaximumVelocityCarriage

(double) TrajectoryPercent: TrajectoryPercent

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodMoveIncrementalControlLimitGet command which is used to Returns the maximum velocity of carriage and the percent of the trajectory executable. Refer to the XPS Programmer's manual to get the command description.

HexapodSGammaParametersDistanceGet

Syntax

C# prototype

```
int HexapodSGammaParametersDistanceGet(string PositionerName, double Displacement, double Velocity, double Acceleration, double MinJerkTime, double MaxJerkTime, out double DisplacementDuringAcc, out double DisplacementDuringVel, out string errstring)
```

Python prototype

```
[DisplacementDuringAcc, DisplacementDuringVel, errstring] HexapodSGammaParametersDistanceGet (PositionerName, Displacement, Velocity, Acceleration, MinJerkTime, MaxJerkTime)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) Displacement: Displacement

(double) Velocity: Velocity

(double) Acceleration: Acceleration

(double) MinJerkTime: MinJerkTime

(double) MaxJerkTime: MaxJerkTime

Output parameters

(double) DisplacementDuringAcc: DisplacementDuringAcc

(double) DisplacementDuringVel: DisplacementDuringVel

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HexapodSGammaParametersDistanceGet command which is used to Returns distance during acceleration phase and distance during constant velocity phase for a virtual SGamma profiler. Refer to the XPS Programmer's manual to get the command description.

MaskClampDisable

Syntax

C# prototype

```
int MaskClampDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] MaskClampDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MaskClampDisable command which is used to Disable Mask clamp. Refer to the XPS Programmer's manual to get the command description.

MaskClampEnable

Syntax

C# prototype

```
int MaskClampEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] MaskClampEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MaskClampEnable command which is used to Enable Mask clamp. Refer to the XPS Programmer's manual to get the command description.

MaskClampStateGet

Syntax

C# prototype

```
int MaskClampStateGet(string GroupName, out bool Status, out string errstring)
```

Python prototype

```
[Status, errstring] MaskClampStateGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(bool) Status: Status

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous MaskClampStateGet command which is used to Get Mask clamp state. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaClampDisable

Syntax

C# prototype

```
int SingleAxisThetaClampDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaClampDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaClampDisable command which is used to Set clamping disable on selected group. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaClampEnable

Syntax

C# prototype

```
int SingleAxisThetaClampEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaClampEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaClampEnable command which is used to Set clamping enable on selected group. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaFeedforwardParametersGet

Syntax

C# prototype

```
int SingleAxisThetaFeedforwardParametersGet(string GroupName, out double KFeedforwardX, out double KFeedforwardY, out string errstring)
```

Python prototype

```
[KFeedforwardX, KFeedforwardY, errstring] SingleAxisThetaFeedforwardParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) KFeedforwardX: KFeedforwardX

(double) KFeedforwardY: KFeedforwardY

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaFeedforwardParametersGet command which is used to Get XY to Theta feedforward gains. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaFeedforwardParametersSet

Syntax

C# prototype

```
int SingleAxisThetaFeedforwardParametersSet(string GroupName, double KFeedforwardX, double KFeedforwardY,
out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaFeedforwardParametersSet (GroupName, KFeedforwardX, KFeedforwardY)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) KFeedforwardX: KFeedforwardX

(double) KFeedforwardY: KFeedforwardY

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaFeedforwardParametersSet command which is used to Set XY to Theta feedforward gains. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaFeedforwardJerkParametersGet

Syntax

C# prototype

```
int SingleAxisThetaFeedforwardJerkParametersGet(string GroupName, out double KFeedforwardJerkX, out double KFeedforwardJerkY, out string errstring)
```

Python prototype

```
[KFeedforwardJerkX, KFeedforwardJerkY, errstring] SingleAxisThetaFeedforwardJerkParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) KFeedforwardJerkX: KFeedforwardJerkX

(double) KFeedforwardJerkY: KFeedforwardJerkY

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaFeedforwardJerkParametersGet command which is used to Get XY to Theta feedforward acceleration gains. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaFeedforwardJerkParametersSet

Syntax

C# prototype

```
int SingleAxisThetaFeedforwardJerkParametersSet(string GroupName, double KFeedforwardJerkX, double
KFeedforwardJerkY, out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaFeedforwardJerkParametersSet (GroupName, KFeedforwardJerkX, KFeedforwardJerkY)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) KFeedforwardJerkX: KFeedforwardJerkX

(double) KFeedforwardJerkY: KFeedforwardJerkY

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaFeedforwardJerkParametersSet command which is used to Set XY to Theta feedforward jerk gains. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaSlaveModeEnable

Syntax

C# prototype

```
int SingleAxisThetaSlaveModeEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaSlaveModeEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaSlaveModeEnable command which is used to Enable the slave mode. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaSlaveModeDisable

Syntax

C# prototype

```
int SingleAxisThetaSlaveModeDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaSlaveModeDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaSlaveModeDisable command which is used to Disable the slave mode. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaSlaveParametersGet

Syntax

C# prototype

```
int SingleAxisThetaSlaveParametersGet(string GroupName, out string PositionerName, out double Ratio, out string errstring)
```

Python prototype

```
[PositionerName, Ratio, errstring] SingleAxisThetaSlaveParametersGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) PositionerName: PositionerName

(double) Ratio: Ratio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaSlaveParametersGet command which is used to Get slave parameters. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaSlaveParametersSet

Syntax

C# prototype

```
int SingleAxisThetaSlaveParametersSet(string GroupName, string PositionerName, double Ratio, out string errstring)
```

Python prototype

```
[errstring] SingleAxisThetaSlaveParametersSet (GroupName, PositionerName, Ratio)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) PositionerName: PositionerName

(double) Ratio: Ratio

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaSlaveParametersSet command which is used to Set slave parameters. Refer to the XPS Programmer's manual to get the command description.

TZMotorDecouplingMatrixGet

Syntax

C# prototype

```
int TZMotorDecouplingMatrixGet(string GroupName, out double Value11, out double Value12, out double Value13,
out double Value21, out double Value22, out double Value23, out double Value31, out double Value32, out double
Value33, out string errstring)
```

Python prototype

```
[Value11, Value12, Value13, Value21, Value22, Value23, Value31, Value32, Value33, errstring]
TZMotorDecouplingMatrixGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) Value11: Value11

(double) Value12: Value12

(double) Value13: Value13

(double) Value21: Value21

(double) Value22: Value22

(double) Value23: Value23

(double) Value31: Value31

(double) Value32: Value32

(double) Value33: Value33

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZMotorDecouplingMatrixGet command which is used to Get TZ motor decoupling matrix. Refer to the XPS Programmer's manual to get the command description.

TZMotorDecouplingMatrixSet

Syntax

C# prototype

```
int TZMotorDecouplingMatrixSet(string GroupName, double Value11, double Value12, double Value13, double Value21, double Value22, double Value23, double Value31, double Value32, double Value33, out string errstring)
```

Python prototype

```
[errstring] TZMotorDecouplingMatrixSet (GroupName, Value11, Value12, Value13, Value21, Value22, Value23, Value31, Value32, Value33)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) Value11: Value11

(double) Value12: Value12

(double) Value13: Value13

(double) Value21: Value21

(double) Value22: Value22

(double) Value23: Value23

(double) Value31: Value31

(double) Value32: Value32

(double) Value33: Value33

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZMotorDecouplingMatrixSet command which is used to Set TZ motor decoupling matrix. Refer to the XPS Programmer's manual to get the command description.

TZMotorDecouplingModeGet

Syntax

C# prototype

```
int TZMotorDecouplingModeGet(string GroupName, out Int32 Mode, out string errstring)
```

Python prototype

```
[Mode, errstring] TZMotorDecouplingModeGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) Mode: Mode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZMotorDecouplingModeGet command which is used to Get TZ motor decoupling mode. Refer to the XPS Programmer's manual to get the command description.

TZMotorDecouplingModeSet

Syntax

C# prototype

int TZMotorDecouplingModeSet(string GroupName, Int32 Mode, out string errstring)

Python prototype

[errstring] TZMotorDecouplingModeSet (GroupName, Mode)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Mode: Mode

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZMotorDecouplingModeSet command which is used to Set TZ motor decoupling mode. Refer to the XPS Programmer's manual to get the command description.

TZEncoderCouplingMatrixGet

Syntax

C# prototype

```
int TZEncoderCouplingMatrixGet(string GroupName, out double Value11, out double Value12, out double Value13,
out double Value21, out double Value22, out double Value23, out double Value31, out double Value32, out double
Value33, out string errstring)
```

Python prototype

```
[Value11, Value12, Value13, Value21, Value22, Value23, Value31, Value32, Value33, errstring]
TZEncoderCouplingMatrixGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) Value11: Value11

(double) Value12: Value12

(double) Value13: Value13

(double) Value21: Value21

(double) Value22: Value22

(double) Value23: Value23

(double) Value31: Value31

(double) Value32: Value32

(double) Value33: Value33

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZEncoderCouplingMatrixGet command which is used to Get TZ encoder coupling matrix. Refer to the XPS Programmer's manual to get the command description.

TZEncoderCouplingMatrixSet

Syntax

C# prototype

```
int TZEncoderCouplingMatrixSet(string GroupName, double Value11, double Value12, double Value13, double Value21, double Value22, double Value23, double Value31, double Value32, double Value33, out string errstring)
```

Python prototype

```
[errstring] TZEncoderCouplingMatrixSet (GroupName, Value11, Value12, Value13, Value21, Value22, Value23, Value31, Value32, Value33)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) Value11: Value11

(double) Value12: Value12

(double) Value13: Value13

(double) Value21: Value21

(double) Value22: Value22

(double) Value23: Value23

(double) Value31: Value31

(double) Value32: Value32

(double) Value33: Value33

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZEncoderCouplingMatrixSet command which is used to Set TZ encoder coupling matrix. Refer to the XPS Programmer's manual to get the command description.

TZEncoderCouplingModeGet

Syntax

C# prototype

int TZEncoderCouplingModeGet(string GroupName, out Int32 Mode, out string errstring)

Python prototype

[Mode, errstring] TZEncoderCouplingModeGet (GroupName)

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) Mode: Mode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZEncoderCouplingModeGet command which is used to Get TZ encoder coupling mode. Refer to the XPS Programmer's manual to get the command description.

TZEncoderCouplingModeSet

Syntax

C# prototype

```
int TZEncoderCouplingModeSet(string GroupName, Int32 Mode, out string errstring)
```

Python prototype

```
[errstring] TZEncoderCouplingModeSet (GroupName, Mode)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Mode: Mode

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZEncoderCouplingModeSet command which is used to Set TZ encoder coupling mode. Refer to the XPS Programmer's manual to get the command description.

TZMappingModeGet

Syntax

C# prototype

```
int TZMappingModeGet(string GroupName, out Int32 Mode, out string errstring)
```

Python prototype

```
[Mode, errstring] TZMappingModeGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(Int32_i) Mode: Mode

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZMappingModeGet command which is used to Get TZ mapping mode. Refer to the XPS Programmer's manual to get the command description.

TZMappingModeSet

Syntax

C# prototype

int TZMappingModeSet(string GroupName, Int32 Mode, out string errstring)

Python prototype

[errstring] TZMappingModeSet (GroupName, Mode)

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) Mode: Mode

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZMappingModeSet command which is used to Set TZ mapping mode. Refer to the XPS Programmer's manual to get the command description.

TZTrackingCutOffFrequencyGet

Syntax

C# prototype

```
int TZTrackingCutOffFrequencyGet(string GroupName, out double CutOffFrequencyZ1, out double  
CutOffFrequencyZ2, out double CutOffFrequencyZ3, out string errstring)
```

Python prototype

```
[CutOffFrequencyZ1, CutOffFrequencyZ2, CutOffFrequencyZ3, errstring] TZTrackingCutOffFrequencyGet  
(GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) CutOffFrequencyZ1: CutOffFrequencyZ1

(double) CutOffFrequencyZ2: CutOffFrequencyZ2

(double) CutOffFrequencyZ3: CutOffFrequencyZ3

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZTrackingCutOffFrequencyGet command which is used to Get cut-off frequency for tracking control. Refer to the XPS Programmer's manual to get the command description.

TZTrackingCutOffFrequencySet

Syntax

C# prototype

```
int TZTrackingCutOffFrequencySet(string GroupName, double CutOffFrequencyZ1, double CutOffFrequencyZ2,  
double CutOffFrequencyZ3, out string errstring)
```

Python prototype

```
[errstring] TZTrackingCutOffFrequencySet (GroupName, CutOffFrequencyZ1, CutOffFrequencyZ2,  
CutOffFrequencyZ3)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) CutOffFrequencyZ1: CutOffFrequencyZ1

(double) CutOffFrequencyZ2: CutOffFrequencyZ2

(double) CutOffFrequencyZ3: CutOffFrequencyZ3

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZTrackingCutOffFrequencySet command which is used to Set cut-off frequency for tracking control. Refer to the XPS Programmer's manual to get the command description.

TZFocusModeEnable

Syntax

C# prototype

```
int TZFocusModeEnable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] TZFocusModeEnable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZFocusModeEnable command which is used to Enable the focus mode. Refer to the XPS Programmer's manual to get the command description.

TZFocusModeDisable

Syntax

C# prototype

```
int TZFocusModeDisable(string GroupName, out string errstring)
```

Python prototype

```
[errstring] TZFocusModeDisable (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZFocusModeDisable command which is used to Disable the focus mode. Refer to the XPS Programmer's manual to get the command description.

TZTrackingUserMaximumZZZTargetDifferenceGet

Syntax

C# prototype

```
int TZTrackingUserMaximumZZZTargetDifferenceGet(string GroupName, out double  
UserMaximumZZZTargetDifference, out string errstring)
```

Python prototype

```
[UserMaximumZZZTargetDifference, errstring] TZTrackingUserMaximumZZZTargetDifferenceGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) UserMaximumZZZTargetDifference: UserMaximumZZZTargetDifference

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZTrackingUserMaximumZZZTargetDifferenceGet command which is used to Get user maximum ZZZ target difference. Refer to the XPS Programmer's manual to get the command description.

TZTrackingUserMaximumZZZTargetDifferenceSet

Syntax

C# prototype

```
int TZTrackingUserMaximumZZZTargetDifferenceSet(string GroupName, double
UserMaximumZZZTargetDifference, out string errstring)
```

Python prototype

```
[errstring] TZTrackingUserMaximumZZZTargetDifferenceSet (GroupName, UserMaximumZZZTargetDifference)
```

Parameters

Input parameters

(string) GroupName: GroupName

(double) UserMaximumZZZTargetDifference: UserMaximumZZZTargetDifference

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TZTrackingUserMaximumZZZTargetDifferenceSet command which is used to Set user maximum ZZZ target difference. Refer to the XPS Programmer's manual to get the command description.

ExternalModuleSocketReserve

Syntax

C# prototype

```
int ExternalModuleSocketReserve(Int32 ModuleNumber, out string errstring)
```

Python prototype

```
[errstring] ExternalModuleSocketReserve (ModuleNumber)
```

Parameters

Input parameters

(Int32) ModuleNumber: ModuleNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalModuleSocketReserve command which is used to Reserve socket for external module. Refer to the XPS Programmer's manual to get the command description.

ExternalModuleSocketFree

Syntax

C# prototype

```
int ExternalModuleSocketFree(Int32 ModuleNumber, out string errstring)
```

Python prototype

```
[errstring] ExternalModuleSocketFree (ModuleNumber)
```

Parameters

Input parameters

(Int32) ModuleNumber: ModuleNumber

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalModuleSocketFree command which is used to Free socket reserved for external module. Refer to the XPS Programmer's manual to get the command description.

ExternalModuleFirmwareVersionGet

Syntax

C# prototype

```
int ExternalModuleFirmwareVersionGet(Int32 ModuleNumber, out string Version, out string errstring)
```

Python prototype

```
[Version, errstring] ExternalModuleFirmwareVersionGet (ModuleNumber)
```

Parameters

Input parameters

(Int32) ModuleNumber: ModuleNumber

Output parameters

(string) Version: Version

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalModuleFirmwareVersionGet command which is used to Get external module version. Refer to the XPS Programmer's manual to get the command description.

ExternalModuleErrorStringGet

Syntax

C# prototype

```
int ExternalModuleErrorStringGet(Int32 ModuleNumber, Int32 Error, out string ErrorDescription, out string errstring)
```

Python prototype

```
[ErrorDescription, errstring] ExternalModuleErrorStringGet (ModuleNumber, Error)
```

Parameters

Input parameters

(Int32) ModuleNumber: ModuleNumber

(Int32) Error: Error

Output parameters

(string) ErrorDescription: ErrorDescription

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalModuleErrorStringGet command which is used to Get external module error description. Refer to the XPS Programmer's manual to get the command description.

ExternalModuleISRTIME DurationsGet

Syntax

C# prototype

```
int ExternalModuleISRTIME DurationsGet(Int32 ModuleNumber, out double CurrentCorrectorISRDURATION, out double MaximumCorrectorISRDURATION, out double CurrentProfilerISRDURATION, out double MaximumProfilerISRDURATION, out string errstring)
```

Python prototype

```
[CurrentCorrectorISRDURATION, MaximumCorrectorISRDURATION, CurrentProfilerISRDURATION, MaximumProfilerISRDURATION, errstring] ExternalModuleISRTIME DurationsGet (ModuleNumber)
```

Parameters

Input parameters

(Int32) ModuleNumber: ModuleNumber

Output parameters

(double) CurrentCorrectorISRDURATION: CurrentCorrectorISRDURATION

(double) MaximumCorrectorISRDURATION: MaximumCorrectorISRDURATION

(double) CurrentProfilerISRDURATION: CurrentProfilerISRDURATION

(double) MaximumProfilerISRDURATION: MaximumProfilerISRDURATION

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ExternalModuleISRTIME DurationsGet command which is used to Get external module scan durations. Refer to the XPS Programmer's manual to get the command description.

LoginCheck

Syntax

C# prototype

```
int LoginCheck(string Name, string Password, out string errstring)
```

Python prototype

```
[errstring] LoginCheck (Name, Password)
```

Parameters

Input parameters

(string) Name: Name

(string) Password: Password

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous LoginCheck command which is used to Check the Login Parameters. Refer to the XPS Programmer's manual to get the command description.

LoginSCheck

Syntax

C# prototype

```
int LoginSCheck(string Name, string CryptedPassword, out string errstring)
```

Python prototype

```
[errstring] LoginSCheck (Name, CryptedPassword)
```

Parameters

Input parameters

(string) Name: Name

(string) CryptedPassword: CryptedPassword

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous LoginSCheck command which is used to Check the Login Parameters . Refer to the XPS Programmer's manual to get the command description.

GroupAllPositionTrace

Syntax

C# prototype

```
int GroupAllPositionTrace(string GroupName, string TitleTrace, out string errstring)
```

Python prototype

```
[errstring] GroupAllPositionTrace (GroupName, TitleTrace)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TitleTrace: TitleTrace

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupAllPositionTrace command which is used to Trace all position in TraceAllPositions.log. Refer to the XPS Programmer's manual to get the command description.

GroupMotorMatrixTrace

Syntax

C# prototype

```
int GroupMotorMatrixTrace(string GroupName, string TitleTrace, out string errstring)
```

Python prototype

```
[errstring] GroupMotorMatrixTrace (GroupName, TitleTrace)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TitleTrace: TitleTrace

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMotorMatrixTrace command which is used to Trace all position in TraceMotorMatrix.log. Refer to the XPS Programmer's manual to get the command description.

GroupMotorMatrixInverseTrace

Syntax

C# prototype

```
int GroupMotorMatrixInverseTrace(string GroupName, string TitleTrace, out string errstring)
```

Python prototype

```
[errstring] GroupMotorMatrixInverseTrace (GroupName, TitleTrace)
```

Parameters

Input parameters

(string) GroupName: GroupName

(string) TitleTrace: TitleTrace

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupMotorMatrixInverseTrace command which is used to Trace all position in TraceMotorMatrixInverse.log. Refer to the XPS Programmer's manual to get the command description.

GroupPositionCurrentRawGet

Syntax

C# prototype

```
int GroupPositionCurrentRawGet(string GroupName, out double[] RawCurrentEncoderPosition, Int32 nblItems, out string errstring)
```

Python prototype

```
[RawCurrentEncoderPosition, errstring] GroupPositionCurrentRawGet (GroupName, nblItems)
```

Parameters

Input parameters

(string) GroupName: GroupName

(Int32) nblItems: nblItems

Output parameters

(double[]) RawCurrentEncoderPosition: RawCurrentEncoderPosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupPositionCurrentRawGet command which is used to Return raw current positions. Refer to the XPS Programmer's manual to get the command description.

PositionerMotorOutputOffsetGet

Syntax

C# prototype

```
int PositionerMotorOutputOffsetGet(string PositionerName, out double PrimaryDAC1, out double PrimaryDAC2,
out double PrimaryDACDifferentialGain, out double SecondaryDAC1, out double SecondaryDAC2, out double
SecondaryDACDifferentialGain, out string errstring)
```

Python prototype

```
[PrimaryDAC1, PrimaryDAC2, PrimaryDACDifferentialGain, SecondaryDAC1, SecondaryDAC2,
SecondaryDACDifferentialGain, errstring] PositionerMotorOutputOffsetGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) PrimaryDAC1: PrimaryDAC1

(double) PrimaryDAC2: PrimaryDAC2

(double) PrimaryDACDifferentialGain: PrimaryDACDifferentialGain

(double) SecondaryDAC1: SecondaryDAC1

(double) SecondaryDAC2: SecondaryDAC2

(double) SecondaryDACDifferentialGain: SecondaryDACDifferentialGain

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotorOutputOffsetGet command which is used to Get soft . Refer to the XPS Programmer's manual to get the command description.

PositionerMotorOutputOffsetSet

Syntax

C# prototype

```
int PositionerMotorOutputOffsetSet(string PositionerName, double PrimaryDAC1, double PrimaryDAC2, double
PrimaryDACDifferentialGain, double SecondaryDAC1, double SecondaryDAC2, double
SecondaryDACDifferentialGain, out string errstring)
```

Python prototype

```
[errstring] PositionerMotorOutputOffsetSet (PositionerName, PrimaryDAC1, PrimaryDAC2,
PrimaryDACDifferentialGain, SecondaryDAC1, SecondaryDAC2, SecondaryDACDifferentialGain)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(double) PrimaryDAC1: PrimaryDAC1

(double) PrimaryDAC2: PrimaryDAC2

(double) PrimaryDACDifferentialGain: PrimaryDACDifferentialGain

(double) SecondaryDAC1: SecondaryDAC1

(double) SecondaryDAC2: SecondaryDAC2

(double) SecondaryDACDifferentialGain: SecondaryDACDifferentialGain

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotorOutputOffsetSet command which is used to Set soft . Refer to the XPS Programmer's manual to get the command description.

PositionerMotorReferencePositionGet

Syntax

C# prototype

```
int PositionerMotorReferencePositionGet(string PositionerName, out double ReferencePosition, out string errstring)
```

Python prototype

```
[ReferencePosition, errstring] PositionerMotorReferencePositionGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) ReferencePosition: ReferencePosition

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerMotorReferencePositionGet command which is used to Get linear motor reference position. Refer to the XPS Programmer's manual to get the command description.

PositionerEncoderAbsoluteHardConfigurationGet

Syntax

C# prototype

```
int PositionerEncoderAbsoluteHardConfigurationGet(string PositionerName, out double Configuration, out string errstring)
```

Python prototype

```
[Configuration, errstring] PositionerEncoderAbsoluteHardConfigurationGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(double) Configuration: Configuration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerEncoderAbsoluteHardConfigurationGet command which is used to Get absolute encoder hardware parameters. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaPositionRawGet

Syntax

C# prototype

```
int SingleAxisThetaPositionRawGet(string GroupName, out double RawEncoderPosition1, out double RawEncoderPosition2, out double RawEncoderPosition3, out string errstring)
```

Python prototype

```
[RawEncoderPosition1, RawEncoderPosition2, RawEncoderPosition3, errstring] SingleAxisThetaPositionRawGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) RawEncoderPosition1: RawEncoderPosition1

(double) RawEncoderPosition2: RawEncoderPosition2

(double) RawEncoderPosition3: RawEncoderPosition3

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaPositionRawGet command which is used to Get raw encoder positions for single axis theta encoder. Refer to the XPS Programmer's manual to get the command description.

SingleAxisThetaPositionRawCorrectedGet

Syntax

C# prototype

```
int SingleAxisThetaPositionRawCorrectedGet(string GroupName, out double RawCorrectedEncoderPosition1, out double RawCorrectedEncoderPosition2, out double RawCorrectedEncoderPosition3, out string errstring)
```

Python prototype

```
[RawCorrectedEncoderPosition1, RawCorrectedEncoderPosition2, RawCorrectedEncoderPosition3, errstring]  
SingleAxisThetaPositionRawCorrectedGet (GroupName)
```

Parameters

Input parameters

(string) GroupName: GroupName

Output parameters

(double) RawCorrectedEncoderPosition1: RawCorrectedEncoderPosition1

(double) RawCorrectedEncoderPosition2: RawCorrectedEncoderPosition2

(double) RawCorrectedEncoderPosition3: RawCorrectedEncoderPosition3

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SingleAxisThetaPositionRawCorrectedGet command which is used to Get raw corrected encoder positions for single axis theta encoder. Refer to the XPS Programmer's manual to get the command description.

EEPROMCIESet

Syntax

C# prototype

```
int EEPROMCIESet(Int32 CardNumber, string ReferenceString, out string errstring)
```

Python prototype

```
[errstring] EEPROMCIESet (CardNumber, ReferenceString)
```

Parameters

Input parameters

(Int32) CardNumber: CardNumber

(string) ReferenceString: ReferenceString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EEPROMCIESet command which is used to . Refer to the XPS Programmer's manual to get the command description.

EEPROMDACOffsetCIESet

Syntax

C# prototype

int EEPROMDACOffsetCIESet(Int32 PlugNumber, double DAC1Offset, double DAC2Offset, out string errstring)

Python prototype

[errstring] EEPROMDACOffsetCIESet (PlugNumber, DAC1Offset, DAC2Offset)

Parameters

Input parameters

(Int32) PlugNumber: PlugNumber

(double) DAC1Offset: DAC1Offset

(double) DAC2Offset: DAC2Offset

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EEPROMDACOffsetCIESet command which is used to . Refer to the XPS Programmer's manual to get the command description.

EEPROMDriverSet

Syntax

C# prototype

```
int EEPROMDriverSet(Int32 PlugNumber, string ReferenceString, out string errstring)
```

Python prototype

```
[errstring] EEPROMDriverSet (PlugNumber, ReferenceString)
```

Parameters

Input parameters

(Int32) PlugNumber: PlugNumber

(string) ReferenceString: ReferenceString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EEPROMDriverSet command which is used to . Refer to the XPS Programmer's manual to get the command description.

EEPROMINTSet

Syntax

C# prototype

int EEPROMINTSet(Int32 CardNumber, string ReferenceString, out string errstring)

Python prototype

[errstring] EEPROMINTSet (CardNumber, ReferenceString)

Parameters

Input parameters

(Int32) CardNumber: CardNumber

(string) ReferenceString: ReferenceString

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EEPROMINTSet command which is used to . Refer to the XPS Programmer's manual to get the command description.

CPUCoreAndBoardSupplyVoltagesGet

Syntax

C# prototype

```
int CPUCoreAndBoardSupplyVoltagesGet(out double VoltageCPUCore, out double SupplyVoltage1P5V, out double
SupplyVoltage3P3V, out double SupplyVoltage5V, out double SupplyVoltage12V, out double SupplyVoltageM12V,
out double SupplyVoltageM5V, out double SupplyVoltage5VSB, out string errstring)
```

Python prototype

```
[VoltageCPUCore, SupplyVoltage1P5V, SupplyVoltage3P3V, SupplyVoltage5V, SupplyVoltage12V,
SupplyVoltageM12V, SupplyVoltageM5V, SupplyVoltage5VSB, errstring] CPUCoreAndBoardSupplyVoltagesGet ()
```

Parameters

Input parameters

None

Output parameters

(double) VoltageCPUCore: VoltageCPUCore

(double) SupplyVoltage1P5V: SupplyVoltage1P5V

(double) SupplyVoltage3P3V: SupplyVoltage3P3V

(double) SupplyVoltage5V: SupplyVoltage5V

(double) SupplyVoltage12V: SupplyVoltage12V

(double) SupplyVoltageM12V: SupplyVoltageM12V

(double) SupplyVoltageM5V: SupplyVoltageM5V

(double) SupplyVoltage5VSB: SupplyVoltage5VSB

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CPUCoreAndBoardSupplyVoltagesGet command which is used to . Refer to the XPS Programmer's manual to get the command description.

CPUTemperatureAndFanSpeedGet

Syntax

C# prototype

int CPUTemperatureAndFanSpeedGet(out double CPUTemperature, out double CPUFanSpeed, out string errstring)

Python prototype

[CPUTemperature, CPUFanSpeed, errstring] CPUTemperatureAndFanSpeedGet ()

Parameters

Input parameters

None

Output parameters

(double) CPUTemperature: CPUTemperature

(double) CPUFanSpeed: CPUFanSpeed

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CPUTemperatureAndFanSpeedGet command which is used to . Refer to the XPS Programmer's manual to get the command description.

CIEHeaderGet

Syntax

C# prototype

int CIEHeaderGet(Int32 CIEBoardIndex, out string HeaderString, out string errstring)

Python prototype

[HeaderString, errstring] CIEHeaderGet (CIEBoardIndex)

Parameters

Input parameters

(Int32) CIEBoardIndex: CIEBoardIndex

Output parameters

(string) HeaderString: HeaderString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CIEHeaderGet command which is used to Return the CIE board header string. Refer to the XPS Programmer's manual to get the command description.

CIEReset

Syntax

C# prototype

```
int CIEReset( out string errstring)
```

Python prototype

```
[errstring] CIEReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CIEReset command which is used to Stop and reset all CIE boards. Refer to the XPS Programmer's manual to get the command description.

ActionListGet

Syntax

C# prototype

```
int ActionListGet(out string ActionList, out string errstring)
```

Python prototype

```
[ActionList, errstring] ActionListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) ActionList: ActionList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ActionListGet command which is used to Action list. Refer to the XPS Programmer's manual to get the command description.

ActionExtendedListGet

Syntax

C# prototype

```
int ActionExtendedListGet(out string ActionList, out string errstring)
```

Python prototype

```
[ActionList, errstring] ActionExtendedListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) ActionList: ActionList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ActionExtendedListGet command which is used to Action extended list. Refer to the XPS Programmer's manual to get the command description.

APIExtendedListGet

Syntax

C# prototype

int APIExtendedListGet(out string Method, out string errstring)

Python prototype

[Method, errstring] APIExtendedListGet ()

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous APIExtendedListGet command which is used to API method list. Refer to the XPS Programmer's manual to get the command description.

APIListGet

Syntax

C# prototype

int APIListGet(out string Method, out string errstring)

Python prototype

[Method, errstring] APIListGet ()

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous APIListGet command which is used to User API method list without extended API. Refer to the XPS Programmer's manual to get the command description.

APIListStandardGet

Syntax

C# prototype

```
int APIListStandardGet(out string Method, out string errstring)
```

Python prototype

```
[Method, errstring] APIListStandardGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous APIListStandardGet command which is used to Standard API method list without extended API. Refer to the XPS Programmer's manual to get the command description.

APIListAMATGet

Syntax

C# prototype

int APIListAMATGet(out string Method, out string errstring)

Python prototype

[Method, errstring] APIListAMATGet ()

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous APIListAMATGet command which is used to AMAT API method list without extended API. Refer to the XPS Programmer's manual to get the command description.

TCLScriptsListGet

Syntax

C# prototype

int TCLScriptsListGet(out string Method, out string errstring)

Python prototype

[Method, errstring] TCLScriptsListGet ()

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TCLScriptsListGet command which is used to Returns the list of existing TCL scripts in /Admin/Public/Scripts folder. Refer to the XPS Programmer's manual to get the command description.

TrajectoryFilesListGet

Syntax

C# prototype

```
int TrajectoryFilesListGet(out string Method, out string errstring)
```

Python prototype

```
[Method, errstring] TrajectoryFilesListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) Method: Method

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TrajectoryFilesListGet command which is used to Returns the list of existing trajectory files in /Admin/Public/Trajectories folder. Refer to the XPS Programmer's manual to get the command description.

ControllerStatusListGet

Syntax

C# prototype

```
int ControllerStatusListGet(out string ControllerStatusList, out string errstring)
```

Python prototype

```
[ControllerStatusList, errstring] ControllerStatusListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) ControllerStatusList: ControllerStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerStatusListGet command which is used to Controller status list. Refer to the XPS Programmer's manual to get the command description.

ErrorListGet

Syntax

C# prototype

```
int ErrorListGet(out string ErrorsList, out string errstring)
```

Python prototype

```
[ErrorsList, errstring] ErrorListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) ErrorsList: ErrorsList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ErrorListGet command which is used to Error list. Refer to the XPS Programmer's manual to get the command description.

EventListGet

Syntax

C# prototype

```
int EventListGet(out string EventList, out string errstring)
```

Python prototype

```
[EventList, errstring] EventListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) EventList: EventList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventListGet command which is used to General event list. Refer to the XPS Programmer's manual to get the command description.

GatheringListGet

Syntax

C# prototype

```
int GatheringListGet(out string list, out string errstring)
```

Python prototype

```
[list, errstring] GatheringListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringListGet command which is used to Gathering type list. Refer to the XPS Programmer's manual to get the command description.

GatheringExtendedListGet

Syntax

C# prototype

```
int GatheringExtendedListGet(out string list, out string errstring)
```

Python prototype

```
[list, errstring] GatheringExtendedListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExtendedListGet command which is used to Gathering type extended list. Refer to the XPS Programmer's manual to get the command description.

GatheringExternalListGet

Syntax

C# prototype

```
int GatheringExternalListGet(out string list, out string errstring)
```

Python prototype

```
[list, errstring] GatheringExternalListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringExternalListGet command which is used to External Gathering type list. Refer to the XPS Programmer's manual to get the command description.

GroupStatusListGet

Syntax

C# prototype

```
int GroupStatusListGet(out string GroupStatusList, out string errstring)
```

Python prototype

```
[GroupStatusList, errstring] GroupStatusListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) GroupStatusList: GroupStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GroupStatusListGet command which is used to Group status list. Refer to the XPS Programmer's manual to get the command description.

HardwareInternalListGet

Syntax

C# prototype

```
int HardwareInternalListGet(out string InternalHardwareList, out string errstring)
```

Python prototype

```
[InternalHardwareList, errstring] HardwareInternalListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) InternalHardwareList: InternalHardwareList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous HardwareInternalListGet command which is used to Internal hardware list. Refer to the XPS Programmer's manual to get the command description.

ObjectsListGet

Syntax

C# prototype

```
int ObjectsListGet(out string ObjectsList, out string errstring)
```

Python prototype

```
[ObjectsList, errstring] ObjectsListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) ObjectsList: ObjectsList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ObjectsListGet command which is used to Group name and positioner name. Refer to the XPS Programmer's manual to get the command description.

PositionerErrorListGet

Syntax

C# prototype

```
int PositionerErrorListGet(out string PositionerErrorList, out string errstring)
```

Python prototype

```
[PositionerErrorList, errstring] PositionerErrorListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) PositionerErrorList: PositionerErrorList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerErrorListGet command which is used to Positioner error list. Refer to the XPS Programmer's manual to get the command description.

PositionerHardwareStatusListGet

Syntax

C# prototype

```
int PositionerHardwareStatusListGet(out string PositionerHardwareStatusList, out string errstring)
```

Python prototype

```
[PositionerHardwareStatusList, errstring] PositionerHardwareStatusListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) PositionerHardwareStatusList: PositionerHardwareStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerHardwareStatusListGet command which is used to Positioner hardware status list. Refer to the XPS Programmer's manual to get the command description.

PositionerDriverStatusListGet

Syntax

C# prototype

```
int PositionerDriverStatusListGet(out string PositionerDriverStatusList, out string errstring)
```

Python prototype

```
[PositionerDriverStatusList, errstring] PositionerDriverStatusListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) PositionerDriverStatusList: PositionerDriverStatusList

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous PositionerDriverStatusListGet command which is used to Positioner driver status list. Refer to the XPS Programmer's manual to get the command description.

ReferencingActionListGet

Syntax

C# prototype

```
int ReferencingActionListGet(out string list, out string errstring)
```

Python prototype

```
[list, errstring] ReferencingActionListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ReferencingActionListGet command which is used to Get referencing action list. Refer to the XPS Programmer's manual to get the command description.

ReferencingSensorListGet

Syntax

C# prototype

```
int ReferencingSensorListGet(out string list, out string errstring)
```

Python prototype

```
[list, errstring] ReferencingSensorListGet ()
```

Parameters

Input parameters

None

Output parameters

(string) list: list

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ReferencingSensorListGet command which is used to Get referencing sensor list. Refer to the XPS Programmer's manual to get the command description.

SystemIniParameterGet

Syntax

C# prototype

```
int SystemIniParameterGet(string SectionName, string ParameterName, out string ParameterValue, out string errstring)
```

Python prototype

```
[ParameterValue, errstring] SystemIniParameterGet (SectionName, ParameterName)
```

Parameters

Input parameters

(string) SectionName: SectionName

(string) ParameterName: ParameterName

Output parameters

(string) ParameterValue: ParameterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SystemIniParameterGet command which is used to Return the system.ini parameter. Refer to the XPS Programmer's manual to get the command description.

SystemIniParameterSet

Syntax

C# prototype

```
int SystemIniParameterSet(string SectionName, string ParameterName, string ParameterValue, out string errstring)
```

Python prototype

```
[errstring] SystemIniParameterSet (SectionName, ParameterName, ParameterValue)
```

Parameters

Input parameters

(string) SectionName: SectionName

(string) ParameterName: ParameterName

(string) ParameterValue: ParameterValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SystemIniParameterSet command which is used to Save the system.ini parameter. Refer to the XPS Programmer's manual to get the command description.

SystemRefParameterGet

Syntax

C# prototype

```
int SystemRefParameterGet(string SectionName, string ParameterName, out string ParameterValue, out string errstring)
```

Python prototype

```
[ParameterValue, errstring] SystemRefParameterGet (SectionName, ParameterName)
```

Parameters

Input parameters

(string) SectionName: SectionName

(string) ParameterName: ParameterName

Output parameters

(string) ParameterValue: ParameterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SystemRefParameterGet command which is used to Return the system.ref parameter. Refer to the XPS Programmer's manual to get the command description.

SystemRefParameterSet

Syntax

C# prototype

```
int SystemRefParameterSet(string SectionName, string ParameterName, string ParameterValue, out string  
errstring)
```

Python prototype

```
[errstring] SystemRefParameterSet (SectionName, ParameterName, ParameterValue)
```

Parameters

Input parameters

(string) SectionName: SectionName

(string) ParameterName: ParameterName

(string) ParameterValue: ParameterValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SystemRefParameterSet command which is used to Save the system.ref parameter. Refer to the XPS Programmer's manual to get the command description.

FirmwareRefParameterGet

Syntax

C# prototype

```
int FirmwareRefParameterGet(string SectionName, string ParameterName, out string ParameterValue, out string errstring)
```

Python prototype

```
[ParameterValue, errstring] FirmwareRefParameterGet (SectionName, ParameterName)
```

Parameters

Input parameters

(string) SectionName: SectionName

(string) ParameterName: ParameterName

Output parameters

(string) ParameterValue: ParameterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous FirmwareRefParameterGet command which is used to Return the system.ref parameter. Refer to the XPS Programmer's manual to get the command description.

GatheringUserDatasGet

Syntax

C# prototype

```
int GatheringUserDatasGet(out double UserData1, out double UserData2, out double UserData3, out double
UserData4, out double UserData5, out double UserData6, out double UserData7, out double UserData8, out double
UserData9, out double UserData10, out double UserData11, out double UserData12, out double UserData13, out
double UserData14, out double UserData15, out double UserData16, out string errstring)
```

Python prototype

```
[UserData1, UserData2, UserData3, UserData4, UserData5, UserData6, UserData7, UserData8, UserData9,
UserData10, UserData11, UserData12, UserData13, UserData14, UserData15, UserData16, errstring]
GatheringUserDatasGet ()
```

Parameters

Input parameters

None

Output parameters

(double) UserData1: UserData1

(double) UserData2: UserData2

(double) UserData3: UserData3

(double) UserData4: UserData4

(double) UserData5: UserData5

(double) UserData6: UserData6

(double) UserData7: UserData7

(double) UserData8: UserData8

(double) UserData9: UserData9

(double) UserData10: UserData10

(double) UserData11: UserData11

(double) UserData12: UserData12

(double) UserData13: UserData13

(double) UserData14: UserData14

(double) UserData15: UserData15

(double) UserData16: UserData16

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringUserDatasGet command which is used to Return UserData values. Refer to the XPS Programmer's manual to get the command description.

GatheringUserDatasReset

Syntax

C# prototype

```
int GatheringUserDatasReset( out string errstring)
```

Python prototype

```
[errstring] GatheringUserDatasReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous GatheringUserDatasReset command which is used to Reset UserData values. Refer to the XPS Programmer's manual to get the command description.

SocketsStatusGet

Syntax

C# prototype

```
int SocketsStatusGet(out string SocketsStatus, out string errstring)
```

Python prototype

```
[SocketsStatus, errstring] SocketsStatusGet ()
```

Parameters

Input parameters

None

Output parameters

(string) SocketsStatus: SocketsStatus

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous SocketsStatusGet command which is used to Get sockets current status. Refer to the XPS Programmer's manual to get the command description.

TestTCP

Syntax

C# prototype

```
int TestTCP(string InputString, out string ReturnString, out string errstring)
```

Python prototype

```
[ReturnString, errstring] TestTCP (InputString)
```

Parameters

Input parameters

(string) InputString: InputString

Output parameters

(string) ReturnString: ReturnString

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous TestTCP command which is used to Test TCP/IP transfert. Refer to the XPS Programmer's manual to get the command description.

OptionalModuleExecute

Syntax

C# prototype

```
int OptionalModuleExecute(string ModuleFileName, out string errstring)
```

Python prototype

```
[errstring] OptionalModuleExecute (ModuleFileName)
```

Parameters

Input parameters

(string) ModuleFileName: ModuleFileName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous OptionalModuleExecute command which is used to Execute an optional module. Refer to the XPS Programmer's manual to get the command description.

OptionalModuleKill

Syntax

C# prototype

```
int OptionalModuleKill(string TaskName, out string errstring)
```

Python prototype

```
[errstring] OptionalModuleKill (TaskName)
```

Parameters

Input parameters

(string) TaskName: TaskName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous OptionalModuleKill command which is used to Kill an optional module. Refer to the XPS Programmer's manual to get the command description.

CIERegister32ValueGet

Syntax

C# prototype

```
int CIERegister32ValueGet(string PositionerName, Int16 BoardOffset, out Int32 RegisterValue, out string errstring)
```

Python prototype

```
[RegisterValue, errstring] CIERegister32ValueGet (PositionerName, BoardOffset)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int16) BoardOffset: BoardOffset

Output parameters

(Int32_i) RegisterValue: RegisterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CIERegister32ValueGet command which is used to Get register value of the CIE board. Refer to the XPS Programmer's manual to get the command description.

CIERegister64ValueGet

Syntax

C# prototype

```
int CIERegister64ValueGet(string PositionerName, Int16 BoardOffset, out Int32 RegisterValue, out string errstring)
```

Python prototype

```
[RegisterValue, errstring] CIERegister64ValueGet (PositionerName, BoardOffset)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int16) BoardOffset: BoardOffset

Output parameters

(Int32_i) RegisterValue: RegisterValue

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CIERegister64ValueGet command which is used to Get register value of the CIE board. Refer to the XPS Programmer's manual to get the command description.

CIERegisterValueSet

Syntax

C# prototype

```
int CIERegisterValueSet(string PositionerName, Int16 BoardOffset, Int32 RegisterValue, out string errstring)
```

Python prototype

```
[errstring] CIERegisterValueSet (PositionerName, BoardOffset, RegisterValue)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

(Int16) BoardOffset: BoardOffset

(Int32) RegisterValue: RegisterValue

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CIERegisterValueSet command which is used to Set register value of the CIE board. Refer to the XPS Programmer's manual to get the command description.

DRV11StatusGet

Syntax

C# prototype

```
int DRV11StatusGet(string PositionerName, out UInt16 DriverStatusByte1, out UInt16 DriverStatusByte2, out string errstring)
```

Python prototype

```
[DriverStatusByte1, DriverStatusByte2, errstring] DRV11StatusGet (PositionerName)
```

Parameters

Input parameters

(string) PositionerName: PositionerName

Output parameters

(UInt16) DriverStatusByte1: DriverStatusByte1

(UInt16) DriverStatusByte2: DriverStatusByte2

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DRV11StatusGet command which is used to Get driver status bytes. Refer to the XPS Programmer's manual to get the command description.

DebugTraceCommunicationReset

Syntax

C# prototype

```
int DebugTraceCommunicationReset( out string errstring)
```

Python prototype

```
[errstring] DebugTraceCommunicationReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugTraceCommunicationReset command which is used to Reset rolling buffer. Refer to the XPS Programmer's manual to get the command description.

DebugTraceCommunicationSave

Syntax

C# prototype

```
int DebugTraceCommunicationSave( out string errstring)
```

Python prototype

```
[errstring] DebugTraceCommunicationSave ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugTraceCommunicationSave command which is used to Save rolling buffer. Refer to the XPS Programmer's manual to get the command description.

RunTraceloggerProcessWithTimeSetting

Syntax

C# prototype

```
int RunTraceloggerProcessWithTimeSetting(Int32 TraceloggerTime, out string errstring)
```

Python prototype

```
[errstring] RunTraceloggerProcessWithTimeSetting (TraceloggerTime)
```

Parameters

Input parameters

(Int32) TraceloggerTime: TraceloggerTime

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous RunTraceloggerProcessWithTimeSetting command which is used to Run Tracelogger Process With Time Setting. Refer to the XPS Programmer's manual to get the command description.

RunTraceloggerProcessWithRollingBuffer

Syntax

C# prototype

```
int RunTraceloggerProcessWithRollingBuffer( out string errstring)
```

Python prototype

```
[errstring] RunTraceloggerProcessWithRollingBuffer ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous RunTraceloggerProcessWithRollingBuffer command which is used to Run Tracelogger Process With Rolling Buffer. Refer to the XPS Programmer's manual to get the command description.

CreateQNXEvent

Syntax

C# prototype

```
int CreateQNXEvent(Int32 EventNumber, string EventName, out string errstring)
```

Python prototype

```
[errstring] CreateQNXEvent (EventNumber, EventName)
```

Parameters

Input parameters

(Int32) EventNumber: EventNumber

(string) EventName: EventName

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous CreateQNXEvent command which is used to Run Create Event. Refer to the XPS Programmer's manual to get the command description.

StartEventsAcquisition

Syntax

C# prototype

```
int StartEventsAcquisition( out string errstring)
```

Python prototype

```
[errstring] StartEventsAcquisition ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous StartEventsAcquisition command which is used to Start Events Acquisition. Refer to the XPS Programmer's manual to get the command description.

StopEventsAcquisition

Syntax

C# prototype

```
int StopEventsAcquisition( out string errstring)
```

Python prototype

```
[errstring] StopEventsAcquisition ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous StopEventsAcquisition command which is used to Stop Events Acquisition. Refer to the XPS Programmer's manual to get the command description.

EventTriggerSet

Syntax

C# prototype

```
int EventTriggerSet( out string errstring)
```

Python prototype

```
[errstring] EventTriggerSet ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous EventTriggerSet command which is used to Generate internal event. Refer to the XPS Programmer's manual to get the command description.

DebugISRReset

Syntax

C# prototype

```
int DebugISRReset( out string errstring)
```

Python prototype

```
[errstring] DebugISRReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRReset command which is used to Reset ISR load ratios and usage times. Refer to the XPS Programmer's manual to get the command description.

DebugISRCorrectorLoadRatioGet

Syntax

C# prototype

```
int DebugISRCorrectorLoadRatioGet(out double ISRCorrectorLoadRatio, out double ISRCorrectorLoadRatioMin, out double ISRCorrectorLoadRatioMax, out string errstring)
```

Python prototype

```
[ISRCorrectorLoadRatio, ISRCorrectorLoadRatioMin, ISRCorrectorLoadRatioMax, errstring]  
DebugISRCorrectorLoadRatioGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRCorrectorLoadRatio: ISRCorrectorLoadRatio

(double) ISRCorrectorLoadRatioMin: ISRCorrectorLoadRatioMin

(double) ISRCorrectorLoadRatioMax: ISRCorrectorLoadRatioMax

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRCorrectorLoadRatioGet command which is used to Get Corrector load ratio. Refer to the XPS Programmer's manual to get the command description.

DebugISRProfilerLoadRatioGet

Syntax

C# prototype

```
int DebugISRProfilerLoadRatioGet(out double ISRProfilerLoadRatio, out double ISRProfilerLoadRatioMin, out double ISRProfilerLoadRatioMax, out string errstring)
```

Python prototype

```
[ISRProfilerLoadRatio, ISRProfilerLoadRatioMin, ISRProfilerLoadRatioMax, errstring] DebugISRProfilerLoadRatioGet()  
( )
```

Parameters

Input parameters

None

Output parameters

(double) ISRProfilerLoadRatio: ISRProfilerLoadRatio

(double) ISRProfilerLoadRatioMin: ISRProfilerLoadRatioMin

(double) ISRProfilerLoadRatioMax: ISRProfilerLoadRatioMax

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRProfilerLoadRatioGet command which is used to Get Profiler load ratio. Refer to the XPS Programmer's manual to get the command description.

DebugISRServitudesLoadRatioGet

Syntax

C# prototype

```
int DebugISRServitudesLoadRatioGet(out double ISRServitudesLoadRatio, out double ISRServitudesLoadRatioMin, out double ISRServitudesLoadRatioMax, out string errstring)
```

Python prototype

```
[ISRServitudesLoadRatio, ISRServitudesLoadRatioMin, ISRServitudesLoadRatioMax, errstring]  
DebugISRServitudesLoadRatioGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRServitudesLoadRatio: ISRServitudesLoadRatio

(double) ISRServitudesLoadRatioMin: ISRServitudesLoadRatioMin

(double) ISRServitudesLoadRatioMax: ISRServitudesLoadRatioMax

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRServitudesLoadRatioGet command which is used to Get Servitudes load ratio. Refer to the XPS Programmer's manual to get the command description.

DebugISRCorrectorUsageGet

Syntax

C# prototype

```
int DebugISRCorrectorUsageGet(out double ISRCorrectorUsageCurrent, out double
ISRCorrectorPeriodUsageCurrent, out double ISRCorrectorUsageMin, out double ISRCorrectorPeriodUsageMin, out
double ISRCorrectorUsageMax, out double ISRCorrectorPeriodUsageMax, out string errstring)
```

Python prototype

```
[ISRCorrectorUsageCurrent, ISRCorrectorPeriodUsageCurrent, ISRCorrectorUsageMin, ISRCorrectorPeriodUsageMin,
ISRCorrectorUsageMax, ISRCorrectorPeriodUsageMax, errstring] DebugISRCorrectorUsageGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRCorrectorUsageCurrent: ISRCorrectorUsageCurrent

(double) ISRCorrectorPeriodUsageCurrent: ISRCorrectorPeriodUsageCurrent

(double) ISRCorrectorUsageMin: ISRCorrectorUsageMin

(double) ISRCorrectorPeriodUsageMin: ISRCorrectorPeriodUsageMin

(double) ISRCorrectorUsageMax: ISRCorrectorUsageMax

(double) ISRCorrectorPeriodUsageMax: ISRCorrectorPeriodUsageMax

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRCorrectorUsageGet command which is used to Get ISR Corrector time usage and period. Refer to the XPS Programmer's manual to get the command description.

DebugISRProfilerUsageGet

Syntax

C# prototype

```
int DebugISRProfilerUsageGet(out double ISRProfilerUsageCurrent, out double ISRProfilerPeriodUsageCurrent, out double ISRProfilerUsageMin, out double ISRProfilerPeriodUsageMin, out double ISRProfilerUsageMax, out double ISRProfilerPeriodUsageMax, out Int64 ISRProfilerLostCounter, out string errstring)
```

Python prototype

```
[ISRProfilerUsageCurrent, ISRProfilerPeriodUsageCurrent, ISRProfilerUsageMin, ISRProfilerPeriodUsageMin, ISRProfilerUsageMax, ISRProfilerPeriodUsageMax, ISRProfilerLostCounter, errstring] DebugISRProfilerUsageGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRProfilerUsageCurrent: ISRProfilerUsageCurrent

(double) ISRProfilerPeriodUsageCurrent: ISRProfilerPeriodUsageCurrent

(double) ISRProfilerUsageMin: ISRProfilerUsageMin

(double) ISRProfilerPeriodUsageMin: ISRProfilerPeriodUsageMin

(double) ISRProfilerUsageMax: ISRProfilerUsageMax

(double) ISRProfilerPeriodUsageMax: ISRProfilerPeriodUsageMax

(Int64) ISRProfilerLostCounter: ISRProfilerLostCounter

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRProfilerUsageGet command which is used to Get ISR Profiler time usage and period. Refer to the XPS Programmer's manual to get the command description.

DebugISRServitudesUsageGet

Syntax

C# prototype

```
int DebugISRServitudesUsageGet(out double ISRServitudesUsageCurrent, out double
ISRServitudesPeriodUsageCurrent, out double ISRServitudesUsageMin, out double ISRServitudesPeriodUsageMin,
out double ISRServitudesUsageMax, out double ISRServitudesPeriodUsageMax, out string errstring)
```

Python prototype

```
[ISRServitudesUsageCurrent, ISRServitudesPeriodUsageCurrent, ISRServitudesUsageMin,
ISRServitudesPeriodUsageMin, ISRServitudesUsageMax, ISRServitudesPeriodUsageMax, errstring]
DebugISRServitudesUsageGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRServitudesUsageCurrent: ISRServitudesUsageCurrent

(double) ISRServitudesPeriodUsageCurrent: ISRServitudesPeriodUsageCurrent

(double) ISRServitudesUsageMin: ISRServitudesUsageMin

(double) ISRServitudesPeriodUsageMin: ISRServitudesPeriodUsageMin

(double) ISRServitudesUsageMax: ISRServitudesUsageMax

(double) ISRServitudesPeriodUsageMax: ISRServitudesPeriodUsageMax

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugISRServitudesUsageGet command which is used to Get ISR Servitudes time usage and period. Refer to the XPS Programmer's manual to get the command description.

DebugCorrectorTimeUsageGet

Syntax

C# prototype

```
int DebugCorrectorTimeUsageGet(out double ISRCorrectorStartTime, out double PreviousISRCorrectorPeriod, out double ISRInitParam0Duration, out double ISRInitParam1Duration, out double ISRTimerDuration, out double ISRGPIODuration, out double ISRGroupsDuration, out double ISREventDuration, out double ISRGeneralInhibDuration, out double ISRGatheringDuration, out double ISRInitParam2Duration, out string errstring)
```

Python prototype

```
[ISRCorrectorStartTime, PreviousISRCorrectorPeriod, ISRInitParam0Duration, ISRInitParam1Duration, ISRTimerDuration, ISRGPIODuration, ISRGroupsDuration, ISREventDuration, ISRGeneralInhibDuration, ISRGatheringDuration, ISRInitParam2Duration, errstring] DebugCorrectorTimeUsageGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRCorrectorStartTime: ISRCorrectorStartTime

(double) PreviousISRCorrectorPeriod: PreviousISRCorrectorPeriod

(double) ISRInitParam0Duration: ISRInitParam0Duration

(double) ISRInitParam1Duration: ISRInitParam1Duration

(double) ISRTimerDuration: ISRTimerDuration

(double) ISRGPIODuration: ISRGPIODuration

(double) ISRGroupsDuration: ISRGroupsDuration

(double) ISREventDuration: ISREventDuration

(double) ISRGeneralInhibDuration: ISRGeneralInhibDuration

(double) ISRGatheringDuration: ISRGatheringDuration

(double) ISRInitParam2Duration: ISRInitParam2Duration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugCorrectorTimeUsageGet command which is used to Get Corrector segmentation durations. Refer to the XPS Programmer's manual to get the command description.

DebugProfilerTimeUsageGet

Syntax

C# prototype

```
int DebugProfilerTimeUsageGet(out double ISRCorrectProfilDelta, out double ISRGroupsDuration, out double  
ISRInitParamDuration, out string errstring)
```

Python prototype

```
[ISRCorrectProfilDelta, ISRGroupsDuration, ISRInitParamDuration, errstring] DebugProfilerTimeUsageGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRCorrectProfilDelta: ISRCorrectProfilDelta

(double) ISRGroupsDuration: ISRGroupsDuration

(double) ISRInitParamDuration: ISRInitParamDuration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugProfilerTimeUsageGet command which is used to Get profiler segmentation durations. Refer to the XPS Programmer's manual to get the command description.

DebugServitudesTimeUsageGet

Syntax

C# prototype

```
int DebugServitudesTimeUsageGet(out double ISRProfilServDelta, out double ISRErrorCheckDuration, out double
ISRGroupsDuration, out double ISRInitParamDuration, out string errstring)
```

Python prototype

```
[ISRProfilServDelta, ISRErrorCheckDuration, ISRGroupsDuration, ISRInitParamDuration, errstring]
DebugServitudesTimeUsageGet ()
```

Parameters

Input parameters

None

Output parameters

(double) ISRProfilServDelta: ISRProfilServDelta

(double) ISRErrorCheckDuration: ISRErrorCheckDuration

(double) ISRGroupsDuration: ISRGroupsDuration

(double) ISRInitParamDuration: ISRInitParamDuration

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous DebugServitudesTimeUsageGet command which is used to Get Servitudes segmentation durations. Refer to the XPS Programmer's manual to get the command description.

ControllerMotionKernelMinMaxTimeLoadGet

Syntax

C# prototype

```
int ControllerMotionKernelMinMaxTimeLoadGet(out double MinimumCPUTotalLoadRatio, out double
MaximumCPUTotalLoadRatio, out double MinimumCPUCorrectorLoadRatio, out double
MaximumCPUCorrectorLoadRatio, out double MinimumCPUProfilerLoadRatio, out double
MaximumCPUProfilerLoadRatio, out double MinimumCPUServitudesLoadRatio, out double
MaximumCPUServitudesLoadRatio, out string errstring)
```

Python prototype

```
[MinimumCPUTotalLoadRatio, MaximumCPUTotalLoadRatio, MinimumCPUCorrectorLoadRatio,
MaximumCPUCorrectorLoadRatio, MinimumCPUProfilerLoadRatio, MaximumCPUProfilerLoadRatio,
MinimumCPUServitudesLoadRatio, MaximumCPUServitudesLoadRatio, errstring]
ControllerMotionKernelMinMaxTimeLoadGet ()
```

Parameters

Input parameters

None

Output parameters

(double) MinimumCPUTotalLoadRatio: MinimumCPUTotalLoadRatio

(double) MaximumCPUTotalLoadRatio: MaximumCPUTotalLoadRatio

(double) MinimumCPUCorrectorLoadRatio: MinimumCPUCorrectorLoadRatio

(double) MaximumCPUCorrectorLoadRatio: MaximumCPUCorrectorLoadRatio

(double) MinimumCPUProfilerLoadRatio: MinimumCPUProfilerLoadRatio

(double) MaximumCPUProfilerLoadRatio: MaximumCPUProfilerLoadRatio

(double) MinimumCPUServitudesLoadRatio: MinimumCPUServitudesLoadRatio

(double) MaximumCPUServitudesLoadRatio: MaximumCPUServitudesLoadRatio

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerMotionKernelMinMaxTimeLoadGet command which is used to Get controller motion kernel minimum and maximum time load. Refer to the XPS Programmer's manual to get the command description.

ControllerMotionKernelMinMaxTimeLoadReset

Syntax

C# prototype

```
int ControllerMotionKernelMinMaxTimeLoadReset( out string errstring)
```

Python prototype

```
[errstring] ControllerMotionKernelMinMaxTimeLoadReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerMotionKernelMinMaxTimeLoadReset command which is used to Reset controller motion kernel min/max time load. Refer to the XPS Programmer's manual to get the command description.

ControllerMotionKernelPeriodMinMaxGet

Syntax

C# prototype

```
int ControllerMotionKernelPeriodMinMaxGet(out double MinimumCorrectorPeriod, out double
MaximumCorrectorPeriod, out double MinimumProfilerPeriod, out double MaximumProfilerPeriod, out double
MinimumServitudesPeriod, out double MaximumServitudesPeriod, out string errstring)
```

Python prototype

```
[MinimumCorrectorPeriod, MaximumCorrectorPeriod, MinimumProfilerPeriod, MaximumProfilerPeriod,
MinimumServitudesPeriod, MaximumServitudesPeriod, errstring] ControllerMotionKernelPeriodMinMaxGet ()
```

Parameters

Input parameters

None

Output parameters

(double) MinimumCorrectorPeriod: MinimumCorrectorPeriod

(double) MaximumCorrectorPeriod: MaximumCorrectorPeriod

(double) MinimumProfilerPeriod: MinimumProfilerPeriod

(double) MaximumProfilerPeriod: MaximumProfilerPeriod

(double) MinimumServitudesPeriod: MinimumServitudesPeriod

(double) MaximumServitudesPeriod: MaximumServitudesPeriod

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerMotionKernelPeriodMinMaxGet command which is used to Get controller motion kernel min/max periods. Refer to the XPS Programmer's manual to get the command description.

ControllerMotionKernelPeriodMinMaxReset

Syntax

C# prototype

```
int ControllerMotionKernelPeriodMinMaxReset( out string errstring)
```

Python prototype

```
[errstring] ControllerMotionKernelPeriodMinMaxReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ControllerMotionKernelPeriodMinMaxReset command which is used to Reset controller motion kernel min/max periods. Refer to the XPS Programmer's manual to get the command description.

ISRCorrectorCompensateOverrunNumberGet

Syntax

C# prototype

```
int ISRCorrectorCompensateOverrunNumberGet(out Int32 CorrectorOverrunCompensationNumber, out Int32  
CorrectorOverrunCompensationMissNumber, out string errstring)
```

Python prototype

```
[CorrectorOverrunCompensationNumber, CorrectorOverrunCompensationMissNumber, errstring]  
ISRCorrectorCompensateOverrunNumberGet ()
```

Parameters

Input parameters

None

Output parameters

(Int32_i) CorrectorOverrunCompensationNumber: CorrectorOverrunCompensationNumber

(Int32_i) CorrectorOverrunCompensationMissNumber: CorrectorOverrunCompensationMissNumber

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ISRCorrectorCompensateOverrunNumberGet command which is used to Get ISR Corrector Compensate Overrun Number. Refer to the XPS Programmer's manual to get the command description.

ISRCorrectorCompensateOverrunNumberReset

Syntax

C# prototype

```
int ISRCorrectorCompensateOverrunNumberReset( out string errstring)
```

Python prototype

```
[errstring] ISRCorrectorCompensateOverrunNumberReset ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous ISRCorrectorCompensateOverrunNumberReset command which is used to Reset ISR Corrector Compensate Overrun Number. Refer to the XPS Programmer's manual to get the command description.

RunQconn

Syntax

C# prototype

```
int RunQconn( out string errstring)
```

Python prototype

```
[errstring] RunQconn ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous RunQconn command which is used to Run qconn service. Refer to the XPS Programmer's manual to get the command description.

RelockFiles

Syntax

C# prototype

int RelockFiles(out string errstring)

Python prototype

[errstring] RelockFiles ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous RelockFiles command which is used to Relock files service. Refer to the XPS Programmer's manual to get the command description.

UnlockFiles

Syntax

C# prototype

```
int UnlockFiles( out string errstring)
```

Python prototype

```
[errstring] UnlockFiles ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous UnlockFiles command which is used to Unlock files service. Refer to the XPS Programmer's manual to get the command description.

Crash

Syntax

C# prototype

int Crash(out string errstring)

Python prototype

[errstring] Crash ()

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous Crash command which is used to Crash controller. Refer to the XPS Programmer's manual to get the command description.

RunPidin

Syntax

C# prototype

```
int RunPidin( out string errstring)
```

Python prototype

```
[errstring] RunPidin ()
```

Parameters

Input parameters

None

Output parameters

(string) errString: The failure reason

Return

(int) error code: 0 in success and -1 on failure

Description

This function is used to process synchronous RunPidin command which is used to Save pidin result in a Pidin.log file. Refer to the XPS Programmer's manual to get the command description.