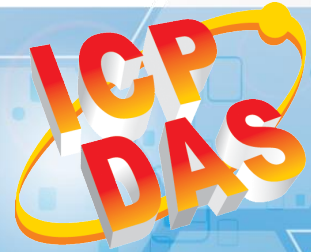


# EtherCAT MainDevice 軟體使用手冊

繁體中文 1.0.30 版本, 2024 年 3 月



## **承諾**

鄭重承諾: 凡泓格科技股份有限公司產品從購買後, 開始享有一年保固, 除人為使用不當的因素除外。

## **責任聲明**

凡使用本系列產品除產品品質所造成的損害, 泓格科技股份有限公司不承擔任何的法律責任。泓格科技股份有限公司有義務提供本系列產品詳細使用資料, 本使用手冊所提及的產品規格或相關資訊, 泓格科技保留所有修訂之權利, 本使用手冊所提及之產品規格或相關資訊有任何修改或變更時, 恕不另行通知, 本產品不承擔使用者非法利用資料對第三方所造成侵害構成的法律責任, 未事先經由泓格科技書面允許, 不得以任何形式複製、修改、轉載、傳送或出版使用手冊內容。

## **版權**

版權所有 © 2024 泓格科技股份有限公司, 保留所有權利。

## **商標**

文件中所涉及所有公司的商標, 商標名稱及產品名稱分別屬於該商標或名稱的擁有者所持有。

## **聯繫我們**

如有任何問題歡迎聯繫我們, 我們將會為您提供完善的諮詢服務。

[service@icpdas.com](mailto:service@icpdas.com); [service.icpdas@gmail.com](mailto:service.icpdas@gmail.com)

## **支援**

ECAT-M801 系列

EMP-9000 系列

# 目錄

1. 簡介.....	16
1.1. 版本更新資訊.....	16
2. 軟體安裝.....	18
2.1. 取得軟體安裝程式.....	18
2.2. 安裝 Windows 驅動程式.....	19
2.3. 移除 Windows 驅動程式.....	27
2.4. 安裝 Linux 驅動程式.....	29
2.4.1. 安裝 Linux 驅動程式.....	29
2.4.2. 移除 Linux 驅動程式.....	32
2.5. 更新韌體.....	33
2.6. 自動更新軟體.....	35
2.6.1. Windows.....	35
2.6.2. Linux.....	37
3. EcatUtility 操作說明.....	38
3.0. 啟動工具軟體.....	38
3.0.0. 連線.....	38
3.0.1. 選擇模組編號定義.....	39
3.0.2. 裝置初始化.....	41
3.0.3. ALIAS 編輯頁面.....	42
3.0.4. Utility 操作頁面.....	44
3.1. 裝置操作工具列說明.....	46
3.1.1. 網絡架構編輯步驟.....	47
3.1.2. 網絡架構編輯(進階配置 PDO).....	50
3.1.3. 網絡架構編輯(進階配置 PDO for CiA402 驅動器).....	52
3.1.4. 開始及停止 EtherCAT 操作任務步驟.....	56
3.2. 錯誤訊息面板.....	56
3.3. 裝置狀態列.....	56
3.4. 從站模組操作界面說明.....	58
3.4.1. 從站模組基本操作步驟.....	59
3.4.2. 從站模組 SDO 操作步驟.....	59
3.4.3. 從站模組 PDO 及 DI/DO LED 操作步驟.....	61
3.4.4. 從站模組 PDO 監看與分析.....	63
3.4.5. 從站模組韌體更新(FoE).....	65
3.5. 運動控制初始化操作工具列說明.....	66
3.5.1. 運動控制初始化參數檔編輯步驟.....	67

3.5.2.	運動控制初始化步驟 .....	71
3.6.	運動控制操作介面說明 .....	72
3.6.1.	單軸運動控制操作頁面說明 .....	72
3.6.2.	群組運動控制操作頁面說明 .....	77
3.6.3.	運動位置顯示頁面說明 .....	80
3.7.	裝置 I/O 操作頁面 .....	82
3.7.1.	裝置 DO 控制操作步驟 .....	82
3.7.2.	裝置 DI 控制操作步驟 .....	82
3.8.	PID 控制器操作介面說明 .....	84
3.8.1.	PID 控制器控制操作頁面說明 .....	84
3.9.	EtherCAT 診斷 .....	87
3.9.1.	EtherCAT 診斷操作頁面說明 .....	87
3.9.2.	EtherCAT 診斷說明 .....	88
4.	函式庫使用流程說明 .....	91
4.1.	裝置操作流程 .....	91
4.2.	從站模組操作流程 .....	93
4.3.	運動控制操作流程 .....	94
4.3.1.	運動控制初始化流程 .....	94
4.3.2.	單軸運動控制操作流程 .....	96
4.3.3.	單軸自動原點復歸流程 .....	98
4.3.4.	單軸錯誤狀態處理流程 .....	100
4.3.5.	群組運動控制操作流程 .....	102
4.4.	通信錯誤處理流程 .....	104
4.5.	函式庫在 Windows 程式編程說明 .....	106
4.5.1.	Visual Studio 工具使用說明 .....	106
5.	裝置操作函式集說明 .....	108
5.1.	ECAT_GetDeviceCnt .....	108
5.2.	ECAT_OpenDevice .....	110
5.3.	ECAT_CloseDevice .....	112
5.4.	ECAT_GetDeviceSerialNo .....	114
5.5.	ECAT_GetDllVersion .....	116
5.6.	ECAT_GetFirmwareVersion .....	118
5.7.	ECAT_GetDeviceDI .....	120
5.8.	ECAT_GetDeviceDIBit .....	122
5.9.	ECAT_GetDeviceDO .....	124
5.10.	ECAT_GetDeviceDOBit .....	126
5.11.	ECAT_SetDeviceDO .....	128
5.12.	ECAT_SetDeviceDOBit .....	130

5.13.	ECAT_SetDeviceEncProperty.....	132
5.14.	ECAT_GetDeviceEncProperty .....	135
5.15.	ECAT_GetDeviceEncCount .....	137
5.16.	ECAT_ResetDeviceEncCount.....	139
5.17.	ECAT_SetDeviceCmpTrigProperty .....	141
5.18.	ECAT_GetDeviceCmpTrigProperty.....	143
5.19.	ECAT_SetDeviceCmpTrigData .....	145
5.20.	ECAT_SetDeviceContCmpTrigData .....	147
5.21.	ECAT_SetDeviceCmpDisable .....	149
5.22.	ECAT_SetDeviceEmg .....	151
5.23.	ECAT_GetDeviceEmg.....	153
5.24.	ECAT_GetDeviceEmgStatus .....	155
5.25.	ECAT_SetDeviceEmgSoftSig .....	157
5.26.	ECAT_SetDeviceMPG .....	159
5.27.	ECAT_GetDeviceMPG .....	162
5.28.	ECAT_GetDeviceState.....	164
5.29.	ECAT_GetDeviceStateEx .....	167
5.30.	ECAT_StartDeviceOpTask.....	170
5.31.	ECAT_StopDeviceOpTask.....	173
5.32.	ECAT_SetTimer .....	175
5.33.	ECAT_SetTimerStop .....	177
5.34.	ECAT_WaitforTimer .....	179
5.35.	ECAT_GetProcessTime .....	181
5.36.	ECAT_SetHeartBeat .....	183
5.37.	ECAT_SetHeartBeatStatus .....	185
5.38.	ECAT_SetDeviceIgnoreWC .....	187
5.39.	ECAT_GetDeviceIgnoreWC.....	189
5.40.	ECAT_SetCheckSlaveCnt.....	191
6.	從站操作函式集說明.....	193
6.1.	ECAT_SetSlaveNoType .....	193
6.2.	ECAT_GetSlaveNoType .....	196
6.3.	ECAT_GetSlaveInfo .....	199
6.4.	ECAT_GetSlaveSdoObject .....	202
6.5.	ECAT_SetSlaveSdoObject.....	204
6.6.	ECAT_SetSlaveRxPdoData_Ex .....	206
6.7.	ECAT_GetSlaveRxPdoData_Ex.....	209
6.8.	ECAT_GetSlaveTxPdoData_Ex .....	212
6.9.	ECAT_SetSlaveDIMap .....	215

6.10.	ECAT_SetSlaveDIMap_16bit .....	218
6.11.	ECAT_GetSlaveDI.....	221
6.12.	ECAT_GetSlaveDI_Directly.....	223
6.13.	ECAT_GetSlaveDI_Directly_16bit .....	224
6.14.	ECAT_GetSlaveDIBit .....	228
6.15.	ECAT_GetSlaveDIBit_Directly .....	230
6.16.	ECAT_GetSlaveDO.....	231
6.17.	ECAT_GetSlaveDO_Directly.....	233
6.18.	ECAT_GetMultiSlaveDO_Ex .....	234
6.19.	ECAT_GetSlaveDOBit .....	236
6.20.	ECAT_GetSlaveDOBit_Directly.....	238
6.21.	ECAT_SetSlaveDO .....	239
6.22.	ECAT_SetMultiSlaveDO_Ex .....	241
6.23.	ECAT_SetMultiSlaveDO_AutoOff_Ex .....	243
6.24.	ECAT_SetSlaveDOBit.....	246
6.25.	ECAT_do_cfg_save.....	248
6.26.	ECAT_do_cfg_load.....	250
6.27.	ECAT_ClearDoQueue.....	252
6.28.	ECAT_PreSetSlaveDO .....	254
6.29.	ECAT_SetSlaveAoProperty .....	256
6.30.	ECAT_GetSlaveAoProperty .....	258
6.31.	ECAT_SetSlaveAoRawData .....	260
6.32.	ECAT_GetSlaveAoRawData.....	262
6.33.	ECAT_SetSlaveAoVoltData .....	264
6.34.	ECAT_GetSlaveAoVoltData.....	266
6.35.	ECAT_SetSlaveAiProperty.....	268
6.36.	ECAT_GetSlaveAiProperty .....	271
6.37.	ECAT_GetSlaveAiRawData.....	273
6.38.	ECAT_GetSlaveAiVoltData .....	275
6.39.	ECAT_GetSlaveAiMADData.....	277
6.40.	ECAT_Set_ECAT2016_AiProperty .....	279
6.41.	ECAT_Get_ECAT2016_AiProperty.....	281
6.42.	ECAT_Get_ECAT2016_AiRawData.....	283
6.43.	ECAT_Get_ECAT2016_AiVoltData.....	285
6.44.	ECAT_SetSlaveEncProperty.....	287
6.45.	ECAT_GetSlaveEncProperty .....	289
6.46.	ECAT_GetSlaveEncCount .....	291
6.47.	ECAT_ResetSlaveEncCount.....	293

6.48.	ECAT_SetSlaveEncCount.....	295
6.49.	ECAT_SetSlaveEnclIdxLatchProperty .....	297
6.50.	ECAT_GetSlaveEnclIdxLatchProperty.....	299
6.51.	ECAT_GetSlaveEnclIdxLatchCnt.....	301
6.52.	ECAT_ResetSlaveEnclIdxLatchCnt .....	303
6.53.	ECAT_SetSlaveEnclIdxLatchBufferEnable.....	305
6.54.	ECAT_GetSlaveEnclIdxLatchBuffer .....	307
6.55.	ECAT_GetSlaveEnclIdxLatchBuffer.....	309
6.56.	ECAT_ResetSlaveEnclIdxLatchBuffer .....	311
6.57.	ECAT_SetSlaveEncExtLatchProperty.....	313
6.58.	ECAT_GetSlaveEncExtLatchProperty .....	315
6.59.	ECAT_GetSlaveEncExtLatchCnt .....	317
6.60.	ECAT_ResetSlaveEncExtLatchCnt.....	319
6.61.	ECAT_SetSlaveCmpTrigProperty .....	321
6.62.	ECAT_GetSlaveCmpTrigProperty.....	323
6.63.	ECAT_SetSlaveCmpTrigData .....	325
6.64.	ECAT_SetSlaveContCmpTrigData .....	327
6.65.	ECAT_SetSlaveArrCmpPos.....	329
6.66.	ECAT_GetSlaveArrCmpPos .....	332
6.67.	ECAT_SetSlaveArrCmpEnable .....	334
6.68.	ECAT_GetSlaveArrCmpEnable .....	338
6.69.	ECAT_SetSlaveArrCmpEndIdx .....	340
6.70.	ECAT_GetSlaveArrCmpEndIdx .....	344
6.71.	ECAT_SetSlaveArrCmpTrig.....	346
6.72.	ECAT_SetSlaveSaveArrCmpData .....	349
6.73.	ECAT_SetTxPdoBufParam .....	351
6.74.	ECAT_GetTxPdoBufParam.....	353
6.75.	ECAT_SetTxPdoBufEnable .....	355
6.76.	ECAT_GetTxPdoBufEnable .....	357
6.77.	ECAT_GetTxPdoBufValue .....	359
6.78.	ECAT_SetAiFilterParam.....	361
6.79.	ECAT_GetAiFilterParam .....	363
6.80.	ECAT_SetAiFilterEnable .....	365
6.81.	ECAT_GetAiFilterEnable.....	367
6.82.	ECAT_SetAiFilterFreq.....	369
6.83.	ECAT_GetAiFilterFreq .....	371
6.84.	ECAT_GetAiFilterOutput .....	373
6.85.	ECAT_SetPdoInToOutParam .....	376

6.86.	ECAT_GetPdInToOutParam .....	378
6.87.	ECAT_SetPdInToOutCoeff .....	380
6.88.	ECAT_GetPdInToOutCoeff .....	382
6.89.	ECAT_SetPdInToOutEnable .....	384
6.90.	ECAT_GetPdInToOutEnable .....	386
6.91.	ECAT_SlaveNonBlockRegErrReadRequest .....	388
6.92.	ECAT_SlaveNonBlockRegErrReadState .....	392
6.93.	ECAT_SlaveNonBlockRegErClrRequest .....	397
6.94.	ECAT_SlaveNonBlockRegErrClrState .....	401
6.95.	ECAT_Get_2074A .....	405
7.	運動控制函式集說明 .....	408
7.1.	運動控制初始化 .....	408
7.1.1.	ECAT_McInIt .....	408
7.1.2.	ECAT_McInIt_Ext .....	411
7.2.	單軸運動控制設定 .....	413
7.2.1.	ECAT_McSetAxisDefaultMode .....	413
7.2.2.	ECAT_McGetAxisDefaultMode .....	415
7.2.3.	ECAT_McSetAxisServoOn .....	417
7.2.4.	ECAT_McSetAxisPPU .....	419
7.2.5.	ECAT_McGetAxisPPU .....	422
7.2.6.	ECAT_McSetAxisVelAccScale .....	424
7.2.7.	ECAT_McGetAxisVelAccScale .....	426
7.2.8.	ECAT_McSetProfileData .....	428
7.2.9.	ECAT_McGetProfileData .....	430
7.2.10.	ECAT_McSetProfileInterval .....	432
7.2.11.	ECAT_McSetProfileCSV .....	434
7.2.12.	ECAT_McGetProfileCSV .....	437
7.2.13.	ECAT_McSetAxisAccDecUnit .....	440
7.2.14.	ECAT_McGetAxisAccDecUnit .....	442
7.2.15.	ECAT_McSetAxisAccTime .....	444
7.2.16.	ECAT_McGetAxisAccTime .....	446
7.2.17.	ECAT_McSetAxisAccDecRate .....	448
7.2.18.	ECAT_McGetAxisAccDecRate .....	450
7.2.19.	ECAT_McSetAxisAccDecTime_Stepper .....	452
7.2.20.	ECAT_McGetAxisAccDecTime_Stepper .....	454
7.2.21.	ECAT_McSetAxisAccUnit_Stepper .....	456
7.2.22.	ECAT_McGetAxisAccUnit_Stepper .....	459
7.2.23.	ECAT_McSetAxisAccDecType .....	461

7.2.24.	ECAT_McGetAxisAccDecType .....	463
7.2.25.	ECAT_McSetAxisEncoderPPR .....	465
7.2.26.	ECAT_McGetAxisEncoderPPR .....	467
7.2.27.	ECAT_McSetAxisMotorPPR.....	469
7.2.28.	ECAT_McGetAxisMotorPPR .....	471
7.2.29.	ECAT_McSetEcamTable .....	473
7.2.30.	ECAT_McGetEcamTable .....	478
7.2.31.	ECAT_McConfigEcamTable .....	480
7.2.32.	ECAT_McSetAxisTouchProbeProperty .....	482
7.2.33.	ECAT_McGetAxisTouchProbeProperty .....	484
7.2.34.	ECAT_McGetAxisTouchProbeValue .....	486
7.2.35.	ECAT_McSetAxisVelocityFeedForwardGain .....	488
7.2.36.	ECAT_McGetAxisVelocityFeedForwardGain .....	490
7.2.37.	ECAT_McSetAxisPosSoftwareLimitStatus .....	492
7.2.38.	ECAT_McGetAxisPosSoftwareLimitStatus .....	496
7.2.39.	ECAT_McSetAxisPosSoftwareLimit .....	498
7.2.40.	ECAT_McGetAxisPosSoftwareLimit.....	500
7.2.41.	ECAT_OpenMotionConfig .....	502
7.2.42.	ECAT_McSetAxisMaxVelocity .....	505
7.2.43.	ECAT_McGetAxisMaxVelocity .....	507
7.2.44.	ECAT_McSetAxisDIActiveLevel .....	509
7.2.45.	ECAT_McGetAxisDIActiveLevel .....	512
7.2.46.	ECAT_McSetAxisActualPosition .....	514
7.2.47.	ECAT_McSetAxisActualPositionBy35 .....	516
7.2.48.	ECAT_McSetAxisCommandPosition.....	518
7.2.49.	ECAT_McSetAxisInpSignal .....	519
7.2.50.	ECAT_McGetAxisInpSignal.....	522
7.2.51.	ECAT_McSetAxisInpCompare .....	524
7.2.52.	ECAT_McGetAxisInpCompare .....	527
7.2.53.	ECAT_McSetAxisInpTimeOut .....	529
7.2.54.	ECAT_McGetAxisInpTimeOut .....	531
7.2.55.	ECAT_McSetAxisWanErrEnable .....	533
7.2.56.	ECAT_McGetAxisWanErrEnable .....	535
7.2.57.	ECAT_McEnable_Directly_Ex .....	537
7.3.	單軸運動狀態.....	539
7.3.1.	ECAT_McGetAxisActualPos.....	539
7.3.2.	ECAT_McGetAxisActualPos_Ex .....	542
7.3.3.	ECAT_McGetAxisActualPos_Directly.....	543



7.3.4.	ECAT_McGetAxisCommandPos .....	544
7.3.5.	ECAT_McGetAxisCommandPos_Ex .....	547
7.3.6.	ECAT_McGetAxisCommandPos_Directly .....	548
7.3.7.	ECAT_McGetAxisActualVel.....	549
7.3.8.	ECAT_McGetAxisActualVel_Ex .....	552
7.3.9.	ECAT_McGetAxisActualPosVel_Ex .....	553
7.3.10.	ECAT_McGetAxisActualTorque .....	556
7.3.11.	ECAT_McGetAxisState .....	558
7.3.12.	ECAT_McGetAxisState_Ex .....	561
7.3.13.	ECAT_McGetAxisState_Directly .....	562
7.3.14.	ECAT_McGetAxisLastError .....	563
7.3.15.	ECAT_McGetAxisLastError_Ex .....	565
7.3.16.	ECAT_McGetAxisDriveError .....	566
7.3.17.	ECAT_McGetAxisDriveError_Ex .....	569
7.3.18.	ECAT_McGetAxisDI .....	570
7.3.19.	ECAT_McGetAxisDI_Ex .....	573
7.3.20.	ECAT_McGetAxisDI_Directly .....	574
7.3.21.	ECAT_McGetAxisDI_60FD.....	575
7.3.22.	ECAT_McGetAxisHomeState.....	577
7.4.	<b>單軸自動原點復歸.....</b>	<b>579</b>
7.4.1.	ECAT_McSetAxisHomeMethod .....	579
7.4.2.	ECAT_McGetAxisHomeMethod .....	583
7.4.3.	ECAT_McSetAxisHomeSpeed .....	585
7.4.4.	ECAT_McGetAxisHomeSpeed.....	587
7.4.5.	ECAT_McSetAxisHomeAcc.....	589
7.4.6.	ECAT_McGetAxisHomeAcc .....	591
7.4.7.	ECAT_McSetAxisHomeOffset.....	593
7.4.8.	ECAT_McGetAxisHomeOffset.....	595
7.4.9.	ECAT_McSetAxisHomeTorque .....	597
7.4.10.	ECAT_McGetAxisHomeTorque.....	599
7.4.11.	ECAT_McSetAxisHomeStable .....	601
7.4.12.	ECAT_McGetAxisHomeStable .....	603
7.4.13.	ECAT_McAxisHome .....	605
7.4.14.	ECAT_McAxisHomeEx.....	608
7.5.	<b>單軸運動控制.....</b>	<b>611</b>
7.5.1.	ECAT_McAxisErrorReset .....	611
7.5.2.	ECAT_McAxisMoveAbs.....	613
7.5.3.	ECAT_McAxisMoveRel .....	616

7.5.4.	ECAT_McAxisMoveAbs_P2P .....	619
7.5.5.	ECAT_McAxisMoveRel_P2P .....	622
7.5.6.	ECAT_McAxisChangePos .....	625
7.5.7.	ECAT_McAxisChangeVel .....	628
7.5.8.	ECAT_McAxisMoveSuperimposed .....	631
7.5.9.	ECAT_McAxisHaltSuperimposed .....	635
7.5.10.	ECAT_McAxisMoveVel .....	638
7.5.11.	ECAT_McAxisMoveVelEx .....	640
7.5.12.	ECAT_McAxisMoveVelByPos .....	642
7.5.13.	ECAT_McAxisMoveTor .....	644
7.5.14.	ECAT_McAxisMoveTorEx .....	646
7.5.15.	ECAT_McAxisGearIn .....	649
7.5.16.	ECAT_McAxisGearOut .....	652
7.5.17.	ECAT_McAxisGearInByPos .....	654
7.5.18.	ECAT_McAxisMoveProfile .....	658
7.5.19.	ECAT_McAxisMoveProfileCSV .....	660
7.5.20.	ECAT_McAxisCamIn .....	662
7.5.21.	ECAT_McAxisCamPhaseShift .....	666
7.5.22.	ECAT_McAxisCamOut .....	671
7.5.23.	ECAT_McAxisGantryIn .....	673
7.5.24.	ECAT_McAxisGantryMaxPosDiff .....	677
7.5.25.	ECAT_McAxisGantryMaxPosDiffStatus .....	680
7.5.26.	ECAT_McAxisGantryGain .....	683
7.5.27.	ECAT_McAxisGantryOut .....	687
7.5.28.	ECAT_McAxisMoveAbsAdv_Ex .....	689
7.5.29.	ECAT_McAxisMoveRelAdv_Ex .....	693
7.5.30.	ECAT_McAxisMove_CiA402_PP .....	697
7.5.31.	ECAT_McAxisMove_CiA402_PV .....	700
7.5.32.	ECAT_McAxisMove_CiA402_PT .....	703
7.5.33.	ECAT_McAxisStop .....	706
7.5.34.	ECAT_McAxisQuickStop .....	709
7.6.	群組運動控制設定 .....	712
7.6.1.	ECAT_McAddAxisToGroup_Ex .....	712
7.6.2.	ECAT_McRemoveAxisFromGroup_Ex .....	714
7.6.3.	ECAT_McUngroupAllAxes_Ex .....	716
7.6.4.	ECAT_McSetGroupCmdMode_Ex .....	718
7.6.5.	ECAT_McSetGroupCmdModeEx_Ex .....	720
7.6.6.	ECAT_McGetGroupCmdMode .....	722

7.6.7.	ECAT_McSetGroupAccTime_Ex .....	724
7.6.8.	ECAT_McSetGroupAccTimeEx .....	726
7.6.9.	ECAT_McGetGroupAccTime .....	728
7.6.10.	ECAT_McSetGroupAccDecType_Ex .....	730
7.6.11.	ECAT_McGetGroupAccDecType .....	732
7.6.12.	ECAT_McSetGroupBlendingPercent_Ex .....	734
7.6.13.	ECAT_McSetGroupBlendingPercentEx_Ex .....	737
7.6.14.	ECAT_McSetGroupPvtDecEnable_Ex .....	741
7.6.15.	ECAT_McGetGroupPvtDecEnable .....	743
7.6.16.	ECAT_McSetGroupCoordinate_Ex .....	745
7.6.17.	ECAT_McSetGroupCoordinateLimit_Ex .....	749
7.6.18.	ECAT_McGetGroupCoordinateLimit_Ex .....	751
7.7.	<b>群組運動狀態</b> .....	753
7.7.1.	ECAT_McGetGroupState .....	753
7.7.2.	ECAT_McGetGroupPauseState .....	756
7.7.3.	ECAT_McGetGroupCmdBuffer .....	758
7.7.4.	ECAT_McSetGroupVelLimitStatus_Ex .....	760
7.7.5.	ECAT_McGetGroupVelLimitStatus .....	762
7.7.6.	ECAT_McSetGroupVelLimitValue_Ex .....	764
7.7.7.	ECAT_McGetGroupVelLimitValue .....	766
7.8.	<b>群組運動控制</b> .....	768
7.8.1.	ECAT_McGroupMoveLineAbs_Ex .....	768
7.8.2.	ECAT_McGroupMoveLineRel_Ex .....	771
7.8.3.	ECAT_McGroupMoveLineAbs_PT_Ex .....	774
7.8.4.	ECAT_McGroupMoveLineRel_PT_Ex .....	777
7.8.5.	ECAT_McGroupMoveLineAbs_PVT .....	780
7.8.6.	ECAT_McGroupMoveLineRel_PVT .....	784
7.8.7.	ECAT_McGroupMoveLineAbs_P2P .....	788
7.8.8.	ECAT_McGroupMoveLineRel_P2P .....	791
7.8.9.	ECAT_McGroupMoveLineAbs_PTexT .....	794
7.8.10.	ECAT_McGroupMoveLineRel_PTexT .....	798
7.8.11.	ECAT_McGroupMoveLineAbs_PPT .....	802
7.8.12.	ECAT_McGroupMoveLineRel_PPT .....	806
7.8.13.	ECAT_McGroupMoveCircularAbs_CP_Angle_Ex .....	810
7.8.14.	ECAT_McGroupMoveCircularRel_CP_Angle_Ex .....	814
7.8.15.	ECAT_McGroupMoveCircularAbs_CP_EP_Ex .....	817
7.8.16.	ECAT_McGroupMoveCircularRel_CP_EP_Ex .....	821
7.8.17.	ECAT_McGroupMoveCircularAbs_BP_EP_Ex .....	824

7.8.18.	ECAT_McGroupMoveCircularRel_BP_EP_Ex .....	828
7.8.19.	ECAT_McGroupMove3DCircularAbs_CP_Angle_Ex .....	831
7.8.20.	ECAT_McGroupMove3DCircularRel_CP_Angle_Ex .....	835
7.8.21.	ECAT_McGroupMove3DCircularAbs_CP_EP_Ex.....	838
7.8.22.	ECAT_McGroupMove3DCircularRel_CP_EP_Ex.....	842
7.8.23.	ECAT_McGroupMove3DCircularAbs_BP_EP_Ex.....	845
7.8.24.	ECAT_McGroupMove3DCircularRel_BP_EP_Ex.....	849
7.8.25.	ECAT_McGroupMoveHelicalAbs_Ex .....	852
7.8.26.	ECAT_McGroupMoveHelicalRel_Ex .....	857
7.8.27.	ECAT_McGroupMove3DHelicalAbs_CP_Angle_Ex.....	860
7.8.28.	ECAT_McGroupMove3DHelicalRel_CP_Angle_Ex.....	865
7.8.29.	ECAT_McGroupMoveConicalHelixAbs_Ex .....	868
7.8.30.	ECAT_McGroupMoveConicalHelixRel_Ex .....	872
7.8.31.	ECAT_McGroupMove3DConicalHelixAbs_CP_Angle_Ex.....	875
7.8.32.	ECAT_McGroupMove3DConicalHelixRel_CP_Angle_Ex .....	879
7.8.33.	ECAT_McGroupMoveProfile_Ex .....	882
7.8.34.	ECAT_McGroupMoveProfileCSV_Ex.....	885
7.8.35.	ECAT_McGroupMoveDwell_Ex.....	888
7.8.36.	ECAT_McGroupMoveDO .....	892
7.8.37.	ECAT_McGroupMoveAO .....	896
7.8.38.	ECAT_McGroupMoveBlendingSync_Ex .....	900
7.8.39.	ECAT_McGroupStop .....	904
7.8.40.	ECAT_McGroupQuickStop .....	907
7.8.41.	ECAT_McSetGroupHold_Ex.....	910
7.8.42.	ECAT_McSetGroupPause_Ex.....	914
7.8.43.	ECAT_McAddPathData .....	918
7.8.44.	ECAT_McSetPathData .....	927
7.8.45.	ECAT_McGetPathData.....	930
7.8.46.	ECAT_McClearPathData.....	931
7.8.47.	ECAT_McGetPathDataSize.....	934
7.8.48.	ECAT_McGroupMovePath .....	936
7.8.49.	ECAT_McGroupMoveLineAbsAdv_Ex .....	939
7.8.50.	ECAT_McGroupMoveLineRelAdv_Ex .....	944
7.8.51.	ECAT_McGroupMoveShaker_Ex .....	949
7.8.52.	ECAT_McAxisTangentInGroup .....	952
7.8.53.	ECAT_McAxisTangentOut.....	961
7.9.	PID 控制器.....	963
7.9.1.	ECAT_PidGetSetPointValue .....	964

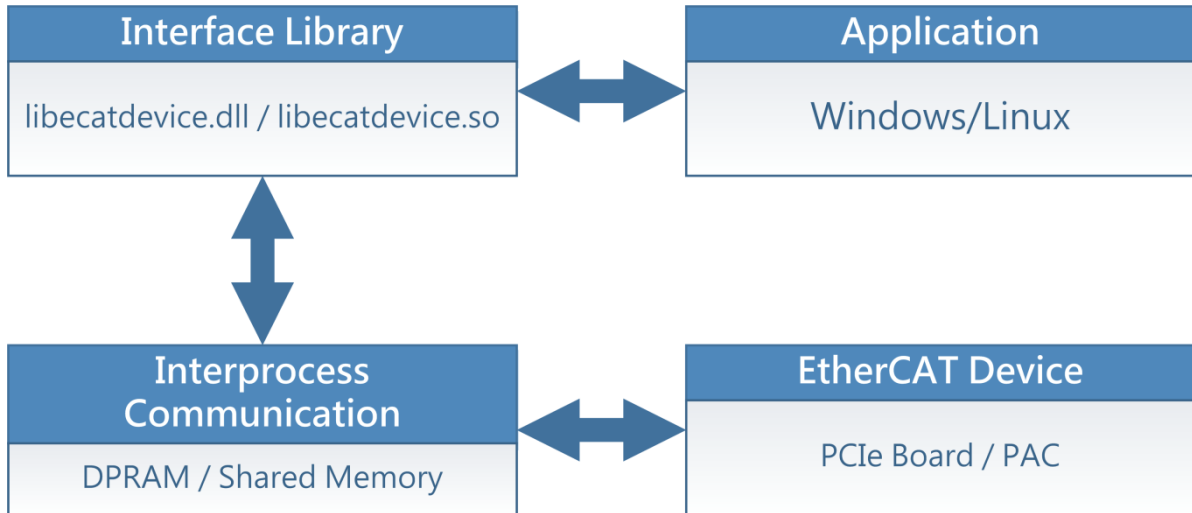
7.9.2.	ECAT_PidSetSetPointValue .....	966
7.9.3.	ECAT_PidGetProcessVariable .....	968
7.9.4.	ECAT_PidGetSampleTime .....	970
7.9.5.	ECAT_PidSetSampleTime .....	972
7.9.6.	ECAT_PidGetStatus .....	974
7.9.7.	ECAT_PidSetStatus .....	976
7.9.8.	ECAT_PidGetSimulateMode .....	978
7.9.9.	ECAT_PidSetSimulateMode.....	980
7.9.10.	ECAT_PidGetParameter .....	982
7.9.11.	ECAT_PidSetParameter .....	984
7.9.12.	ECAT_PidGetProcessVariableModule .....	986
7.9.13.	ECAT_PidSetProcessVariableModule .....	988
7.9.14.	ECAT_PidGetControlOutputModule .....	991
7.9.15.	ECAT_PidSetControlOutputModule .....	993
7.9.16.	ECAT_PidGetControlOutputValue .....	996
7.9.17.	ECAT_PidGetSimulateFeedback .....	999
7.9.18.	ECAT_PidGet_Sp_Err_Op_Pv.....	1002
7.10.	史都華平台 .....	1005
7.10.1.	ECAT_McSetStewartPlatform_M1 .....	1007
7.10.2.	ECAT_McGetStewartPlatform_M1 .....	1009
7.10.3.	ECAT_McSetStewartPlatform_M2 .....	1011
7.10.4.	ECAT_McGetStewartPlatform_M2 .....	1013
7.10.5.	ECAT_McStewartPlatformMoveAbs_PT .....	1015
7.11.	運動記錄 .....	1019
7.11.1.	ECAT_McSetMotionRecord.....	1019
7.11.2.	ECAT_McGetMotionRecordState.....	1021
7.11.3.	ECAT_McClearMotionRecord .....	1023
7.11.4.	ECAT_McSetMotionRecordParam .....	1025
7.11.5.	ECAT_McGetMotionRecordParam .....	1027
7.11.6.	ECAT_McGetMotionRecordValue_Ex.....	1029
7.11.7.	ECAT_McGetMotionRecordValueEx_Ex.....	1033
7.12.	事件觸發 .....	1036
7.12.1.	ECAT_EvEnableEvent.....	1037
7.12.2.	ECAT_EvDisableEvent.....	1039
7.12.3.	ECAT_WaitforEvent.....	1041
7.12.4.	ECAT_AbortWaitforEvent.....	1043
7.12.5.	ECAT_EvSetComparePositionParameters .....	1045
7.12.6.	ECAT_EvSetCompareCmdPositionParameters.....	1047

7.12.7.	ECAT_EvSetCompareDIBitParameters .....	1049
7.12.8.	ECAT_EvSetCompareDIParameters .....	1051
7.12.9.	ECAT_EvSetCompareAxisStateParameters .....	1053
7.12.10.	ECAT_EvSetMotionCompleteParameters .....	1055
7.12.11.	ECAT_EvSetMotionCompleteParameters_Grp .....	1059
7.12.12.	ECAT_EvSetCompareAxisVelStateParameters .....	1063
7.12.13.	ECAT_EvSetCompareAiParameters .....	1068
8.	附錄 .....	1072
8.1.	函式錯誤回傳代碼 .....	1072
8.2.	SDO 操作錯誤終止代碼 .....	1080
8.3.	手冊修訂記錄 .....	1082
8.4.	如何關閉 Windows 10/11 中的快速啟動功能 .....	1083
8.5.	CiA402 原點復歸模式(Homing Mode/ hm mode) .....	1086
8.5.1.	Method 1 .....	1086
8.5.2.	Method 2 .....	1087
8.5.3.	Method 3, 4 .....	1088
8.5.4.	Method 5, 6 .....	1089
8.5.5.	Method 7, 8, 9, 10 .....	1090
8.5.6.	Method 11, 12, 13, 14 .....	1091
8.5.7.	Method 17 .....	1092
8.5.8.	Method 18 .....	1093
8.5.9.	Method 19, 20 .....	1094
8.5.10.	Method 21, 22 .....	1095
8.5.11.	Method 23, 24, 25, 26 .....	1096
8.5.12.	Method 27, 28, 29, 30 .....	1097
8.5.13.	Method 33, 34 .....	1098
8.5.14.	Method 35, 37 .....	1099
8.6.	CiA402 馬達每轉之解析度與電子齒輪比 .....	1100
8.6.1.	驅動器內部參數 .....	1100
8.6.2.	EtherCAT 主站參數 .....	1101
8.7.	CiA402 驅動器馬達運轉方向設定 .....	1103
8.8.	CiA402 驅動器 EEPROM 存檔 .....	1104
8.9.	CiA402 驅動器重要參數 .....	1105
8.9.1.	減速度 .....	1105
8.10.	ECAT-2091S/ ECAT-2094S 使用注意事項 .....	1106
8.10.1.	六線式馬達 .....	1106
8.10.2.	重要參數 .....	1107



# 1. 簡介

EtherCAT MainDevice 系列裝置泓格提供使用者可調用的共享函式庫 libecatdevice(.dll) , 讓使用者能夠在 Windows 作業系統下使用。函式庫提供開發容易及功能強大的功能，讓用戶能夠快速的上手來使用，共享函式庫架構如下圖所示。



## 1.1. 版本更新資訊

功能修改	對應版本
1. 新增 Cam Utility	V1.0.08 以上
1. 修正使用 ECAT2610 時可以進 OP，卻無法操作 PDO 的 BUG 2. 支援多軸 CiA402 驅動器 3. 支援 CiA402 Profile 運動模式(PP、PV、PT) 4. 支援三種二階軟體濾波器(低通、高通以及陷波) 5. 支援 Slave to Slave communication(topology independent)	V1.0.15 以上
1. 修改 PVT 相關算法 2. 新增 PVT 減速停止 3. 支援 Motion Done Event 4. 支援以 Alias 操作從站 5. 修改 Gantry 為交叉耦合式	V1.0.16 以上
1. 新增進階 PDO 編輯	V1.0.17



2. 新增 CiA402 Mapping Mode: User Define	以上
1. 新增 ECAT_AbortWaitforEvent 2. 修正某些模組使用 ECAT_GetSlaveInfo 讀不到模組名稱的問題 3. 修改 ECAT-2092T 部分 API、新增 ECAT-2092T Array compare 相關 API 4. 新增座標轉換(直角坐標命令到極座標機構)功能 5. 新增設定 CiA402 驅動器 DI active level API 6. 新增讀取 CiA402 驅動器 DI(object 0x60fd) API 7. 修正連續下寫 DO 時命令被後一個命令覆蓋之 BUG	V1.0.18 以上
1. 新增支援 ECAT-2093 的鎖存緩衝區 2. 新增速度與加速度單位轉換參數 3. 新增定位偵測 4. 新增非對稱加減速(供 ECAT-2091S、ECAT-2094S)	V1.0.19 以上
1. 增加對多模組同時讀/寫 DO	V1.0.22 以上
1. 支援多 Domain	V1.0.23 以上
1. 改善通訊問題	V1.0.24 以上
1. 減少部分函式所花費的時間 2. 單軸運動可指定加速度單位為時間或是 unit/s^2 3. 修改龍門演算法	V1.0.25 以上
1. Tangent In 函式改為在群組命令緩衝區排隊 2. 加入 Tangent Out 函式 3. MovePath 支援 Tangent In/Out 函式 4. 支援無限旋轉功能 5. 支援飛剪功能 6. 支援 ECAT-2016-3 模組	V1.0.26 以上
1. 支援 ECAT-2074A 模組 2. 支援龍門歸原點 3. 新增 Home 穩定條件設定 4. 可設定 DO 預設值 5. 減少部分 API 花費時間	V1.0.30 以上

## 2. 軟體安裝

本章節將詳細介紹如何取得安裝執行檔、安裝及移除驅動程式及工具程式步驟。

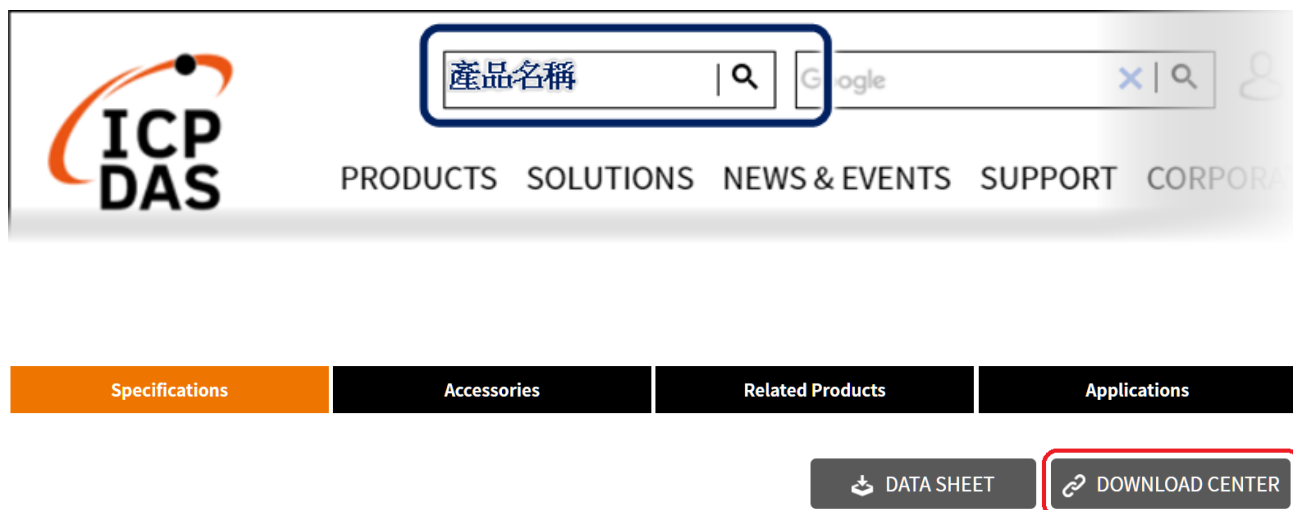
### 2.1. 取得軟體安裝程式

用戶如何在 ICP DAS 網站上搜索 driver、手冊和規格：

- 行動版網站



- 一般網站

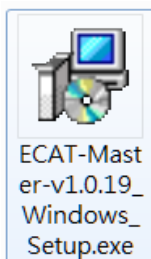


## 2.2. 安裝 Windows 驅動程式

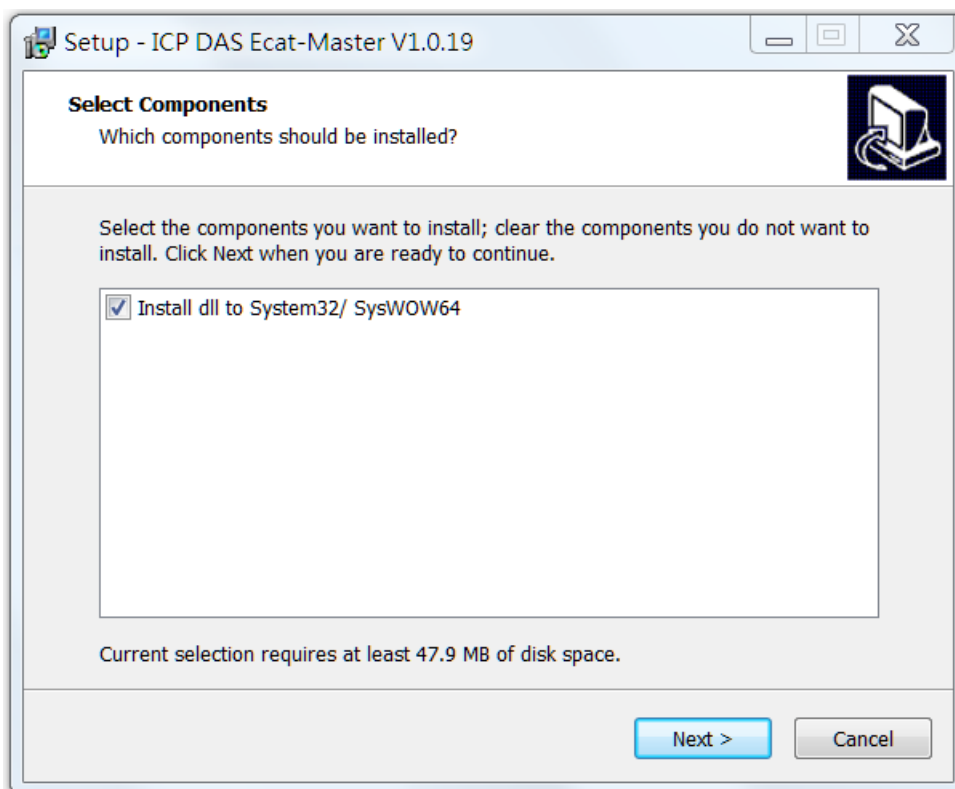
此軟體包含驅動程式、API 與 Utility

依照下列步驟來完成 EcatDevice 驅動程式安裝:

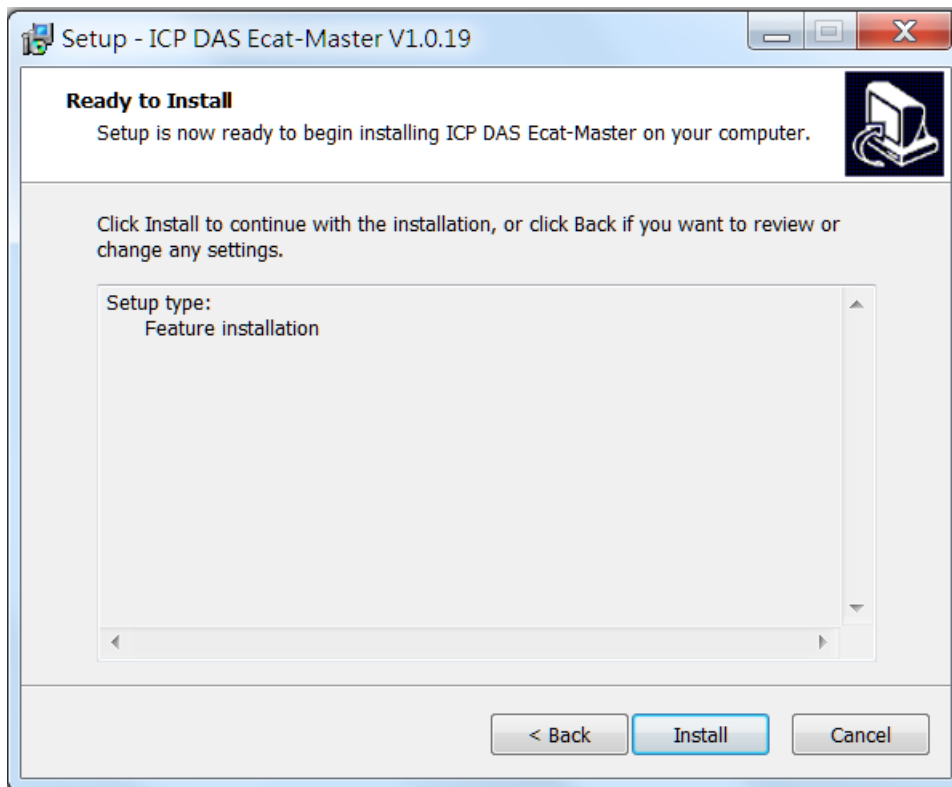
1. 雙擊 ECAT-Master\_vx.x.xx\_Windows\_setup.exe 驅動程式安裝執行檔



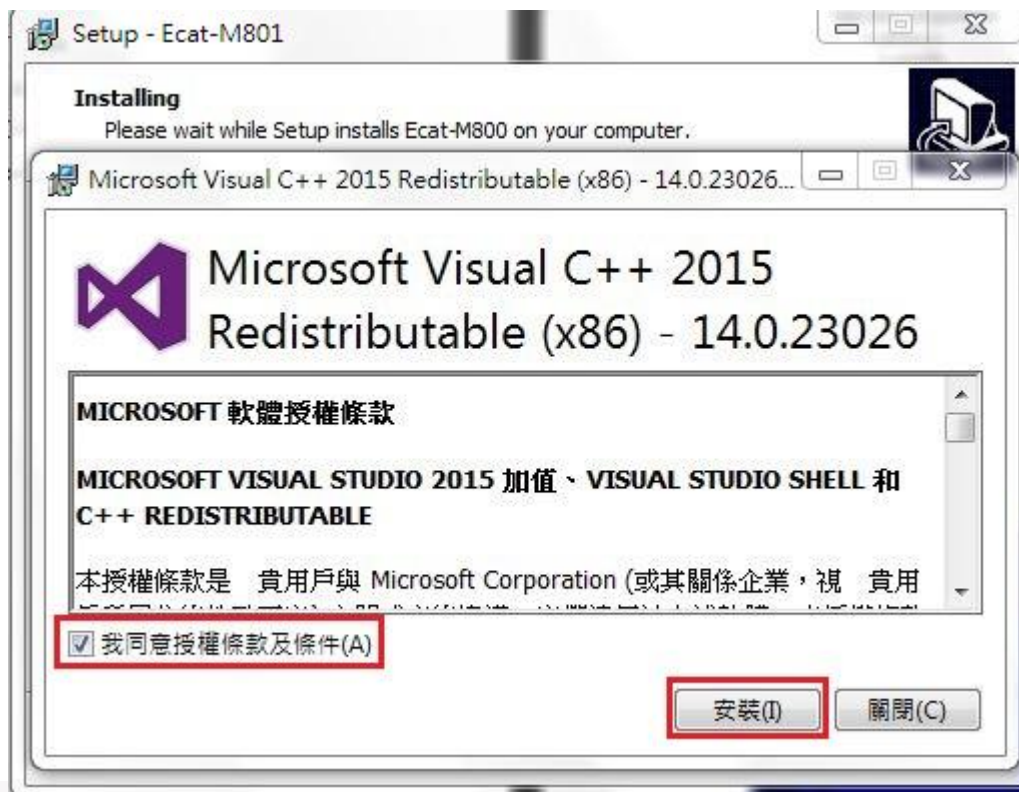
2. 選擇是否安裝 dll 到 System32/ SysWOW64 , 按 “Next >” 按鈕到下一個安裝畫面。



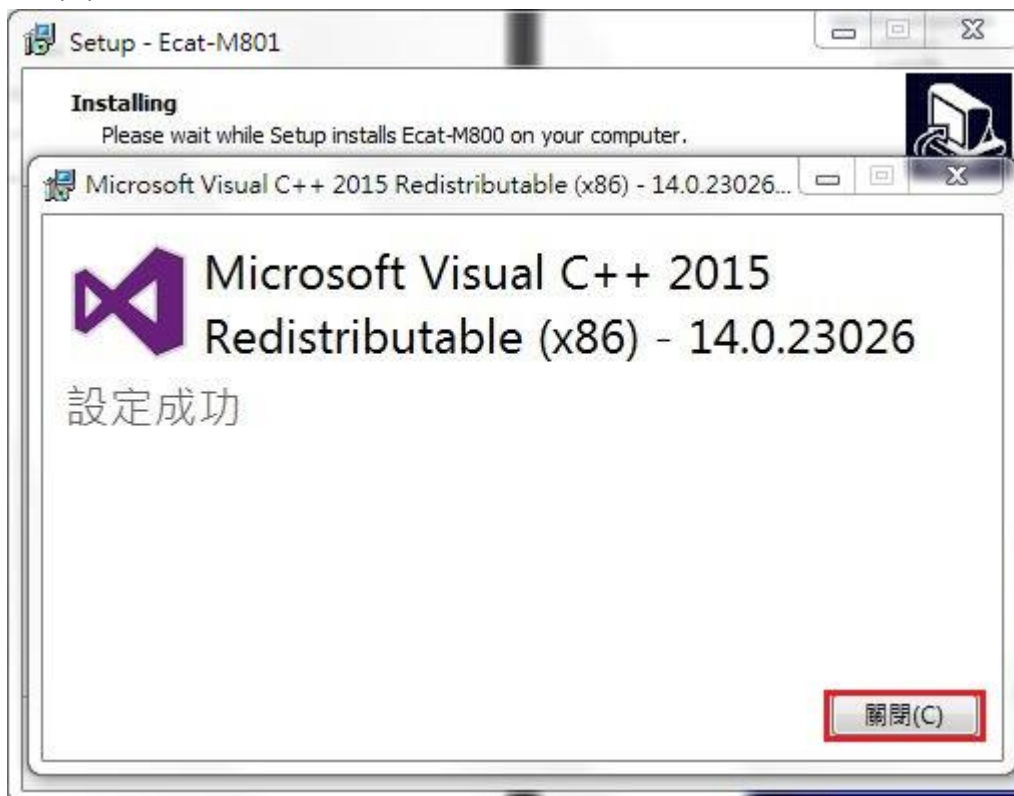
3. 預設安裝路徑 C:\icpdas\Ecat-M801。按 “Install” 按鈕開始安裝。



4. 勾選 “我同意授權條款及條件(A)”，按 “安裝” 按鈕開始安裝。  
若執行驅動程式安裝檔卻未出現以下畫面，代表不用安裝。



5. 點擊“關閉(C)”按鈕

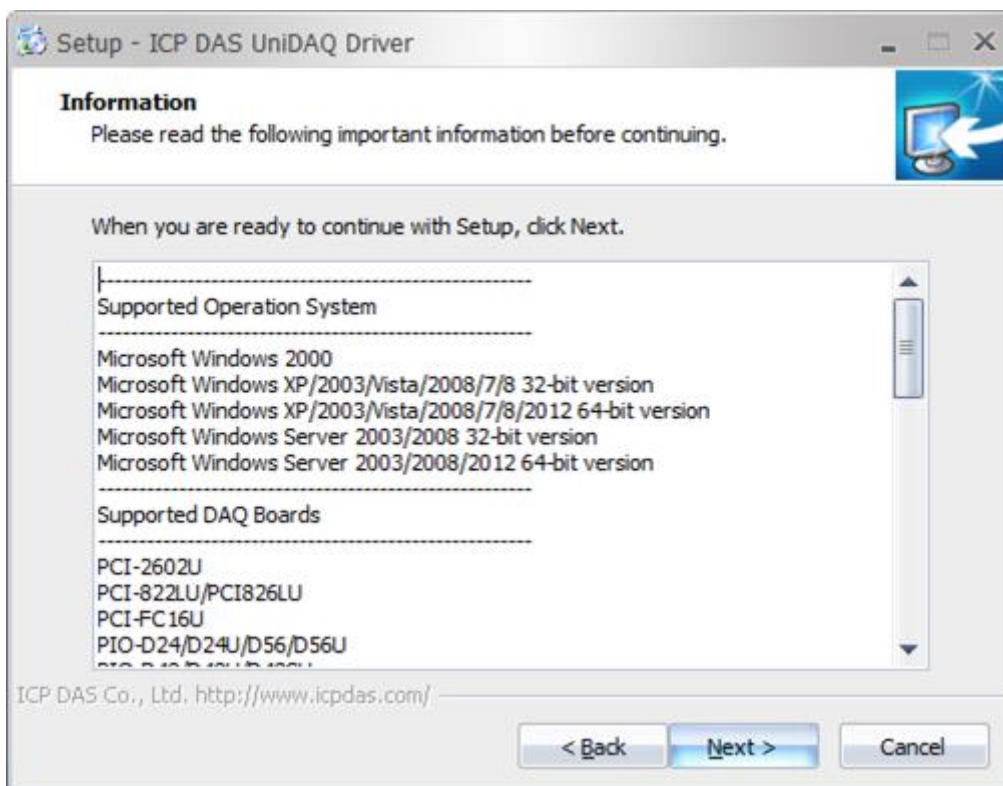


6. 點擊 Next>

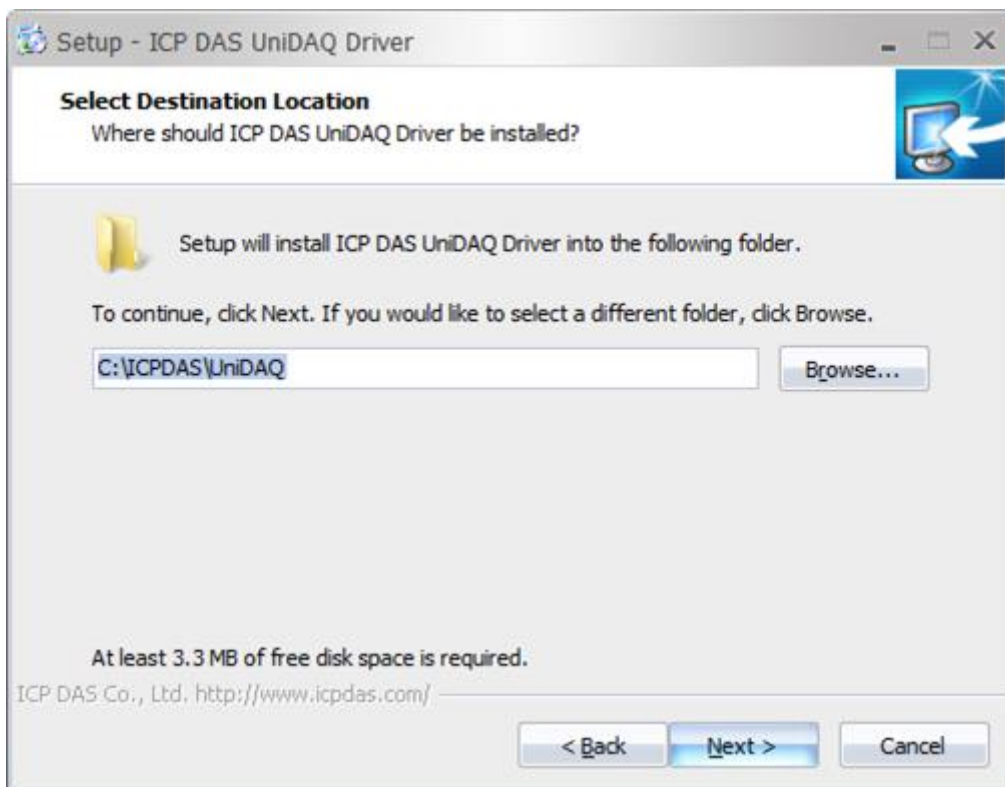
注: V1.0.18 或以下的版本才需要 UniDAQ Driver , 若執行驅動程式安裝檔卻未出現以下畫面, 代表不用安裝



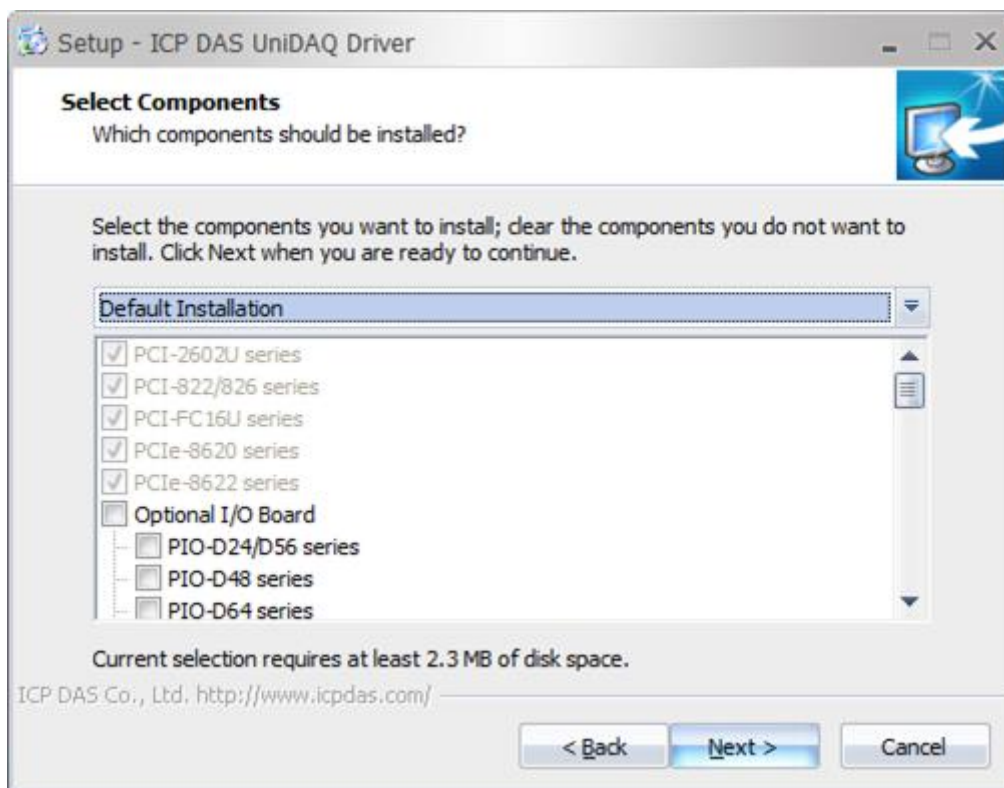
7. 點擊 Next>



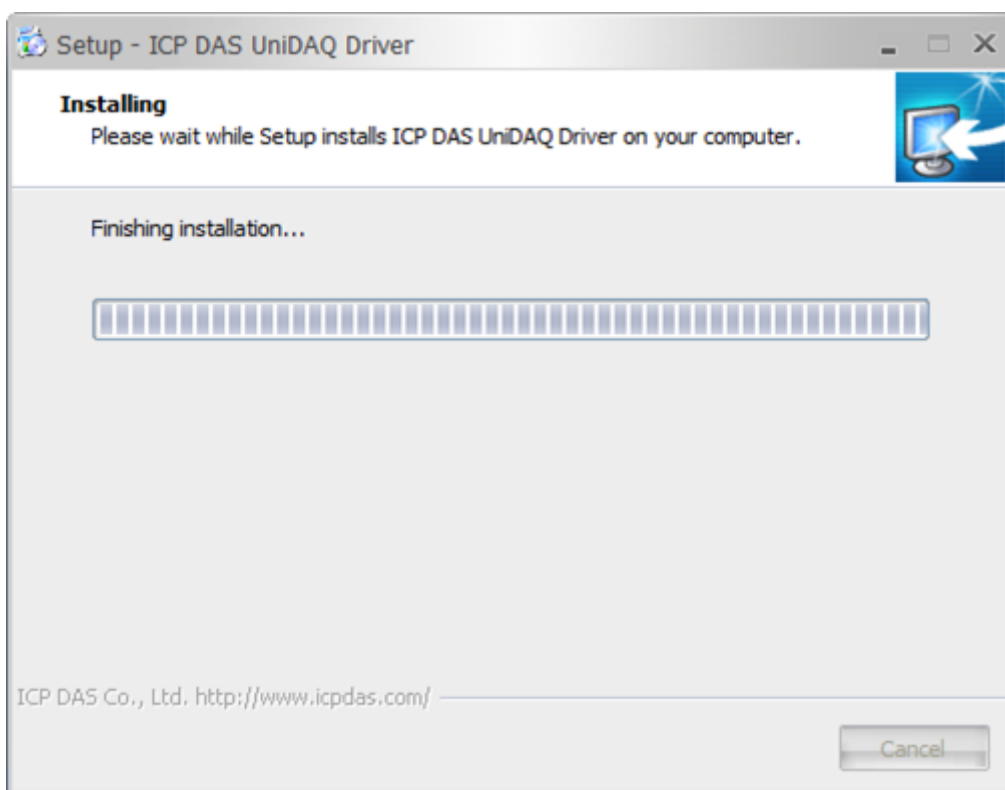
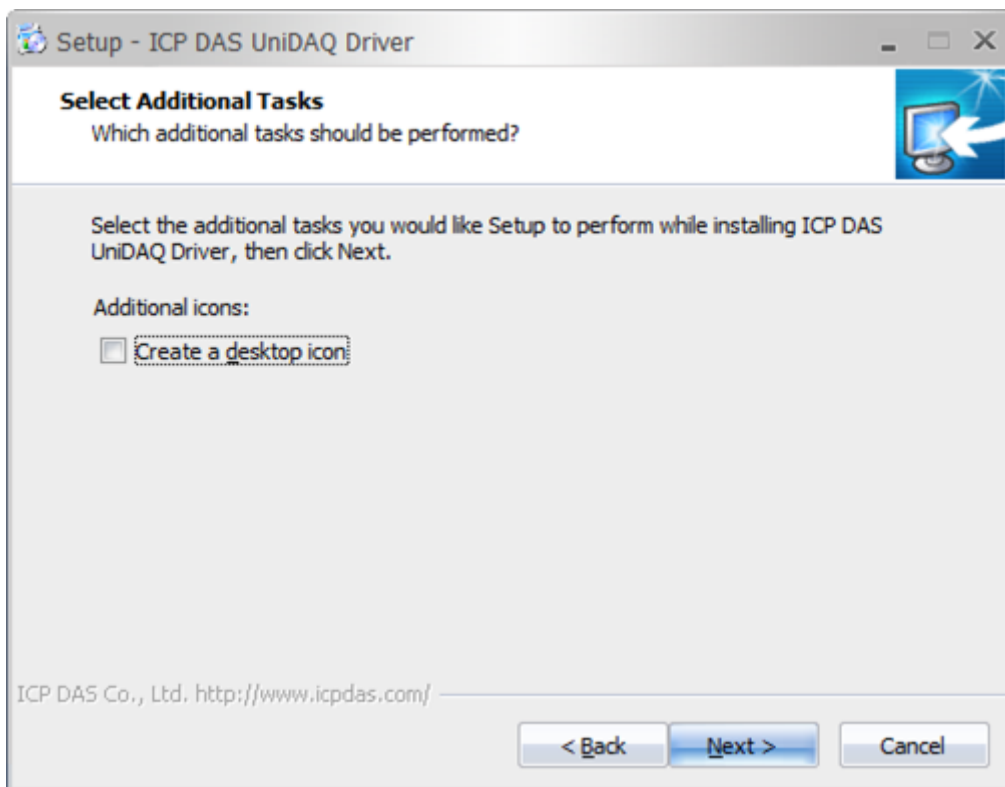
8. 點擊 Next>



9. 點擊 Next>

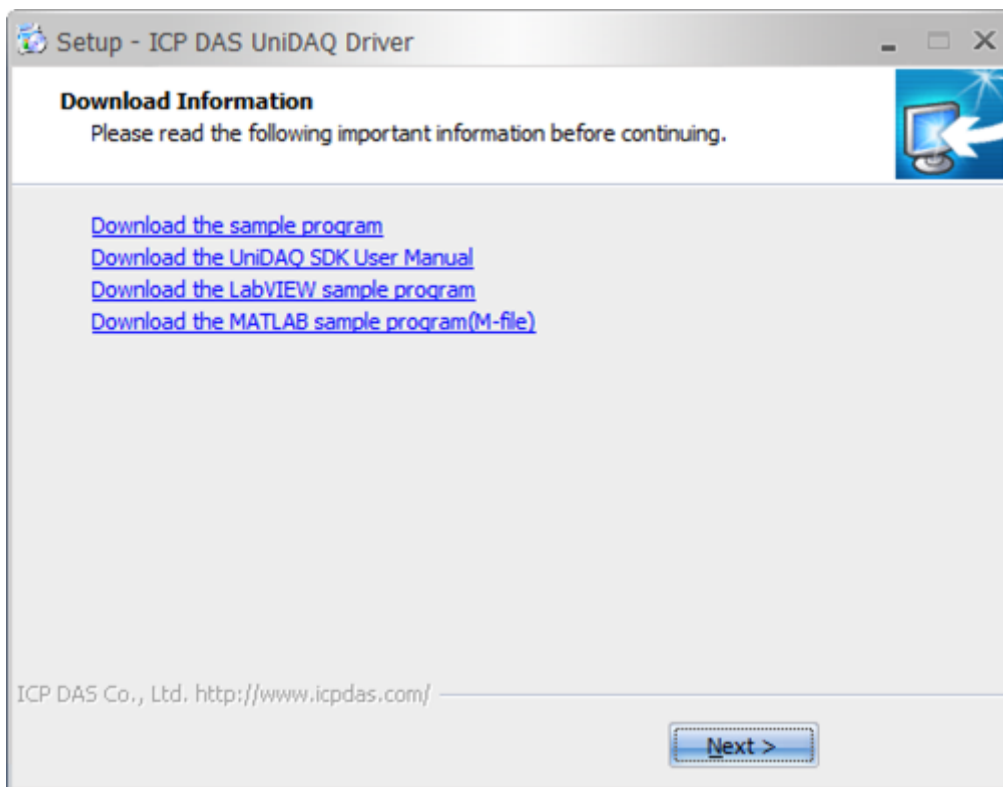


10. 點擊 Next>



11. 點擊 Next>





12. 點擊 Finish



13. 按“Finish”按鈕開完成安裝。



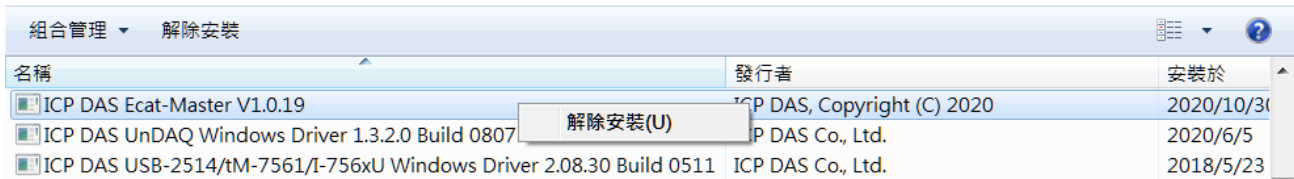
## 2.3. 移除 Windows 驅動程式

驅動程式包括反安裝工具來協助您從電腦上移除軟體，如果您想要移除軟體請完成下列的步驟來執行反安裝工具。

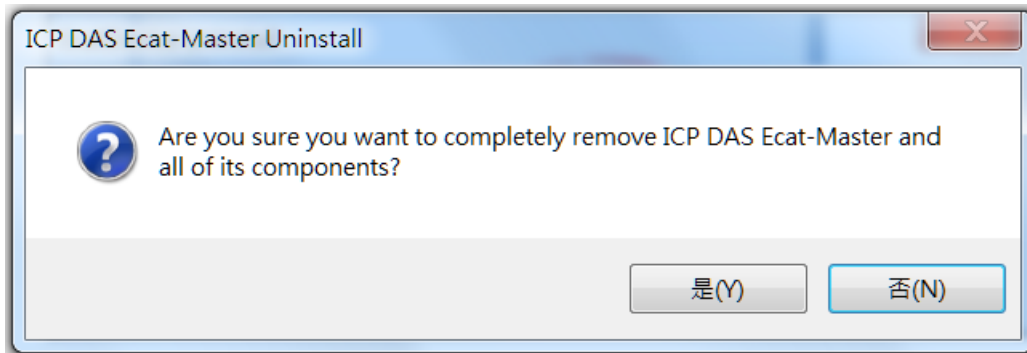
1. 從開始功能表中點選 控制台 >> 解除安裝程式。
2. 在程式清單中點選 ICP DAS Ecat-Master Vx.x.xx 項目，點選解除安裝。

### 解除安裝或變更程式

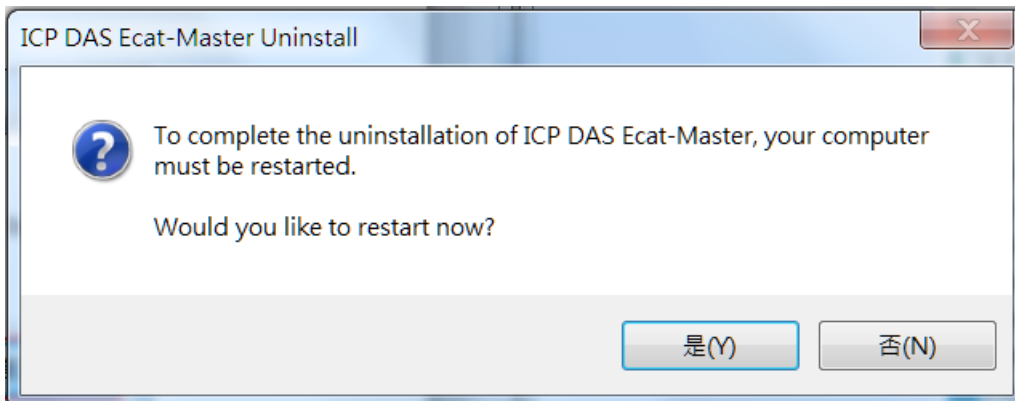
若要解除安裝程式，請從清單選取程式，然後按一下 [解除安裝]、[變更] 或 [修復]。



3. 將會跳出對話框來詢問是否確定要移除此軟體驅動程式，請按下“是(Y)”按鈕開始執行反安裝



4. 為了完成解除安裝，請按下“是(Y)”按鈕，重新啟動後完成移除。





## 2.4. 安裝 Linux 驅動程式

### 2.4.1. 安裝 Linux 驅動程式

支援 Ubuntu 20.04LTS、22.04LTS

#### 1. 解壓縮 “ecat\_m801\_linux\_setup\_vx.xx.xx.tar.gz” 檔

```
bryan@icpdas-mint-ibpc:~/workspace/gg$ tar xvf ecat_m801_linux_setup_v1.0.15.tar.gz
./ecat_m801_linux_setup/
./ecat_m801_linux_setup/libtool
./ecat_m801_linux_setup/.cproject
./ecat_m801_linux_setup/Makefile.am
./ecat_m801_linux_setup/drivers/
./ecat_m801_linux_setup/drivers/_ecat.c
./ecat_m801_linux_setup/drivers/Makefile.am
./ecat_m801_linux_setup/drivers/Makefile
./ecat_m801_linux_setup/drivers/ixecat.remove
./ecat_m801_linux_setup/drivers/ixecat.inst
./ecat_m801_linux_setup/drivers/Makefile.in
./ecat_m801_linux_setup/drivers/_proc.c
./ecat_m801_linux_setup/drivers/Kbuild
./ecat_m801_linux_setup/drivers/_pciecat.c
./ecat_m801_linux_setup/drivers/Kbuild.in
./ecat_m801_linux_setup/Makefile
./ecat_m801_linux_setup/COPYING
./ecat_m801_linux_setup/m4/
./ecat_m801_linux_setup/m4/ltversion.m4
./ecat_m801_linux_setup/m4/ltoptions.m4
./ecat_m801_linux_setup/m4/lt~obsolete.m4
./ecat_m801_linux_setup/m4/ltsugar.m4
./ecat_m801_linux_setup/m4/libtool.m4
./ecat_m801_linux_setup/aclocal.m4
./ecat_m801_linux_setup/README
./ecat_m801_linux_setup/.settings/
./ecat_m801_linux_setup/.settings/language.settings.xml
./ecat_m801_linux_setup/autom4te.cache/
```

#### 2. 進入解壓縮後目錄，並輸入“./configure”

若出現 configure: error: no acceptable C compiler found in\$PATH

請執行 “sudo apt-get install build-essential”

```

bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking for g++... g++
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking dependency style of g++... gcc3
checking for ar... ar
checking the archiver (ar) interface... ar
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking how to print strings... printf
checking for a sed that does not truncate output... /bin/sed

```

```

rd5@rd5-VirtualBox:~/ecat_m801_linux_setup$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... no
checking whether make supports nested variables... no
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: in '/home/rd5/ecat_m801_linux_setup':
configure: error: no acceptable C compiler found in $PATH
See 'config.log' for more details

```

### 3. 輸入“make modules”

```

config.status: creating script/remove
config.status: creating script/Makefile
config.status: creating drivers/Kbuild
config.status: creating drivers/Makefile
config.status: creating include/Makefile
config.status: creating lib/Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ make modules
make -C "/lib/modules/" uname -r "/build" M="/home/bryan/workspace/gg/ecat_m801_linux_setup" modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-20-generic'
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel"
CC [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/_ecat.o
CC [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/_proc.o
LD [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixecat.o
CC [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/_pciecat.o
LD [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixpciecat.o
Building modules, stage 2.
MODPOST 2 modules
CC /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixecat.mod.o
LD [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixecat.ko
CC /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixpciecat.mod.o
LD [M] /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixpciecat.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-20-generic'
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$

```

#### 4. 輸入“sudo make install”進行安裝，預設安裝於/opt/icpdas/ecat\_m801 目錄底下。

若出現 autoheader: command not found 訊息，請執行“sudo apt-get install autoconf”後再次執行“sudo make install”

```

bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ sudo make install
[sudo] password for bryan:
Making install in script
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[2]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/opt/icpdas/ecat_m801/share/script'
/usr/bin/install -c ecat_m801 ecat_m801.conf remove '/opt/icpdas/ecat_m801/share/script'
make install-data-hook
make[3]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
cp /opt/icpdas/ecat_m801/share/script/ecat_m801 /etc/init.d/
update-rc.d ecat_m801 defaults
cp /opt/icpdas/ecat_m801/share/script/ecat_m801.conf /etc/ld.so.conf.d/
make[3]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
Making install in drivers
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make install-am
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[3]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[3]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/opt/icpdas/ecat_m801/drivers'
/usr/bin/install -c -m 644 ixecat.ko ixpciecat.ko '/opt/icpdas/ecat_m801/drivers'
/bin/mkdir -p '/opt/icpdas/ecat_m801/drivers'
/usr/bin/install -c ixecat.inst ixecat.remove '/opt/icpdas/ecat_m801/drivers'
make[3]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'

```

```

/home/rd5/ecat_m801_linux_setup/autoconf/missing: line 81: autoheader: command
not found
WARNING: 'autoheader' is missing on your system.
You should only need it if you modified 'acconfig.h' or
'configure.ac' or m4 files included by 'configure.ac'.
The 'autoheader' program is part of the GNU Autoconf package:
<http://www.gnu.org/software/autoconf/>
It also requires GNU m4 and Perl in order to run:
<http://www.gnu.org/software/m4/>
<http://www.perl.org/>
Makefile:391: recipe for target 'config.h.in' failed
make[1]: *** [config.h.in] Error 127
make[1]: Leaving directory '/home/rd5/ecat_m801_linux_setup'
Makefile:416: recipe for target 'install-recursive' failed
make: *** [install-recursive] Error 1
    
```

5. 最後輸入“sudo ./modules.sh”進行 Ecat\_Utility 安裝，預設安裝於 /opt/icpdas/ecat\_m801/Ecat\_Utility 目錄底下。(ecat\_m801\_linux\_setup\_v1.0.26 以上)

## 2.4.2. 移除 Linux 驅動程式

1. 進入安裝目錄下“share/script”目錄。

```

more information, such as the ld(1) and ld.so(8) manual pages.
-----
make install-exec-hook
make[3]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
ldconfig
make[3]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
Making install in include
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/opt/icpdas/ecat_m801/include'
/usr/bin/install -c -m 644 EcatDeviceAPI.h '/opt/icpdas/ecat_m801/include'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /opt/icpdas/ecat_m801/share/sc
ript/^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /home/
bryan@icpdas-mint-ibpc:/home$ cd /opt/icpdas/ecat_m801/share/script/
    
```

2. 輸入“sudo ./remove”進行驅動程式及函式庫的移除，若無任何錯誤表示安裝成功



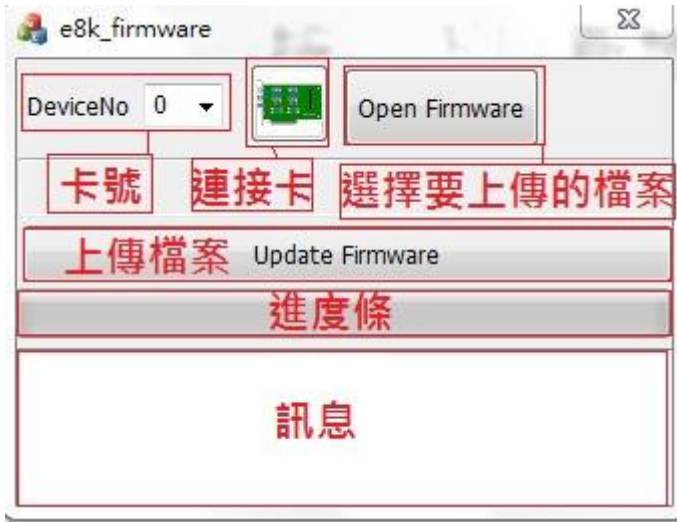
```

make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
Making install in include
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/opt/icpdas/ecat_m801/include'
/usr/bin/install -c -m 644 EcatDeviceAPI.h '/opt/icpdas/ecat_m801/include'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /opt/icpdas/ecat_m801/share/sc
ript/^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /home/
bryan@icpdas-mint-ibpc:home$ cd /opt/icpdas/ecat_m801/share/script/
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ sudo remove
[sudo] password for bryan:
sudo: remove: command not found
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ sudo ./remove
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ ls
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ ls /opt/icpdas/
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ █
    
```

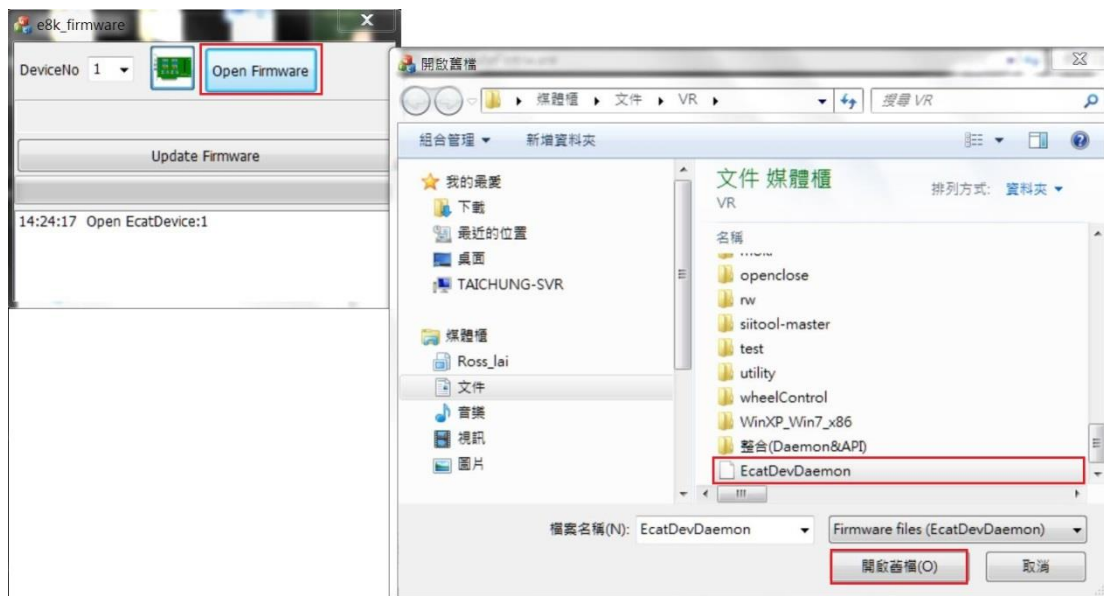
## 2.5. 更新韌體

**警告:** 若 EtherCAT MainDevice 版本為 1.0.10 或以下，請執行下列步驟更新兩次

- (1) 安裝完驅動程式後，開啟 C:\icpdas\Ecat-M801\UpdateFirmware\Update\_firmware.exe



- (2) 從  選擇裝置編號，並點選  與指定裝置建立通訊
- (3) 點選 "Open Firmware"，選擇 "EcatDevDaemon"，按下 "開啟舊檔"，若檔名不是 "EcatDevDaemon"，而是加了版本號碼，如 "EcatDevDaemon\_V1.0.17"，請先將檔名修改為 "EcatDevDaemon"。



(4) 點選”Update Firmware”，更新完成。

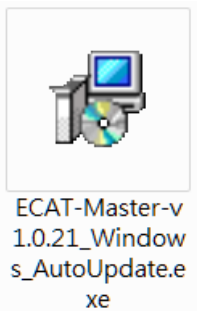
## 2.6. 自動更新軟體

### 2.6.1. Windows

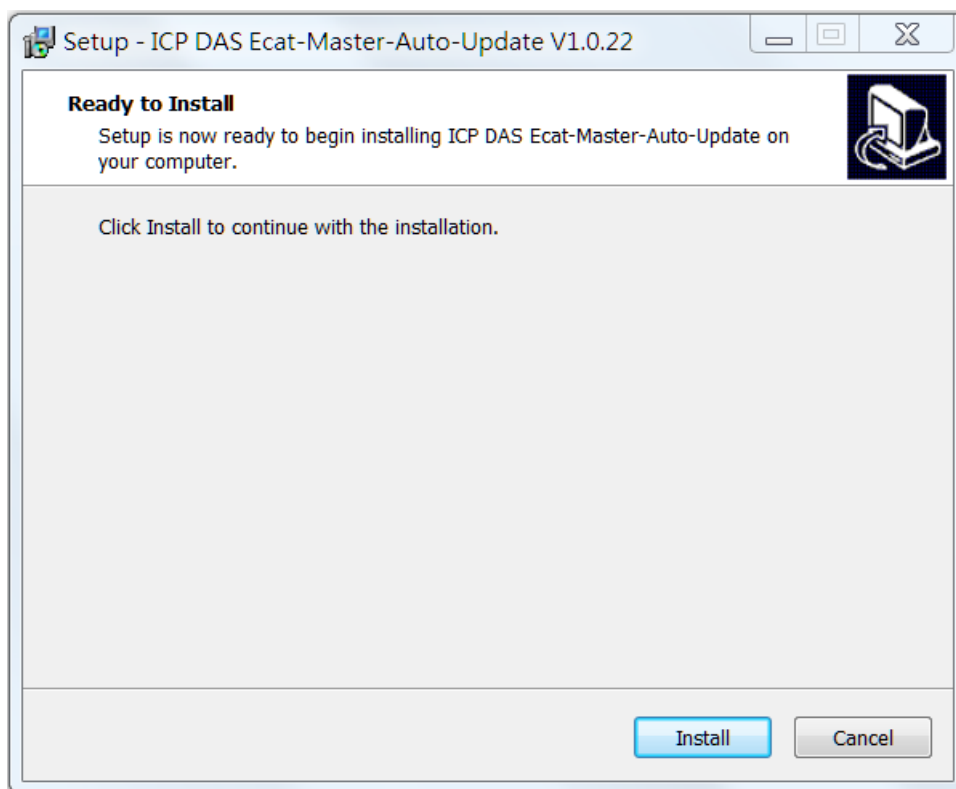
此軟體包含 API、Utility 與韌體更新

依照下列步驟來完成程式安裝：

1. 雙擊 ECAT-Master\_vx.x.xx\_Windows\_AutoUpdate.exe 驅動程式安裝執行檔



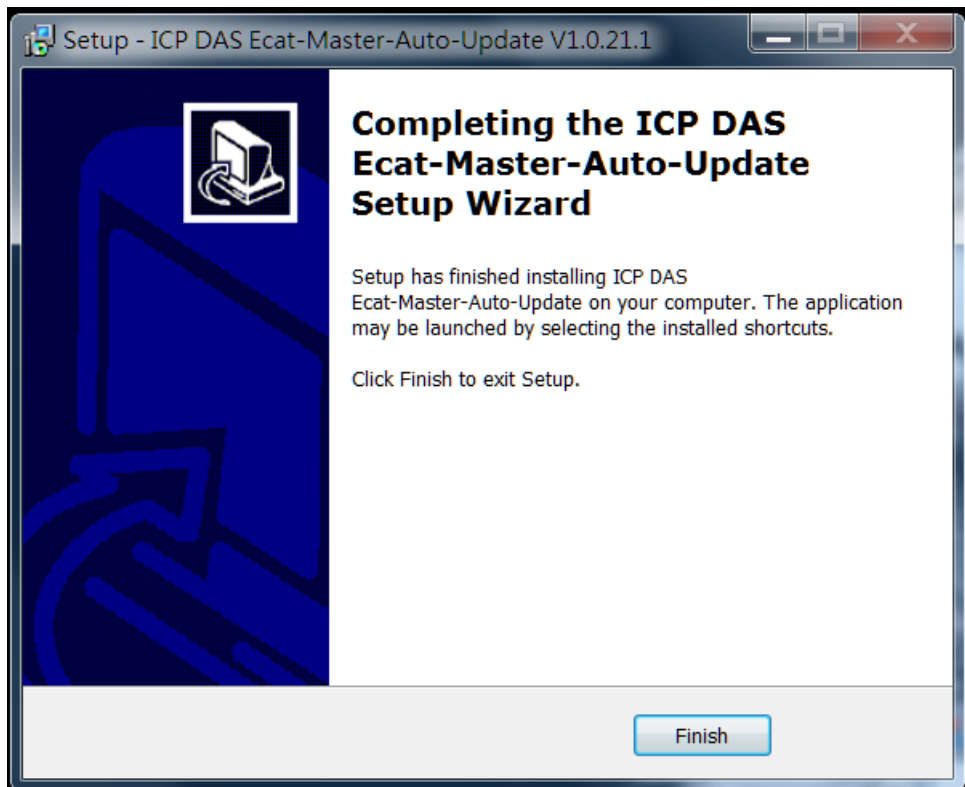
2. 預設安裝路徑 C:\icpdas\Ecat-M801。按 “Install” 按鈕開始安裝。



3. 安裝完成後會顯示當前版本資訊。



4. 按下"Finish"完成安裝

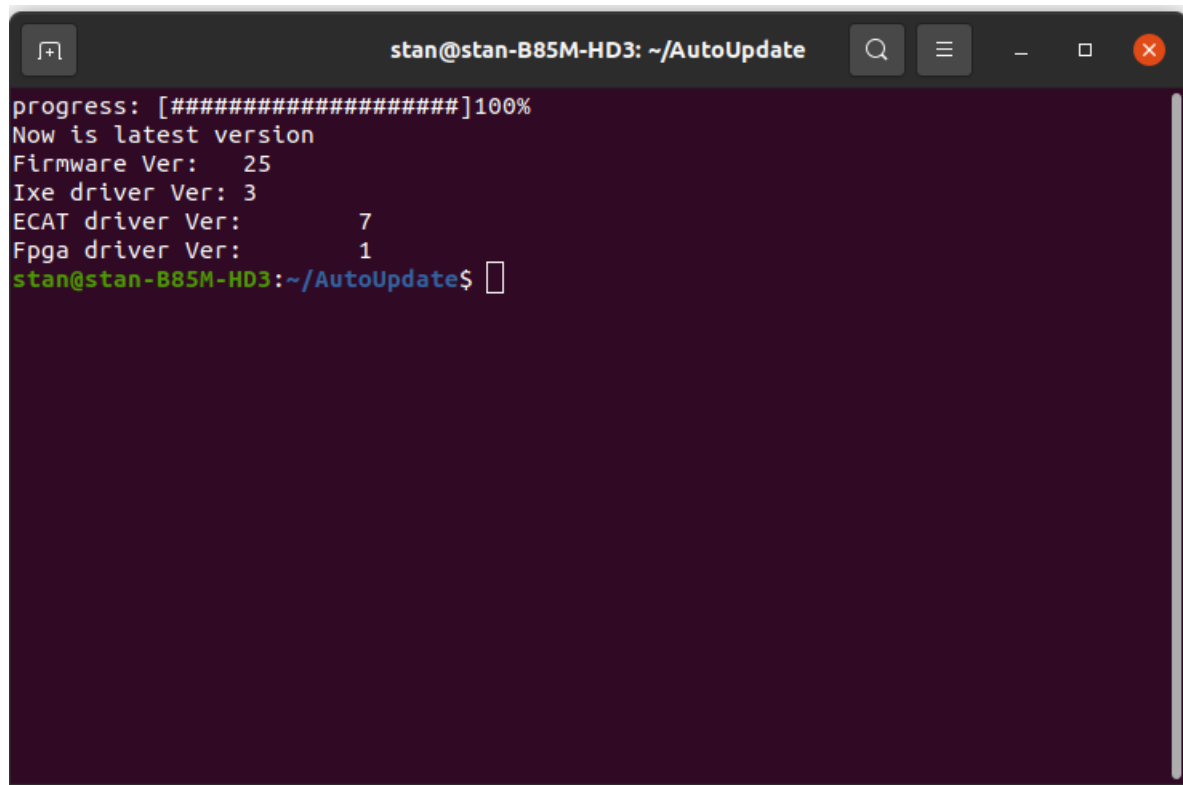


## 2.6.2. Linux

此軟體包含韌體更新

依照下列步驟來完成程式安裝:

1. 解壓縮 ECAT-Master-vx.x.xx-Linux-AutoUpdate.tar.gz
2. 輸入 `python ./main.py`
3. 完成後會顯示當前版本



```
stan@stan-B85M-HD3: ~/AutoUpdate
progress: [#####]100%
Now is latest version
Firmware Ver: 25
Ixe driver Ver: 3
ECAT driver Ver: 7
Fpga driver Ver: 1
stan@stan-B85M-HD3:~/AutoUpdate$
```

## 3. EcatUtility 操作說明

EcatUtility 為泓格所開發的工具軟體，安裝驅動程式時會被安裝至路徑

Windows: C:\icpdas\Ecat-M801\Utility\Utility.exe

Linux: /opt/icpdas/ecat\_m801/ECAT\_Utility/main (驅動程式 v1.0.26 以上)

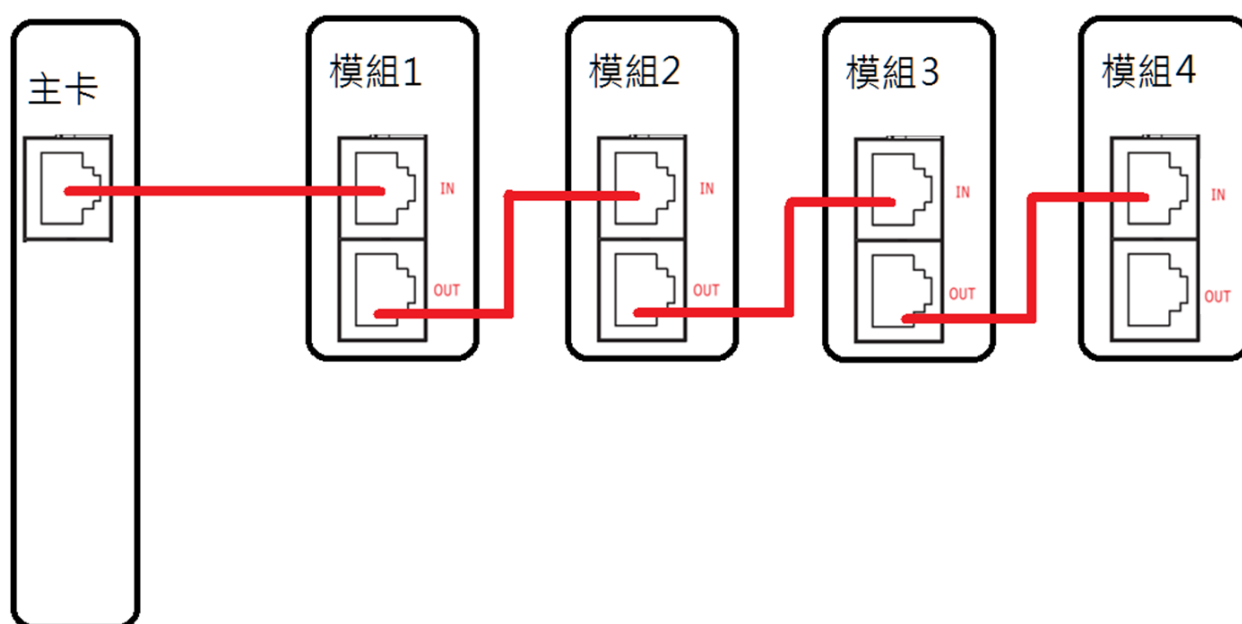
提供使用者編輯裝置網絡資訊及測試從站模組和運動控制上使用。

### 3.0. 啟動工具軟體

#### 3.0.0. 連線

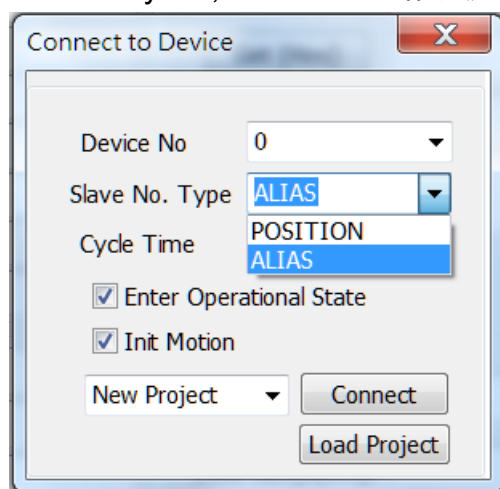
模組的網路孔有 2 個，分別為 IN 和 OUT，連線時，請注意：

主站接到第一個模組的 IN，第一個模組的 OUT 接到第二個模組的 IN，以此類推



### 3.0.1. 選擇模組編號定義

開啟 Utility 時，可以選擇從站編號的定義



項目	說明
(1) POSITION	從站編號為模組的位置 指的是模組在 EtherCAT 網路架構中的位置(MainDevice-Module 0-Module 1...)
(2) ALIAS	從站編號為模組別名(Alias) 不受模組連接順序影響，可自行設定 設定範圍: 1~65534

以圖 3.1 為例:

當從站編號類型為 POSITION 時，從站編號“1”指的是 ECAT-2028

當從站編號類型為 ALIAS 時，從站編號“1”指的是 ECAT-2011H

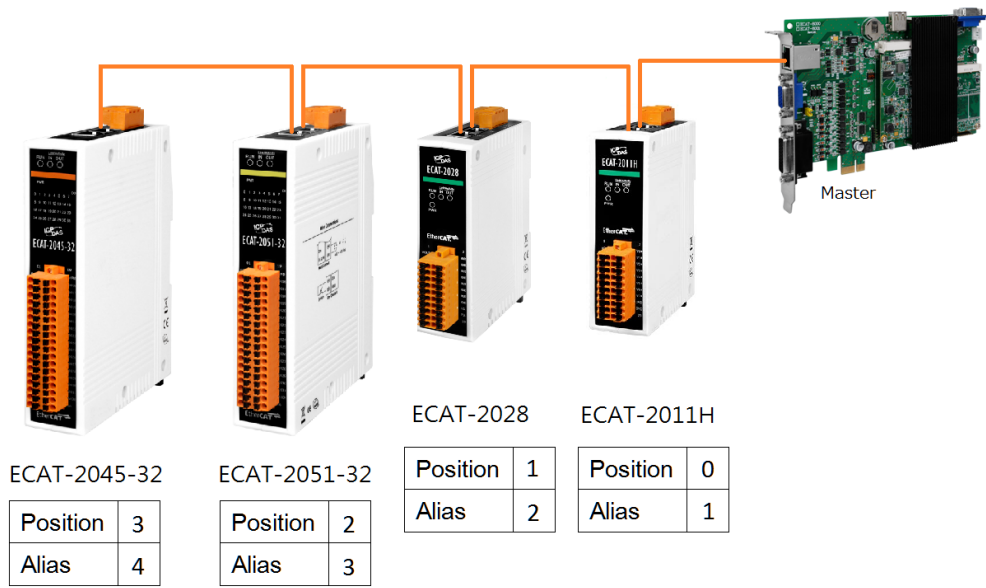
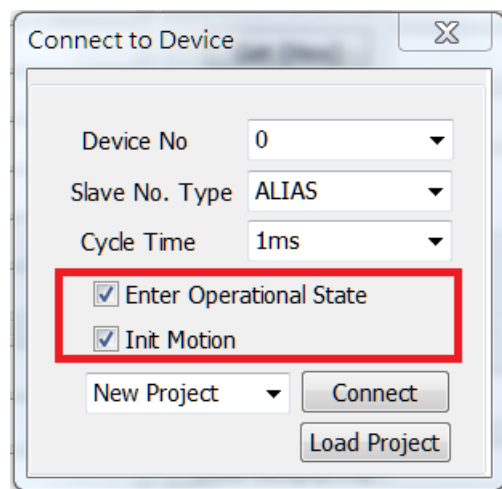


圖 3.1



### 3.0.2. 裝置初始化



當 Enter Operational State 勾選時會自動建立網路資訊(3.1.1~ 3.1.4)

當 Init Motion 勾選時會自動建立軸配置檔(3.5.1 ~ 3.5.2)

### 3.0.3. ALIAS 編輯頁面

當選擇 ALIAS 模式時，會進入 Alias 編輯頁面

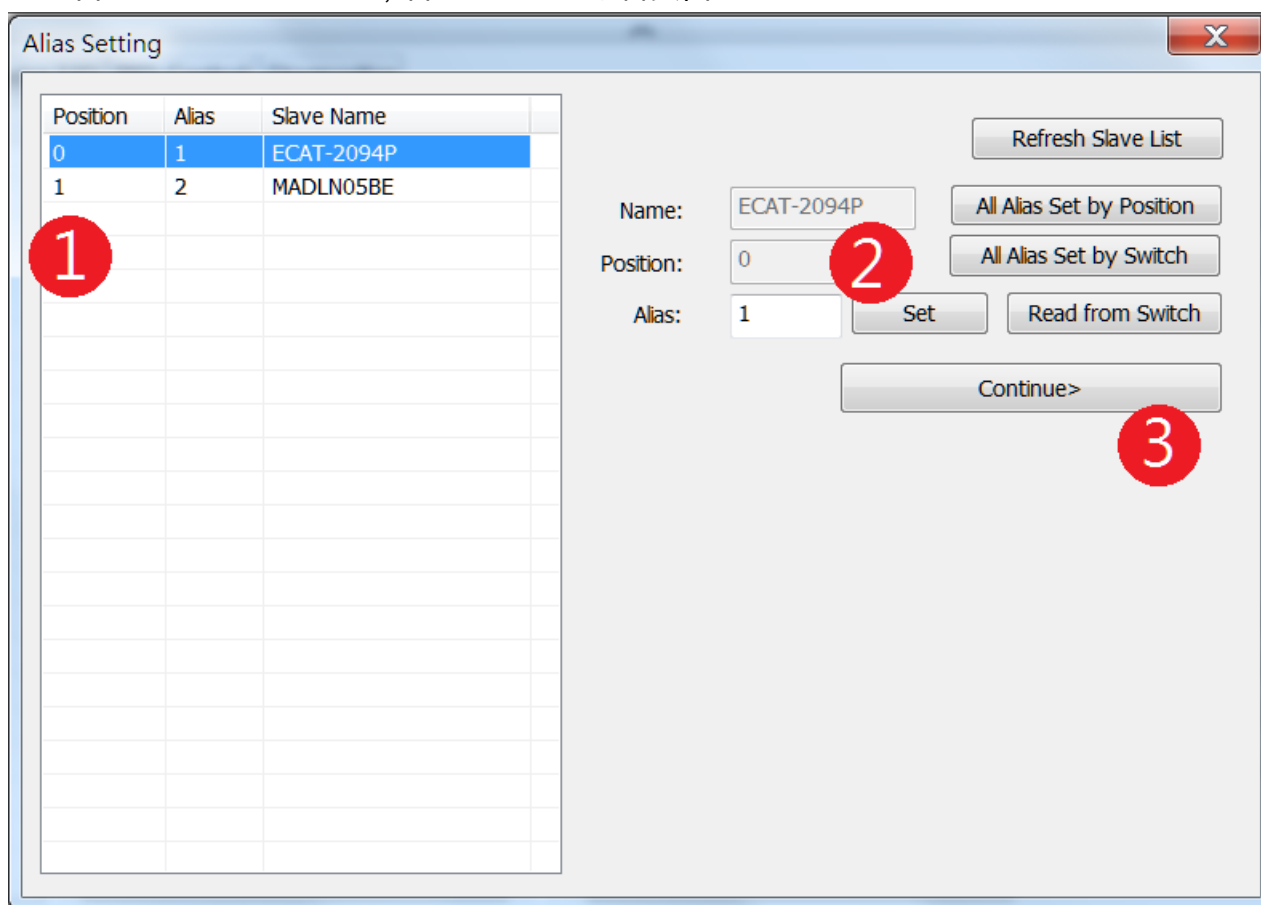
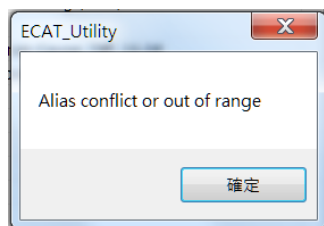


圖 3.2

表 3.1

項目	說明
(1)	從站清單
(2) 	設定 Alias
(3) 	設定完 Alias 後進入模組操作頁面
	有些模組上有別名旋鈕，此按鈕可以讀取模組旋鈕值
	將所有模組的 Alias 按照連線順序自動設定
	將所有模組的 Alias 按照別名旋鈕自動設定

當 Alias 有衝突或是不在允許範圍(1~65534)時，會出現以下錯誤



### 3.0.4. Utility 操作頁面

工具軟體包含幾個部份，如表 3.2 所示。

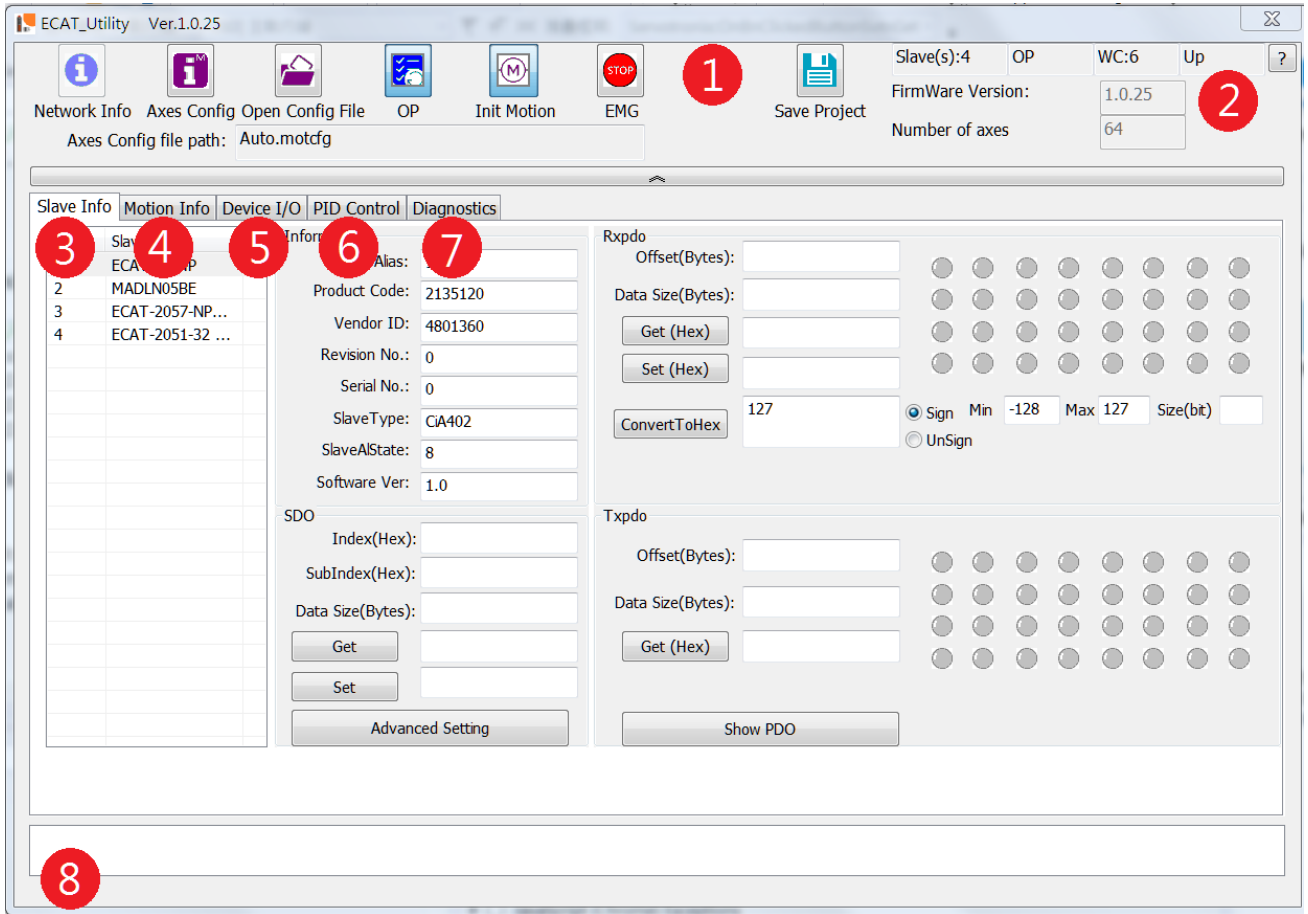


圖 3.3

表 3.2

項目	說明
(1)	工具列(裝置操作與運動控制初始化)
(2)	裝置狀態列
(3)	從站模組操作頁面
(4)	運動控制操作頁面
(5)	裝置 I/O 操作頁面
(6)	PID 操作頁面
(7)	EtherCAT 診斷操作頁面

(8)

錯誤訊息面板

### 3.1. 裝置操作工具列說明

裝置操作工具列如下圖所示，各控制項目定義如下。

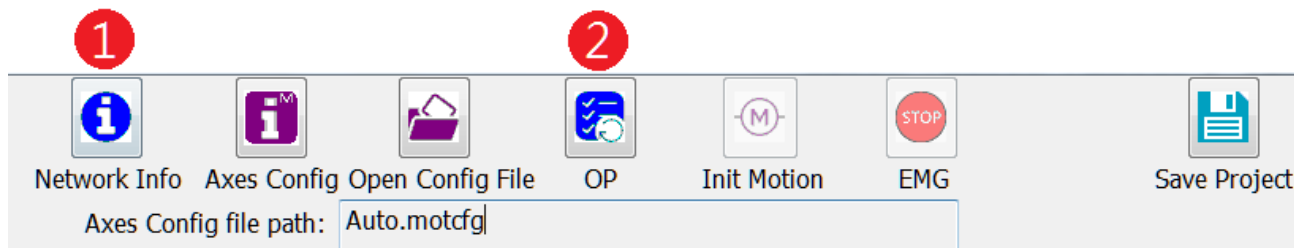


圖 3.4

表 3.3

項目	說明
(1)	進入網絡架構編輯頁面 (編輯 slaves 內容)
(2)	開始及停止指定裝置編號 EtherCAT 操作任務

### 3.1.1. 網絡架構編輯步驟

如下圖所示於裝置操作工具列上點選 **1** 將進入網絡架構編輯頁面，各控制項目定義如下。



圖 3.5



表 3.4

項目	說明
(1)	掃描目前網絡上全部的從站資訊
(2)	根據目前網絡從站資訊，從指定 ESI 檔目錄取得詳細資料來建立裝置所需的網絡架構資料

<p>(3) </p>	<p>將本機端建構完成的網絡架構資料傳至裝置端，編號由(5)指定</p>
<p>(4) </p>	<p>從裝置端傳輸網絡架構資料至本機端，編號由(5)指定</p>
<p>(5) <input type="text" value="0"/></p>	<p>網絡架構編號清單，供上傳或下載使用</p>
<p>(6)</p>	<p>網絡架構資訊面板</p>
<p>(7) <input type="text" value="DC"/> <input type="text" value="Enable"/></p>	<p>從站的當前 DC 預設值(依照從站通訊特性來決定，有些模組可設定 DC，有些則不可設定 DC)</p>
<p>(8) Domain: <input type="text" value="0"/></p>	<p>設定從站屬於的 Domain</p>
<p>(9)</p> <p>Domain Type: <input type="text" value="Individually"/></p>	<p>設定從站的 Domain 模式</p>
<p>(10)</p> <p>CiA402 Pdo Mapping <input type="text" value="Mode:0"/></p>	<p>進行 CiA402 模組之 PDO 配置</p>
<p>(11)</p> <p><input type="checkbox"/> multi-axis</p> <p>number of axes: <input type="text"/></p> <p>Pdo Increment(hex): <input type="text"/></p> <p>Pdo Entry Increment(hex): <input type="text"/></p>	<p>進行支援多軸的 CiA402 模組之設定</p> <p>參數說明:</p> <ul style="list-style-type: none"> <li>➤ number of axes: 該模組支援的軸數</li> <li>➤ Pdo Increment: 每一軸的 Pdo 的位移量，第零軸 Pdo index 為 1A00 若第一軸 Pdo index 為 1A10，則設定 10 若第一軸 Pdo index 為 1A20，則設定 20</li> <li>➤ Pdo Entry Increment: 每一軸的 Pdo Entry 的位移量，以 Controlword 來說明 第零軸 Controlword 的 Index 為 6040h 若第一軸 Controlword 的 Index 為 6840h，則設定 800 若第一軸 Controlword 的 Index 為 7040h，則設定 1000</li> </ul>
<p>(12) <input type="button" value="Pdo Assign"/></p>	<p>進行 PDO 配置</p>
<p>(13)</p> <p><input type="button" value="Save"/></p> <p><input type="button" value="Load"/></p>	<p>編輯完網絡架構之後，可以將當前的網路架構儲存為一個檔案，需要在另一張卡片上使用同一網路架構時，就可以直接讀取先前儲存的檔案，讀取檔案的同時，請確</p>



認是否需要設定模組的 Alias。

1. 點選  可掃描已連接從站資訊(此步驟非必要一定要執行，目的是了解目前從站資訊)。
2. 點選  程式會開始讀取目錄(C:\icpdas\Ecat-M801\ESI)內相關的 ESI 檔來建制裝置所需網絡架構資料，並將相關資訊更新至網絡架構資訊面板 (6)。

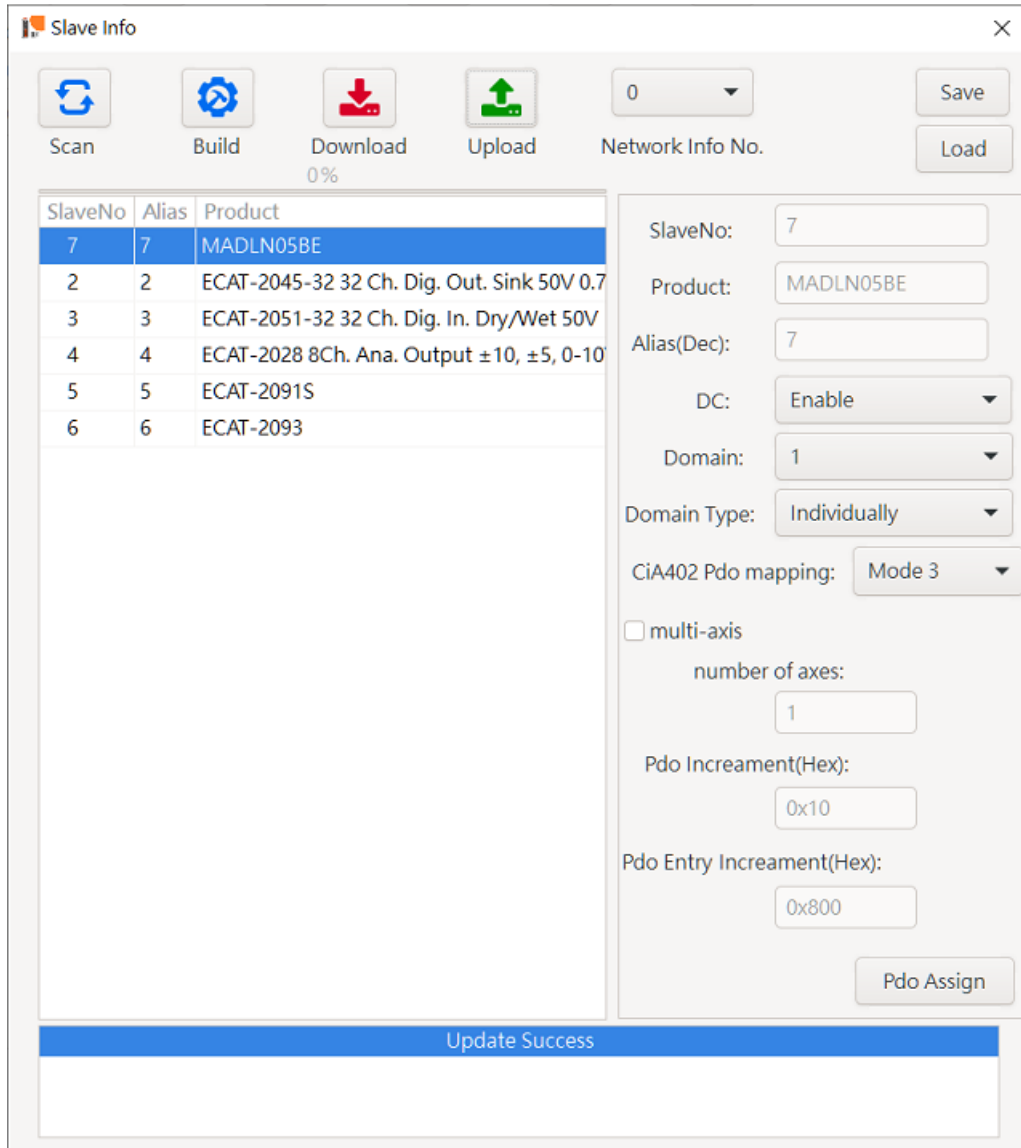


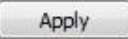

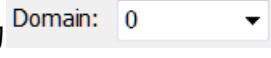

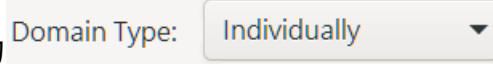
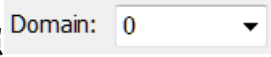



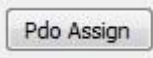



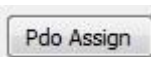


圖 3.6

3. 點選  可設定該從站的  性質，有些模組無法設定，點選  確認更改提交的設定值。
4. 點選  可設定該從站的 ，設定模組在不同的 Domain 時，當有一 Domain 出現通訊錯誤(working counter error)，其餘 Domain 不會受到影響。
5. 點選  可設定所有從站的 ，當 Domain Type 為 Classic 時從站會依照  進行配置；當 Domain Type 為 Individually 時，每個從站的 Domain 會獨立配置，也就是說，當有一模組出現通訊錯誤(working counter error)，其餘模組皆不會受到影響。  
 注：在 EnumCycleTime 為 1ms 時，最多支援 50 個獨立 Domain  
 在 EnumCycleTime 為 0.5ms 時，最多支援 25 個獨立 Domain
6. 確認網絡架構資料內容後，從網絡架構清單中選擇所要指定傳送的網絡架構編號，並點選  進行傳送。此資訊會傳入控制卡內。
7. 使用者也可點選  來將裝置內部網絡架構傳輸至本機端，以顯示其架構資訊。
8. 步驟 2、3 會掃描 ESI 並配置 PDO(使用預設值)。如果需要進階配置 PDO，點選欲變更的模組  後，再點選  進行配置。
9. 編輯完網絡架構之後，可以點選  將當前的網路架構儲存為一個檔案，需要在另一張卡片上使用同一網路架構時，就可以使用  讀取先前儲存的檔案，讀取檔案的同時，請確認是否需要設定模組的 Alias。

### 3.1.2. 網絡架構編輯(進階配置 PDO)

對象(Object) 被映射為過程數據對象(PDO)後，每個 EtherCAT 通訊週期都會進行資料交換，以上的操作會使用模組的預設 PDO，如果只需使用預設的 PDO(如一般 I/O)，可以跳過此章節。

點選欲配置的模組  後，再點選  將進入 Pdo 編輯頁面，各控制項目定義如下。

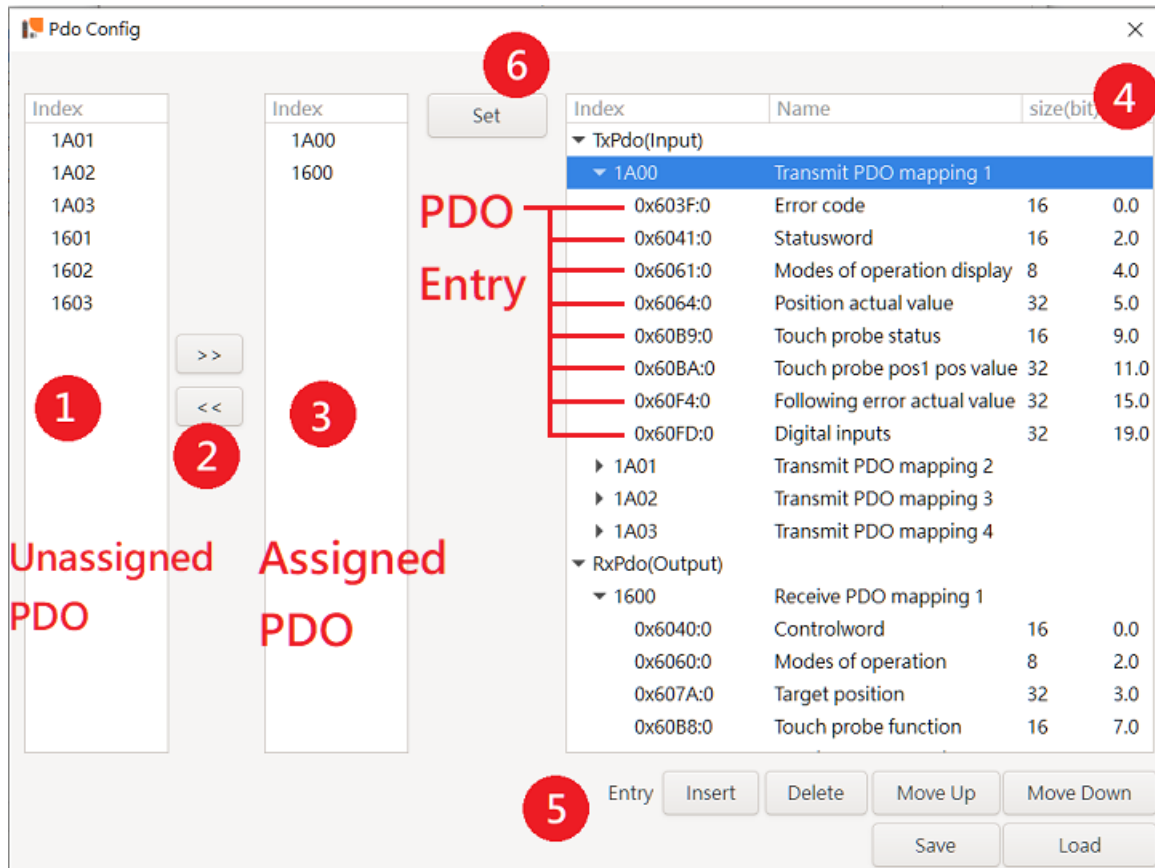
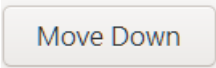
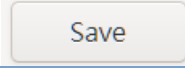
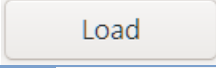
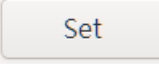

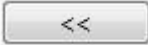

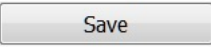
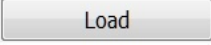


圖 3.7

表 3.5

項目	說明
(1)	不要分配的 Pdo
(2)	分配按鈕
	分配 Pdo
	取消分配 Pdo
(3)	要分配的 Pdo
(4)	Pdo 清單 分為 TxPdo(Input)與 RxPdo(Output) ,
(5)	功能按鈕
	插入一個 PDO Entry
	刪除選取的 PDO Entry
	將選取的 PDO Entry 上移

	將選取的 PDO Entry 下移
	將 Pdo 清單儲存
	讀入 Pdo 清單
(6) 	設定按鈕

1. 在"Unassigned Pdo" 與"Assigned Pdo"區域中選擇欲配置之 PDO，並點選  或  來進行配置，配置到"Assigned Pdo"區域的 PDO 會依照順序被配置到該模組，點選 Set 按鈕即可完成該模組的設定，完成設定後需點選  進行傳送。
2. 在點選 Set 按鈕之前，可以編輯 PDO 的內容，在 PDO List 區域，點選要編輯的 PDO 或是 PDO Entry，然後點選"功能按鈕"，就可以對點選的 PDO / PDO Entry 進行操作。
3. 修改完 PDO List 後，可以點選  將編輯過後的 PDO List 存檔，下次有需要再次修改 PDO List 時，可以點選  讀取編輯過後的 PDO List，並修改該 PDO List，或是供其他相同的模組使用。

注意事項

1. 如果 PDO 後面有顯示(Mandatory) 或是(Fixed)，則無法編輯該 PDO，例: 1A01(Fixed)。
2. 若配置了多個 PDO，請確認各個 PDO 中的 PDO Entry 沒有重複，否則可能會無法進入 Operation Mode 或是工作不正常，以下圖為例，若在"Assigned Pdo"中配置了 1600 與 1601，可以發現 1600 中的 6040:0 與 1601 中的 6040:0 重複，可能造成錯誤。

1600	1st Receive PDO Ma...
+ Exclude	
0x6040:0	Controlword
0x607A:0	Target position
0x6060:0	Modes of operation
1601	2nd Receive PDO M...
+ Exclude	
0x6040:0	Controlword

3. 在新增 PDO 或是 PDO Entry 時，請務必確認該模組支援使用者所增加的 PDO / POD Entry
4. 在新增 PDO Entry 時，請確認該 PDO Entry 屬於 TxPDO(輸入) 或是 RxPDO(輸出)

### 3.1.3. 網絡架構編輯(進階配置 PDO for CiA402 驅動器)


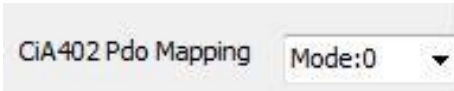
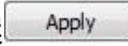
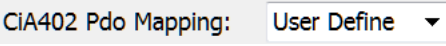
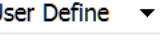

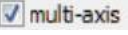
1. 針對 CiA402 模組提供了 4 組 PDO 配置，預設值為 mode 0，各 mode 定義請看表 3.7
2. 如果需要配置 CiA402 模組之 PDO，點選欲變更的模組  後，再點選  進行配置，點選  確認更改提交的設定值。
3. 當該模組配置到不支援的 PDO Entries 時，可能造成該模組無法進入 Operation Mode 或是工作不正常，請務必確認該模組是否支援所選擇的 PDO Mapping mode 中的所有 PDO Entries，若該模組不支援當前選擇的 PDO Mapping Mode，請更改為其他的 PDO Mapping Mode，或是更改為  並按照”網絡架構編輯(進階配置 PDO)”進行進階配置。
4. 當更改為  時，必須含有表 3.6 的 PDO Entries，否則會無法進入 Operation Mode。

表 3.6

RxPdo Entries		TxPdo Entries	
6040	Controlword	6041	Statusword
6060	Modes of operation	6061	Modes of operation display

5. 若模組為支援多軸的 CiA402 馬達驅動器，可進行以下設定

點選 ，勾選 ，輸入 Pdo Increment、Pdo Entry Increment 及 number of axes

參數說明：

- number of axes: 該模組支援的軸數
- Pdo Increment: 每一軸的 Pdo 的位移量，  
第零軸 Pdo index 為 1A00  
若第一軸 Pdo index 為 1A10，則設定 10  
若第一軸 Pdo index 為 1A20，則設定 20
- Pdo Entry Increment: 每一軸的 Pdo Entry 的位移量，以 Controlword 來說明  
第零軸 Controlword 的 Index 為 6040h  
若第一軸 Controlword 的 Index 為 6840h，則設定 800  
若第一軸 Controlword 的 Index 為 7040h，則設定 1000





表 3.7

	RxPdo Entries		TxPdo Entries	
Mode0	6040	Controlword	6041	Statusword
	6060	Modes of operation	603F	Error code
	607A	Target Position	6061	Modes of operation display
	60FF	Target Velocity	6064	Position actual value
	6071	Target Torque	606C	Velocity actual value
	60B8	Touch probe function	60FD	Digital inputs
	60B0	Position offset	6077	Torque actual value
	60B1	Velocity offset		
	60B2	Torque offset		
Mode1	6040	Controlword	6041	Statusword
	6060	Modes of operation	603F	Error code
	607A	Target Position	6061	Modes of operation display
	60FF	Target Velocity	6064	Position actual value
	6071	Target Torque	606C	Velocity actual value
	60B8	Touch probe function	60FD	Digital inputs
	60B0	Position offset	6077	Torque actual value
	60B1	Velocity offset		
	60B2	Torque offset		
Mode2	6040	Controlword	6041	Statusword
	6060	Modes of operation	603F	Error code
	607A	Target Position	6061	Modes of operation display
	60FF	Target Velocity	6064	Position actual value
	6071	Target Torque	606C	Velocity actual value
	60B8	Touch probe function	60FD	Digital inputs
	60B0	Position offset	6077	Torque actual value
	60B1	Velocity offset		
	60B2	Torque offset		
Mode3	6040	Controlword	6041	Statusword

	6060	Modes of operation	603F	Error code
	607A	Target Position	6061	Modes of operation display
	60FF	Target Velocity	6064	Position actual value
	6074	Target Torque	606C	Velocity actual value
	60B8	Touch probe function	60FD	Digital inputs
	60B0	Position offset	6077	Torque actual value
	60B4	Velocity offset		
	60B2	Torque offset		

### 3.1.4. 開始及停止 EtherCAT 操作任務步驟

1. 當使用者完成網絡架構編輯步驟後，於裝置操作工具列中，選擇編輯完成之網絡架構編號。
2. 於從站通訊週期清單中選擇 EtherCAT 操作任務通訊週期時間。
3. 點選  開始 EtherCAT 操作任務，若無任何錯誤訊息出現，待裝置網絡狀態列中 EtherCAT 狀態轉變為 OP 時，即可開始相關 EtherCAT 操作。
4. 如果再次點選 ，則會停止 EtherCAT 操作任務。

## 3.2. 錯誤訊息面板

如下圖所示，當使用者操作各控制項時發生錯誤，該訊息欄會產生發生錯誤時的時間、錯誤訊息和錯誤代碼，以供使用者進行除錯。另外於訊息欄中點選滑鼠右鍵，右鍵選單中有清除錯誤訊息的功能清除全部的錯誤訊息

```
09:08:54 User timeout
09:08:54 Failed to open EcatDevice:-1304
```

圖 3.8

## 3.3. 裝置狀態列

如下圖所示，當裝置通訊操作開啟後，裝置網絡狀態會即時更新，以供使用者了解目前網絡狀態，其各控制項目定義如下。



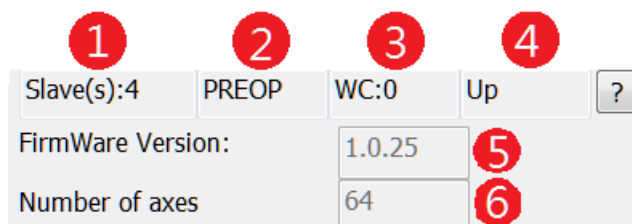


圖 3.9

表 3.8

項目	說明
(1)	目前已連接的從站數量
(2)	目前整個網絡 EtherCAT 狀態，其狀態如下。 (INIT, PREOP, SAFEOP, OP, Unknown)
(3)	EtherCAT 工作計數器數值 (Working Counter)，正常情況下在進入 OP 後會不為 0
(4)	顯示 EtherCAT 網路孔的連接狀態 Down: 主站無連接任何從站 Up: 主站已連接任何從站
(5)	裝置韌體版本
(6)	裝置支援的最大軸數

### 3.4. 從站模組操作界面說明

從站模組操作頁面如下圖所示，其包含表中幾個部份。

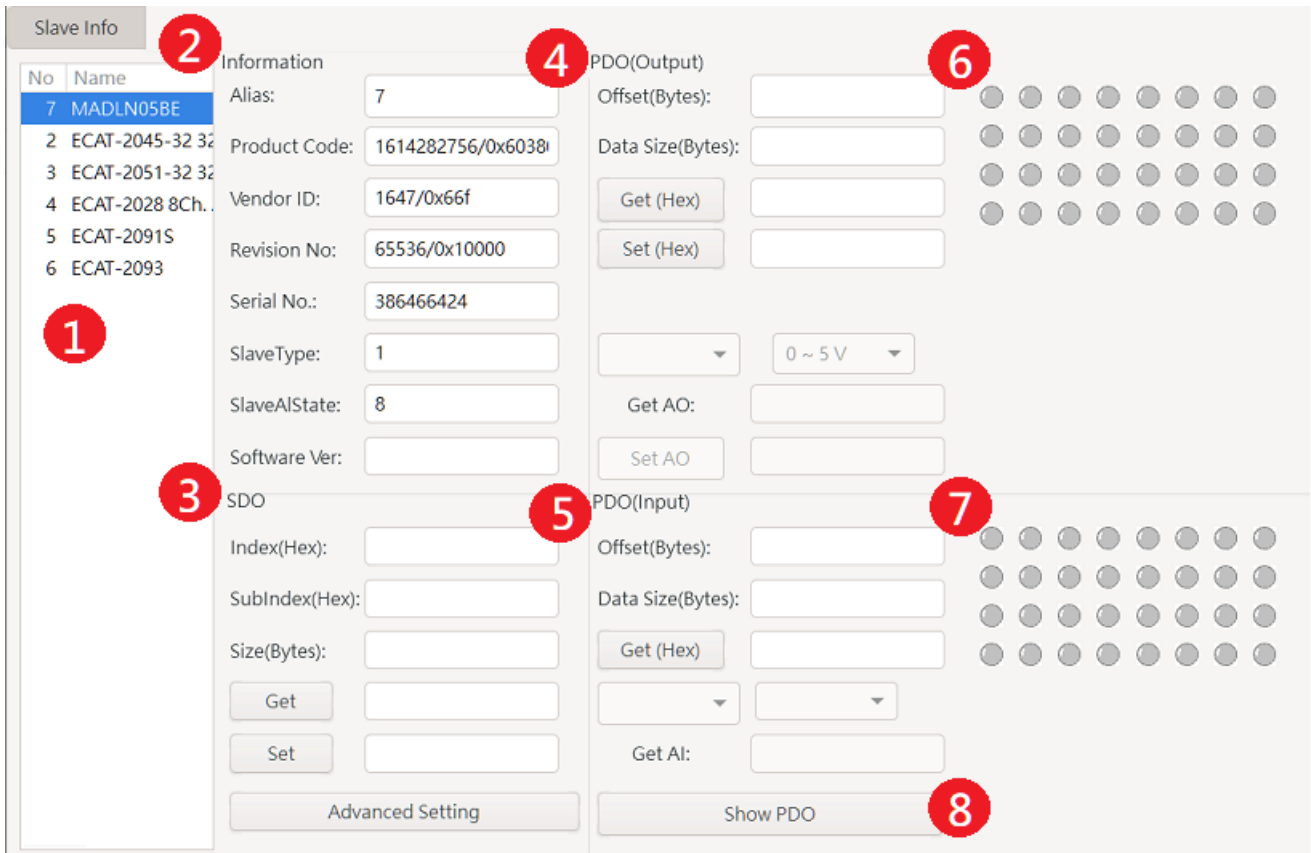


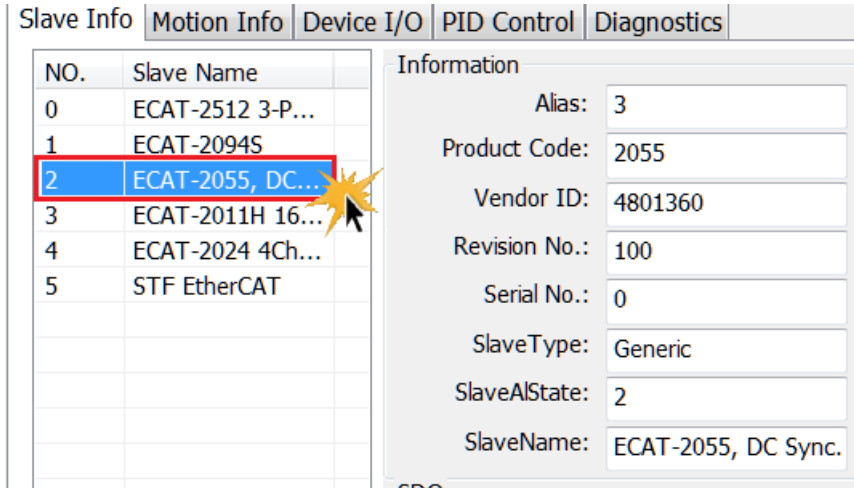
圖 3.10

表 3.9

項目	說明
(1)	從站清單
(2)	從站資訊顯示的群組框
(3)	SDO 讀寫操作的群組框
(4)	PDO(Output)讀寫操作的群組框
(5)	PDO(Input)讀取操作的群組框
(6)	DO LED 操作與狀態顯示的群組框
(7)	DI LED 狀態顯示的群組框
(8)	PDO 顯示框

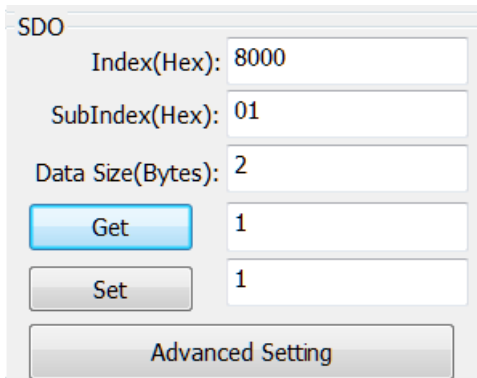
### 3.4.1. 從站模組基本操作步驟

1. 當使用者完成裝置初始化步驟後，使用者於從站清單中點選任意從站模組後，相關的從站資訊會同時顯示於從站資訊群組框內。



### 3.4.2. 從站模組 SDO 操作步驟

1. 當使用者欲進行 SDO 物件的讀寫操作時，請於 SDO 讀寫操作群組框內輸入 Index, SubIndex, Data Size 等資料，並點選"Get/Set"來讀寫從站模組 Object 資料。



2. 當使用者完成"網絡架構編輯步驟"的"點選開啟選擇 ESI 檔目錄對話框"步驟且完成讀取 ESI 檔後，可以開始使用 Advanced Setting 功能。
3. Advanced Setting 按鈕提供使用者方便讀寫 SDO。開啟頁面後，會自動讀取 SDO 並顯示至"Current Value"欄。按下右鍵可對選定的 SDO 進行修改。

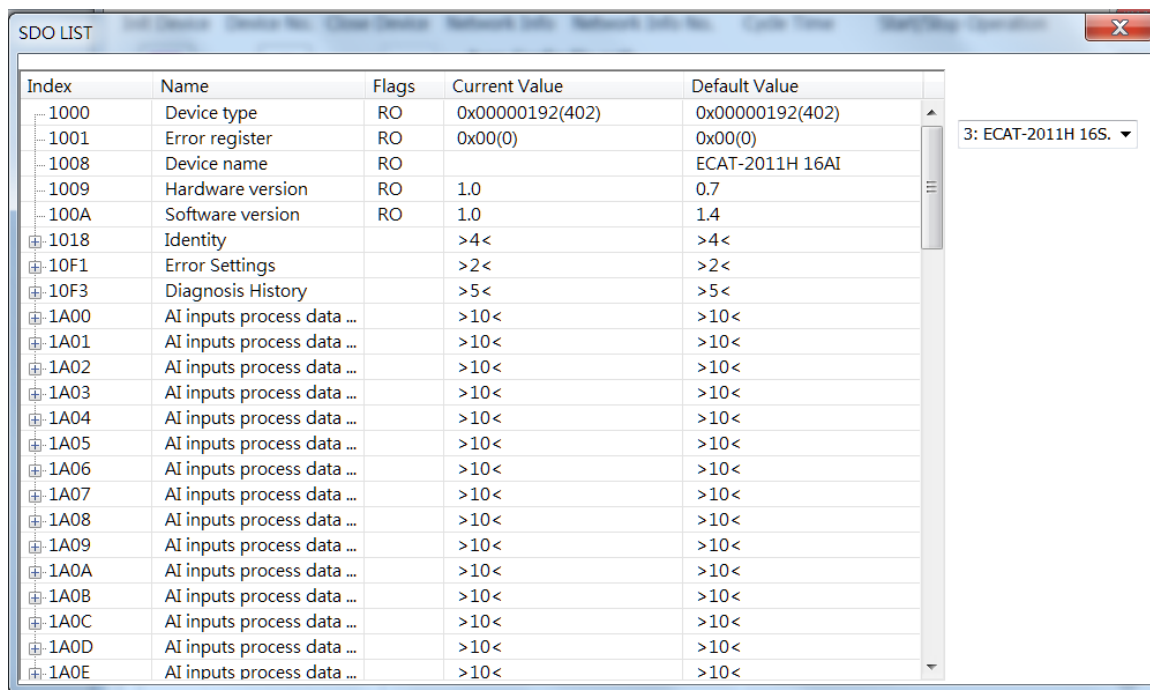


圖 3.11

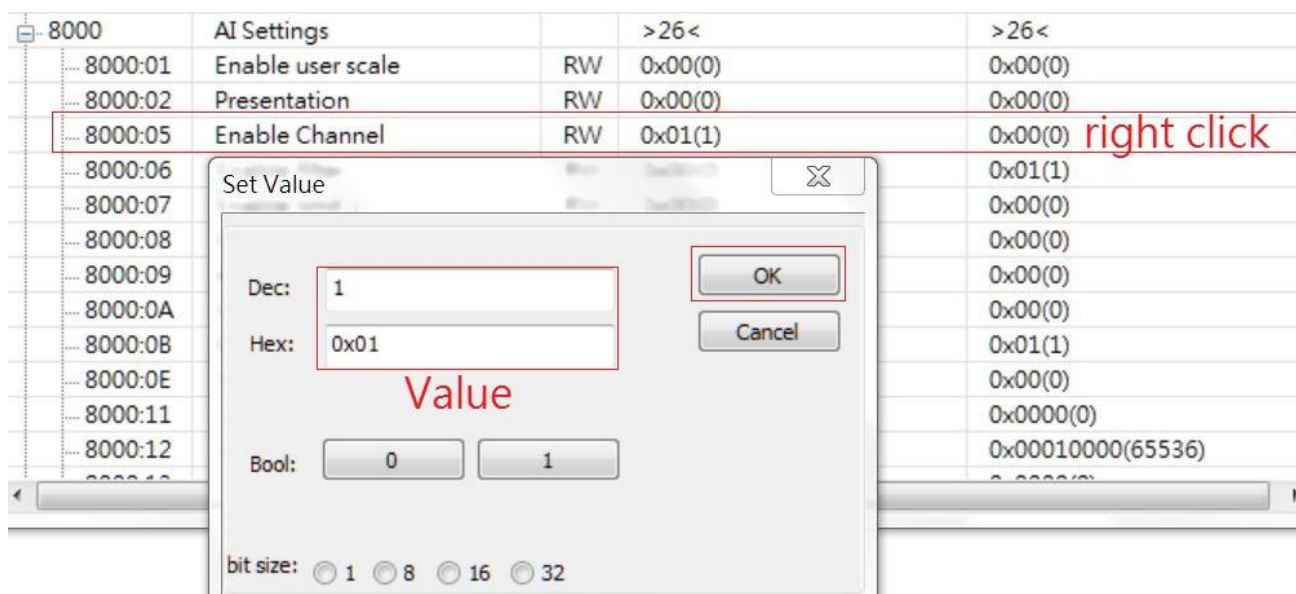


圖 3.12

4. 當點選的模組為 ECAT-2091S/ ECAT-2094S 時，會多出以下項目(表 3.10)


表 3.10

項目	說明
(1) 	將當前從站的 SDO 數據存檔 (Index 範圍 0x8000~0x8321)

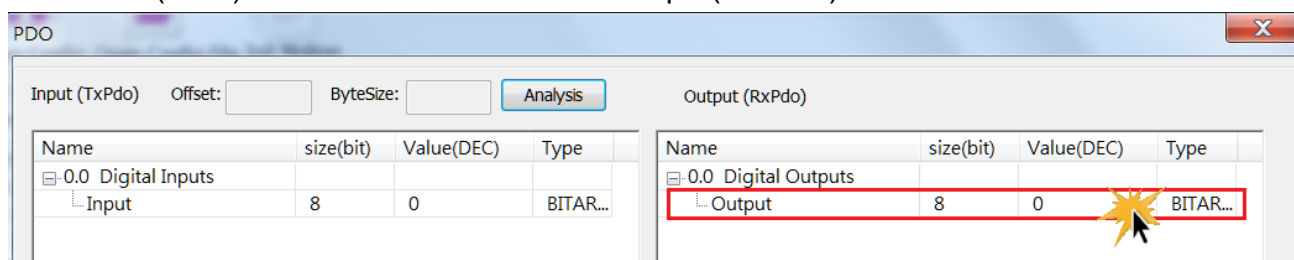
<p>(2) <input type="button" value="Load from File"/></p>	<p>讀取 SDO 數據檔並寫入當前模組，並確認是否燒錄 EEPROM</p>
<p>(3) <input type="button" value="Load File for All"/></p>	<p>讀取 SDO 數據檔並寫入所有與主站連線的匹配模組，並確認是否燒錄 EEPROM</p>

### 3.4.3. 從站模組 PDO 及 DI/DO LED 操作步驟

The screenshot shows a software interface for configuring PDO (Process Data Object) for a slave module. It is divided into two main sections: PDO(Output) and PDO(Input). Each section contains input fields for 'Offset(Bytes)' and 'Data Size(Bytes)', buttons for 'Get (Hex)' and 'Set (Hex)', and a 'Show PDO' button at the bottom. To the right of the input fields are visual indicators consisting of a grid of circles, likely representing the state of individual data points or LEDs. The PDO(Output) section also includes a dropdown menu and a '0 ~ 5 V' label, along with 'Get AO' and 'Set AO' buttons.

1. 當使用者完成 EtherCAT 操作任務步驟後，即可開始 PDO 的讀寫操作。請於 PDO 讀寫操作群組框內輸入 Offset, Data Size 等資料，這裡的 Offset 指的是相對於從站模組 Process Image 資料開始位置偏移量，其單位為 Byte，輸入完畢後點選”Get/Set”來即可讀寫從站模組 PDO 資料。寫入資料時每個 Byte 的資料使用逗點進行分隔，例如欲寫入 2 Bytes 資料大小時，資料寫入文字框內請輸入(FF,02)即表示第 1 Byte 資料寫入 0xFF，第 2 Byte 資料寫入 0x02。
2. DO/DI LED 操作基本上是 RxPDO 與 TxPDO 操作的進階處理，將數值變成燈號顯示。目前此功能只支援簡單的 DI/DO 從站模組，若該從站模組有支援此功能時將顯示 DI/DO 操作群組框，使用者可利用 DI/DO LED 進行寫入或監看。
3. 當使用者完成”網絡架構編輯步驟”的”點選  開啟選擇 ESI 檔目錄對話框”步驟且完成讀取 ESI 檔後，可以開始使用 Show PDO 功能。
4. Show PDO 按鈕提供使用者方便讀寫 PDO。開啟頁面後，會自動讀取 PDO 並顯示至”

Value(DEC)”欄。按下右鍵可對選定的 Output(RxPDO)進行修改。



### 3.4.4. 從站模組 PDO 監看與分析

1. Show PDO 按鈕功能: 用於監看 RxPdo 以及 TxPdo。

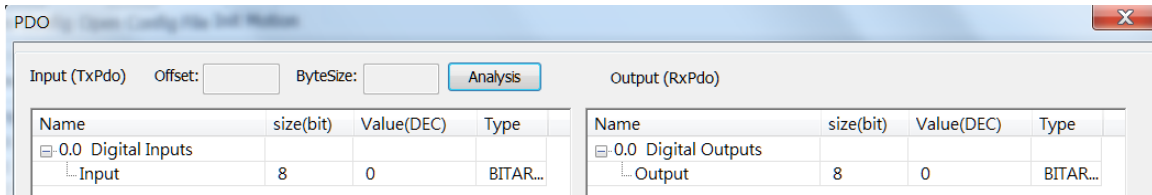


圖 3.13

2. PDO 分析(EtherCAT MainDevice Firmware Ver 1.0.15 或以上)

左鍵點擊欲分析的資料，上方顯示該資料的 Offset 以及 ByteSize，若未顯示，代表該資料無法進行分析，點擊 Analysis 開始進行分析

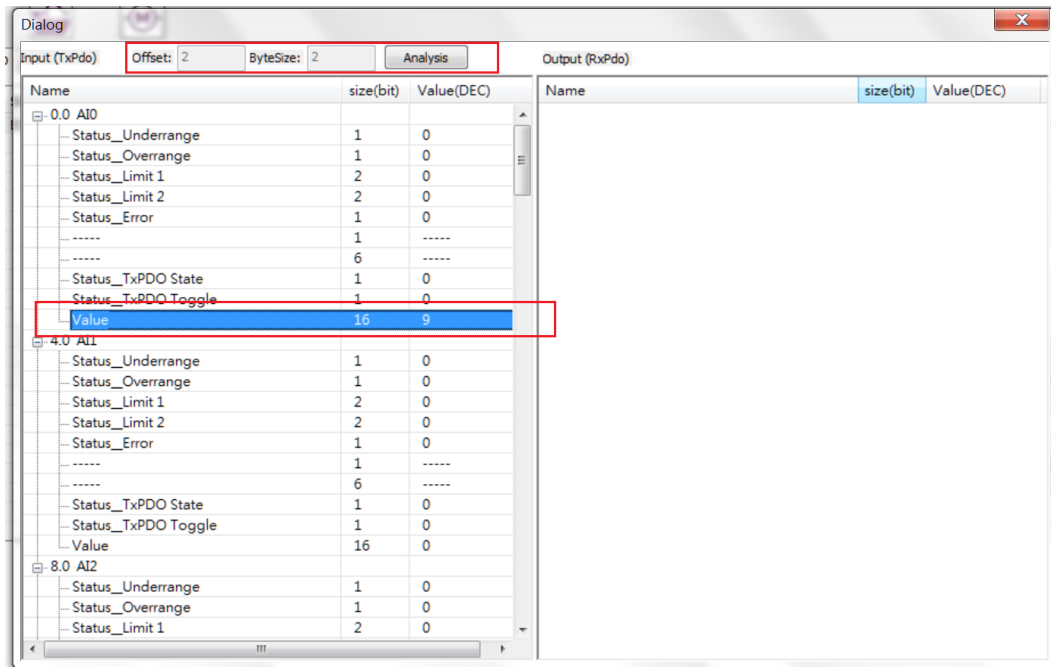


圖 3.14

### 3. PDO 分析、設定濾波器

圖 3.15 為 PDO 分析得結果，上方顯示頻域，下方顯示時域

- 可用來分析該資料是否有特定的雜訊頻率以及雜訊強度大小
- 可設定軟體濾波器以減少雜訊干擾
- 軟體濾波器使用以下 API
  - `int32_t ECAT_SetAiFilterFreq(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t FilterType, double Frequency)`
  - `int32_t ECAT_SetAiFilterParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint16_t Enable)`
- Notch filter(點阻濾波器):可減少特定頻率的雜訊。如:60hz 雜訊。
- Low Pass filter(低通濾波器):可減少高頻雜訊。
- High Pass filter(高通濾波器):可減少低頻雜訊。

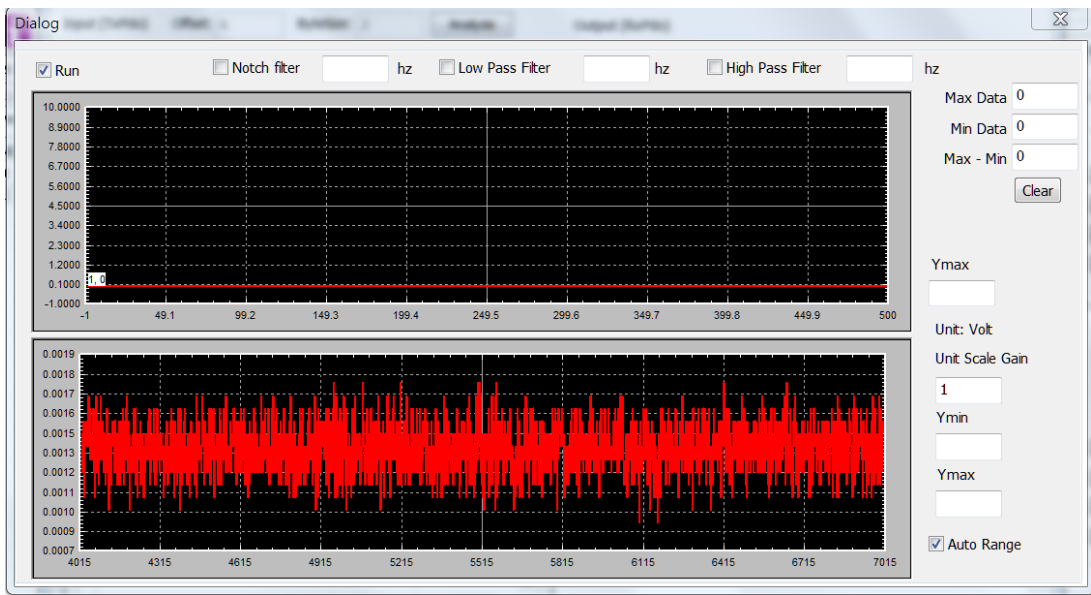
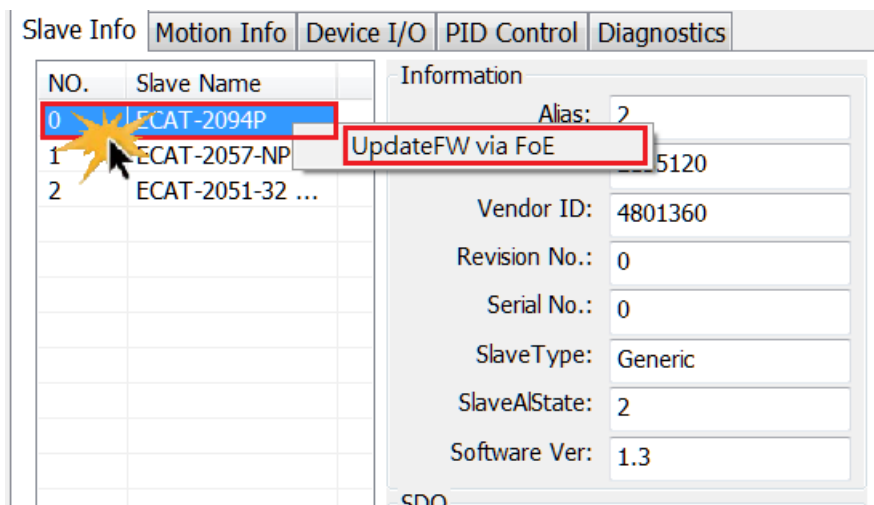


圖 3.15



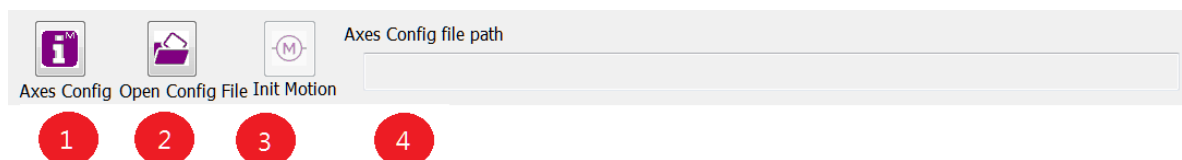
### 3.4.5. 從站模組韌體更新(FoE)






使用者於從站清單中對從站模組下右鍵後，點選”UpdateFW via FoE”，然後選擇要更新的檔案即可


### 3.5. 運動控制初始化操作工具列說明

在開始運動控制操作之前需針對各軸號進行初始化動作，此動作可使用如下圖所示運動控制初始化操作列進行，各控制項目定義如下。



項目	說明
(1) 	開啟運動控制初始化參數檔編輯頁面
(2) 	指定運動控制初始化參數檔路徑
(3) 	使用(2)指定的參數檔來做運動控制初始化
(4)	目前運動控制初始化參數檔路徑

### 3.5.1. 運動控制初始化參數檔編輯步驟

如下圖所示於運動控制初始化操作工具列上點選  將進入運動控制初始化參數檔編輯頁面，各控制項定義如下。

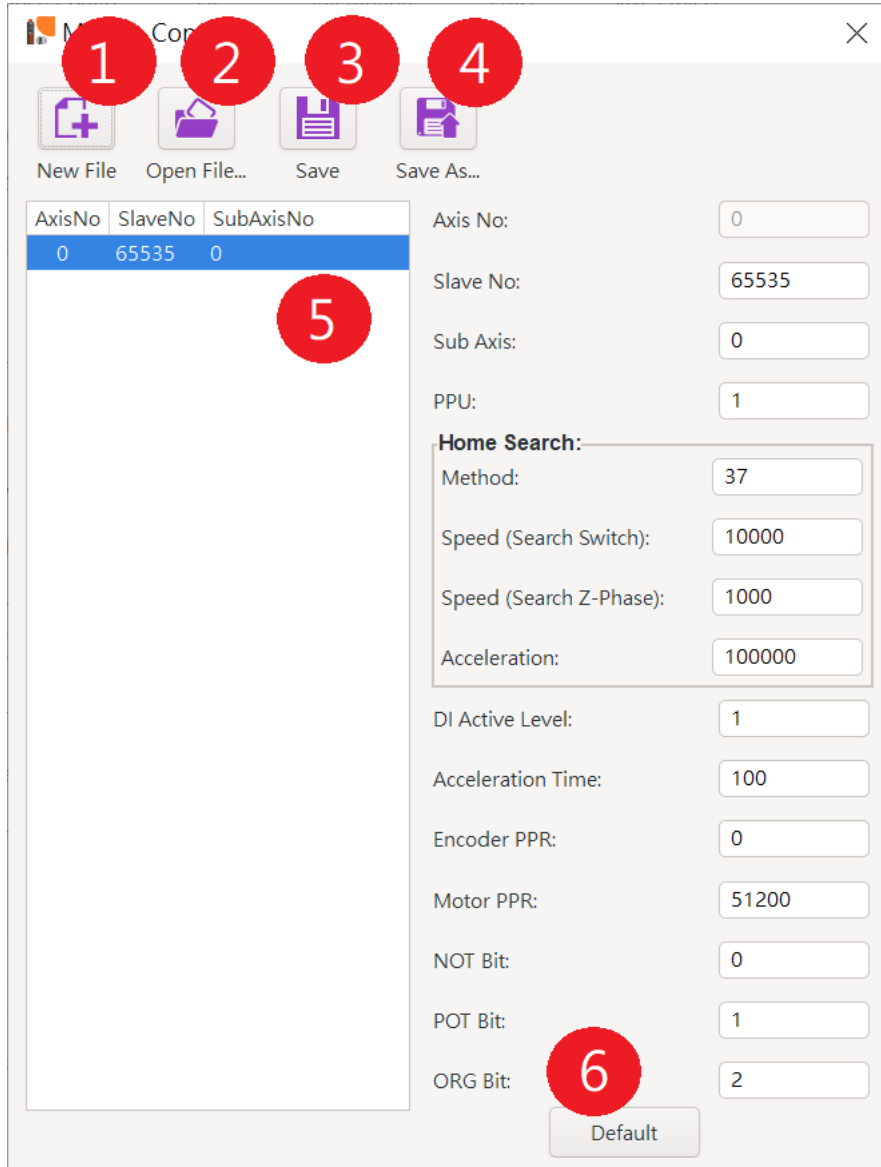







圖 3.16

表 3.11

項目	說明
(1) 	建立新的初始化參數檔
(2) 	開啟已建立的初始化參數檔
(3) 	儲存目前開啟的初始化參數檔

(4) 	將目前開啟的初始化參數檔另存新檔
(5)	軸控初始化參數資訊欄
(6) Default	將設定值設定為預設值

1. 點選  建立全新的初始化參數檔，預設會先建立 1 軸參數資料。

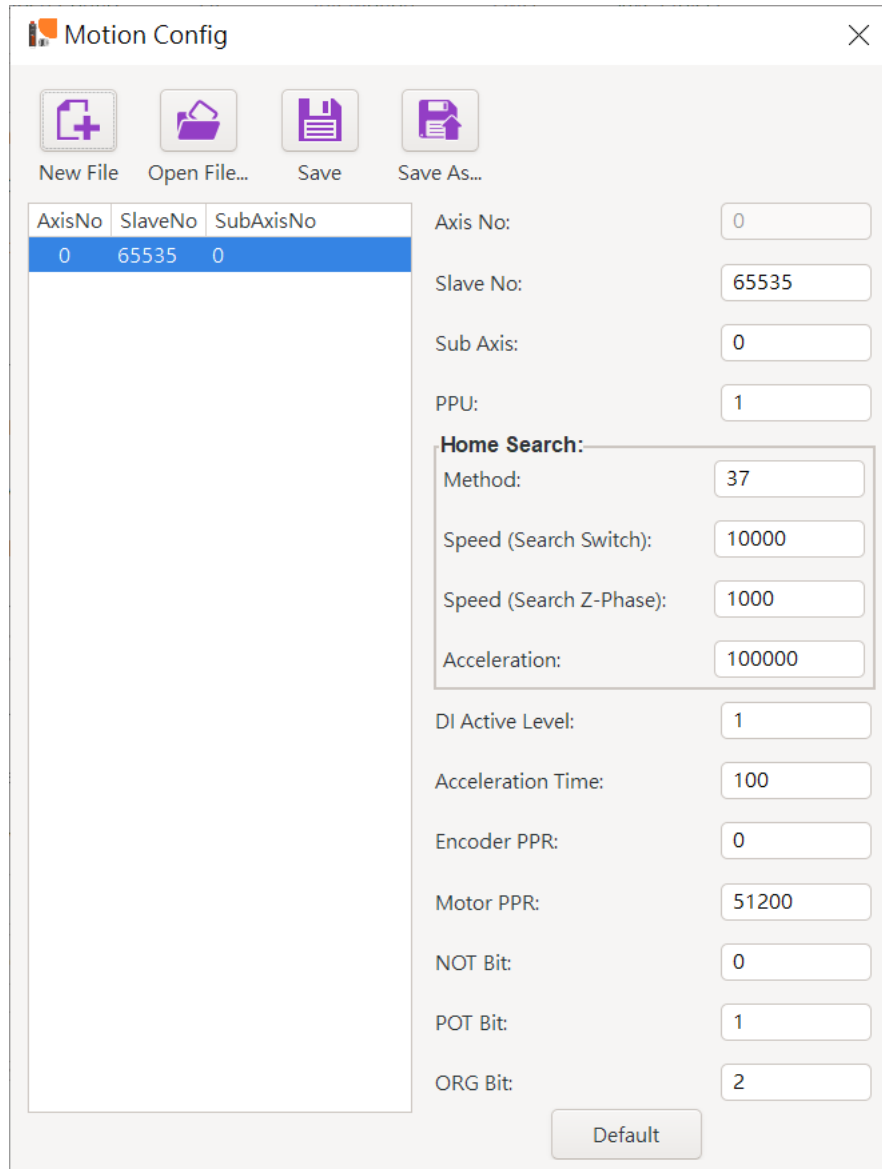


圖 3.17

- 設定 SlaveNo 該參數為軸號對應的從站編號，軸號為 Axis 清單順序，若需設為虛擬軸，請將 SlaveNo 設為 65535
- SubAxis 是給支援多軸的馬達控制模組使用(EtherCAT MainDevice FirmWare Version 需在 1.0.15 或以上，否則只支援 ECAT-2094S)，設定 0 為選定該模組第零軸，設定 1 為選定該模組第壹軸，以此類推。
- 設定 PPU (Pulses Per Unit，每一單位為幾個 pulses)，若要將單位設為圈，而每轉動 1 圈需要 4194304 個 pulse，那就將 PPU 設為 4194304。需要注意的是，表 3.12 內的所有 PDO Entry 的單位都需要是 pulse 才可以自定義 PPU，而當驅動器的 PDO

Entry(表 3.12)單位不為 pulse 時，需要將 PPU 設為 1。

表 3.12

RxPDO		TxPDO	
607A	Target Position	6064	Position actual value
60FF	Target Velocity	606C	Velocity actual value

- Home Method (找 Home 的方式)
  - Home Speed(Search Switch) (尋找 Switch 的速度)
  - Home Speed(Search Z-Phase) (尋找 Z 相訊號的速度)
  - Home Acceleration (找 Home 的加速度)
  - DI Active Level: 極限與原點開關的檢出準位，通常都是”High”，DI(Object 0x60FD)的極限開關為 ON 狀態時檢出，此時馬達無法運轉，但是 MR-JET-G-N1 剛好相反，DI(Object 0x60FD)的極限開關訊號為 OFF 時馬達無法運轉，也就是說，檢出準位為”Low”，因此需要將此參數設為”Low”。若驅動器極限與開關的檢出準位為”Low”，則需要確認 EtherCAT MainDevice 版本為 1.0.18 或以上才可使用。
  - Acceleration Time (單軸運動時使用的加速度時間)
  - EncoderPPR (Encoder 每一轉為幾個 pulses，符合 CiA402 規範的模組不需設定)，支援 ECAT-2091S/ECAT-2094S，有安裝 Encoder 時才需要設定。
  - MotorPPR (馬達每一轉為幾個 pulses，符合 CiA402 規範的模組不需設定)等參數，支援 ECAT-2091S/ECAT-2094S，有安裝 Encoder 時才需要設定。
  - NOT Bit 負極限訊號的來源，訊號的來源通常是(Object 0x60FD)的第 0 個 Bit
  - POT Bit 正極限訊號的來源，訊號的來源通常是(Object 0x60FD)的第 1 個 Bit
  - ORG Bit 原點訊號的來源，訊號的來源通常是(Object 0x60FD)的第 2 個 Bit
2. 於任何軸的節點上按下滑鼠右鍵以顯示右鍵選單，有三項可以操作。點選加入軸”Add Axis”會於最後的軸節點之後再增加一個軸參數；點選插入軸”Insert Axis”則會於當前節點之後再增加一個軸參數；點選刪除軸”Delete Axis”則將刪除所點選節點的軸參數。

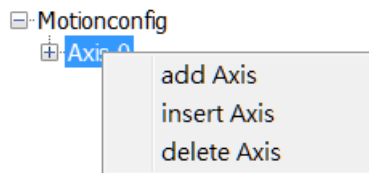




圖 3.18

3. 完成編輯後，點選可進行儲存動作，若之前從未儲存過，會開啟另存新檔對話框來指定

參數檔儲存路徑。

4. 或點選進行另存新檔的動作。

例:

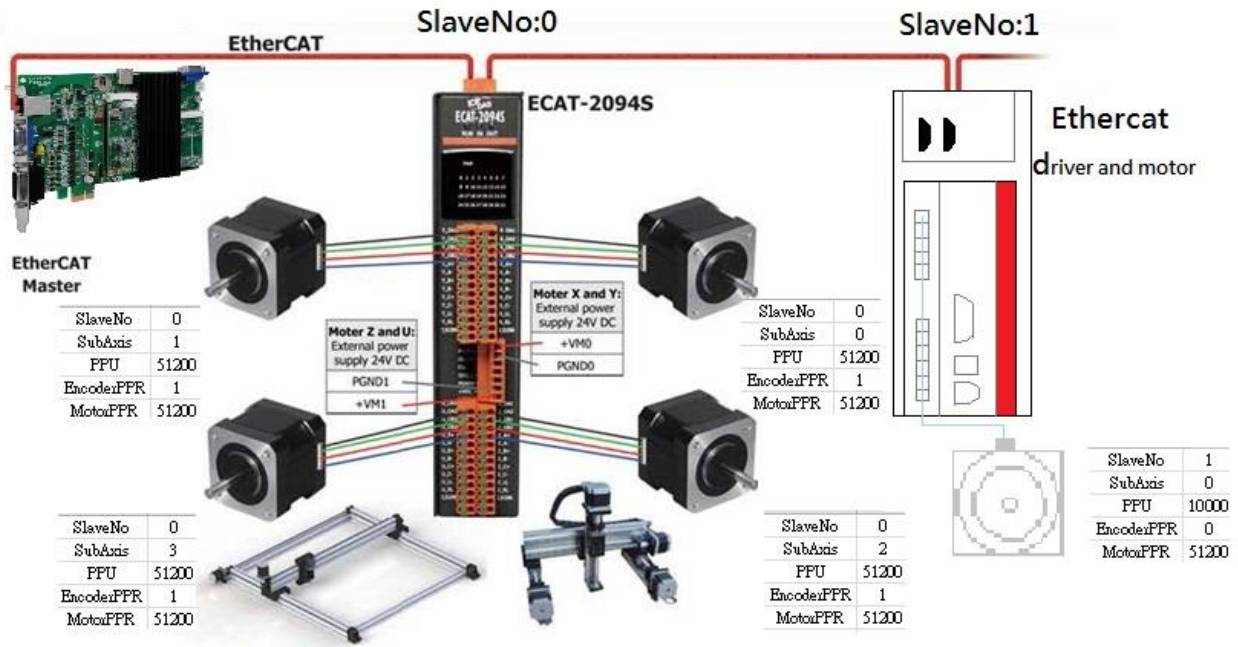




圖 3.19

### 3.5.2. 運動控制初始化步驟

1. 當使用者完成運動控制初始化參數檔編輯步驟後，於運動控制初始化操作工具列上點選開啟已輯編完成的初始化參數檔。
2. 點選時，將使用該初始化參數檔進行初始化的相關動作。
3. 若無任何錯誤產生即表示初始化成功。

### 3.6. 運動控制操作介面說明

完成運動控制初始化步驟後，使用者即可開始運動控制的相關操作，運動控制操作介面可分成兩部份：(1)單軸運動控制操作頁面。(2)群組運動控制操作頁面。

#### 3.6.1. 單軸運動控制操作頁面說明

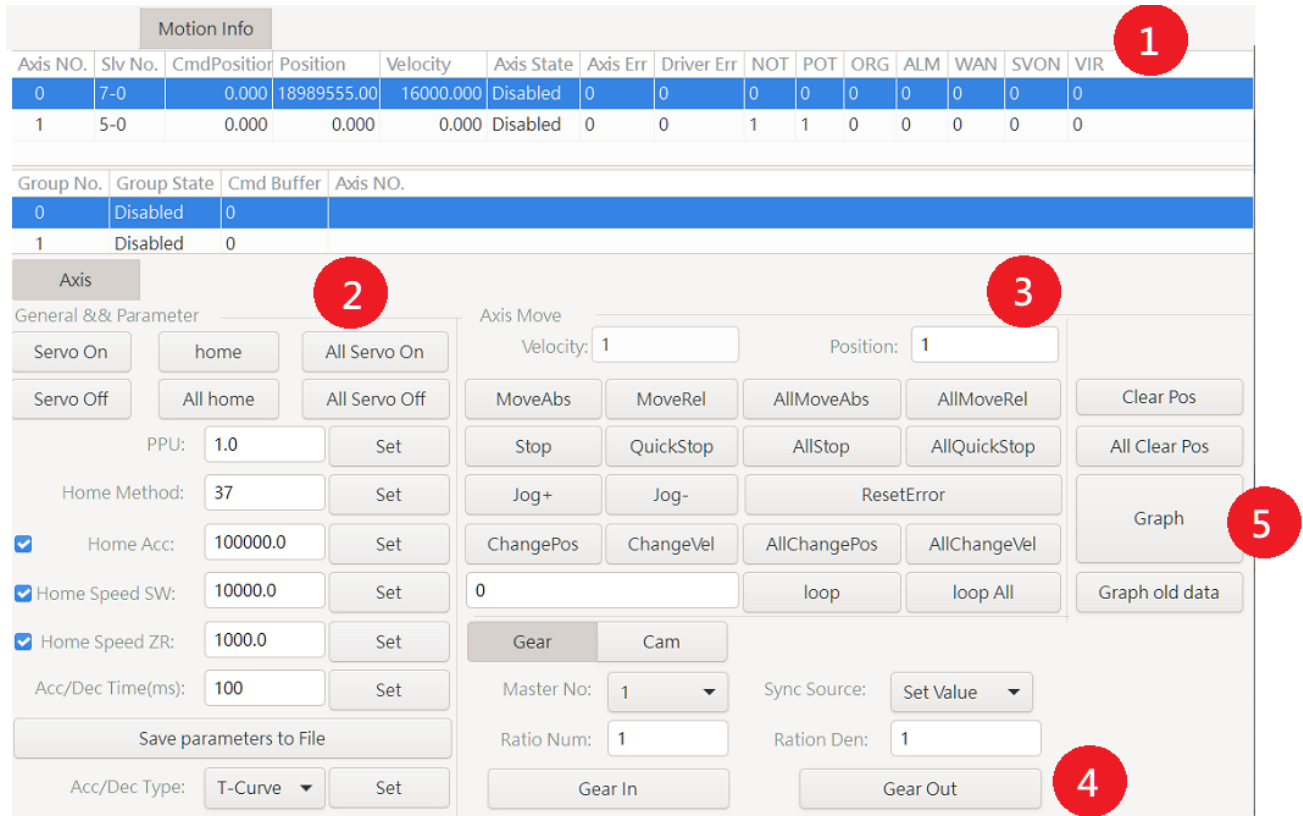


圖 3.20

表 3.13

項目	說明
(1)	單軸運動資訊欄
(2)	單軸通用性操作及參數設定框 (點選(1)裡的任一軸，該軸資訊會顯示在這裡，也可以修改設定值)
(3)	單軸運動控制框 (可以做簡易功能測試)
(4)	單軸電子齒輪/電子凸輪(可以做簡易功能測試)
(5)	單軸位置與速度簡易圖形顯示



## 單軸運動資訊欄內容定義

表 3.14

項目	說明
Axis No.	軸號
Position	當前軸位置
Velocity	當前軸速度
AxisState	軸狀態
Axis Error	軸最後出錯的代碼
Drive Error	驅動器出錯的代碼
NOT	負極限開關的狀態
POT	正極限開關的狀態
ORG	原點開關的狀態
ALM	驅動器錯誤的狀態
WAN	驅動器警告的狀態
SVN	Servo ON/OFF 狀態
VIR	是否為虛擬軸

## 單軸通用性操作及參數設定

1. 在單軸運動運制資訊面板中選擇要進行操作的軸號。
2. 點選驅動器伺服致能”Servo ON”或驅動器伺服除能”Servo OFF”按鈕進行致能/除能動作。
3. 點選原點復歸按鈕”Home”來進行單軸原點自動復歸運動。
4. 使用者可根據相關參數點選設定按鈕”Set”進行參數的設定動作，但此動作不會改變原始設定檔的資訊。

## 單軸運動控制

1. 使用者可先點選單軸運動資訊欄內的軸號來指定欲操作之單軸。
2. 設定速度”Velocity”、位置”Position”等參數。
3. 點選絕對運動”MoveAbs”或相對運動” MoveRel”按鈕進行單軸運動控制。
4. 單軸運動控制中若想停止，可點選停止按鈕”Stop”或快速停止”QuickStop”來停止單軸運動控制。

5. 若想對全部軸進行運動控制操作時，設定速度"Velocity"、位置"Position"參數，接著點選全部單軸絕對運動"All MoveAbs"或全部單軸相對運動"All MoveRel"按鈕即可使用同一參數下達全部軸數進行一樣的單軸運動控制。此動作並非補間運動，只是使用同一參數針對全部軸數各別下達命令。
6. 亦可點選全部單軸停止"All Stop"或全部單軸快速停止"All QuickStop"按鈕來停止全部軸數運動控制。
7. 按下正方向寸動"Jog+"或負方向寸動"Jog-"按鈕時，會利用單軸運動控制框內速度"Velocity"進行等速度運動控制，放開按鈕時停止等速度運動控制。
8. 先設定數值，點選 loop 或 loop All 時，會從該軸目前的位置項與設定位置之間來回運動。

---

### 單軸電子齒輪跟隨運動控制

---

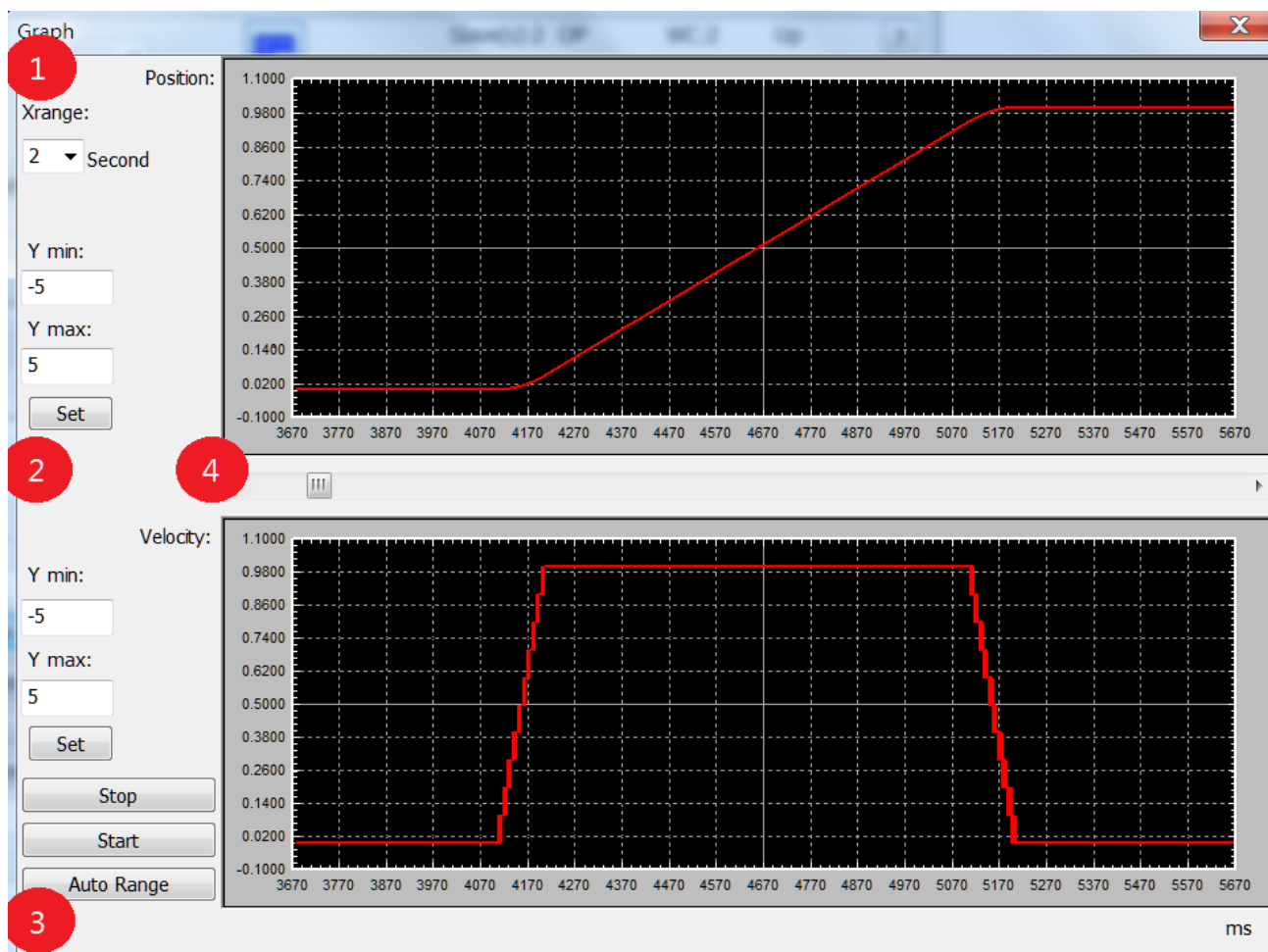
1. 使用者可先點選單軸運動資訊欄內的軸號來指定欲操作之單軸。
2. 設定主軸的軸號"Master No"。
3. 設定跟隨來源"Sync Source"分別為跟隨主軸的理論位置"Set Value"，或是跟隨主軸的實際位置"Actual Value"。
4. 最後設定電子齒輪比，分子為"Ratio Num"、分母為"Ratio Den"參數。
5. 開始電子齒輪跟隨運動請點選"GearIn"按鈕，操作軸號的軸狀態會變為 SyncMotion，即代表該操作軸已經處於跟隨某主軸狀態，此時只要針對跟隨主軸下達單軸運動控制命令，則該操作軸即會自動跟隨主軸進行電子齒輪跟隨運動控制。
6. 處於電子齒輪跟隨運動控制中的軸，如果要停止電子齒輪跟隨運動時，需要下"GearOut"指令。選擇跟隨軸然後點選"GearOut"按鈕後，該軸會從跟隨運動控制轉變為等速度運動控制，其速度是最後跟隨之速度，如果要真正停止運動則需要再下 Stop 指令。若想直接停止跟隨運動控制，可點選跟隨軸並按下停止按鈕"Stop"，或是快速停止"QuickStop"來停止運動控制並脫離跟隨運動。

## 單軸電子凸輪跟隨運動控制

1. 使用者可先點選單軸運動資訊欄內的軸號來指定欲操作之單軸。
2. 設定電子凸輪表單。
3. 設定主軸的軸號"Master No"。
4. 設定跟隨來源，"Sync Source"可選跟隨主軸的理論位置"Set Value"，或是隨主軸的實際位置"Actual Value"。
5. 最後設定電子凸輪表單數值倍率、表單編號以及絕對或相對位置處理。
6. 選一軸然後點選"CamIn"按鈕開始電子凸輪跟隨運動，該軸狀態會變為 SyncMotion，即代表該操作軸已經處於跟隨主軸狀態，此時只要針對主軸下達單軸運動控制命令，則該跟隨軸即會跟隨主軸進行電子凸輪跟隨運動控制。
7. 於電子凸輪跟隨運動控制中，如果要停止電子齒輪跟隨運動時，需要下"CamOut"或"Stop"指令。畫面上可針對操作軸點選停止按鈕"Stop"或快速停止"QuickStop"來停止運動並且解除跟隨運動。

### 單軸位置與速度簡易圖形顯示

1. 使用者可先點選單軸運動資訊欄內的軸號來指定欲顯示之單軸。
2. 點擊”Graph”按鈕。



項目	說明
(1)	X(時間)軸顯示範圍
(2)	Y(速度、位置)軸顯示範圍。
(3)	停止、開始與自動顯示範圍(Y 軸)
(4)	位置、速度顯示框

### 3.6.2. 群組運動控制操作頁面說明

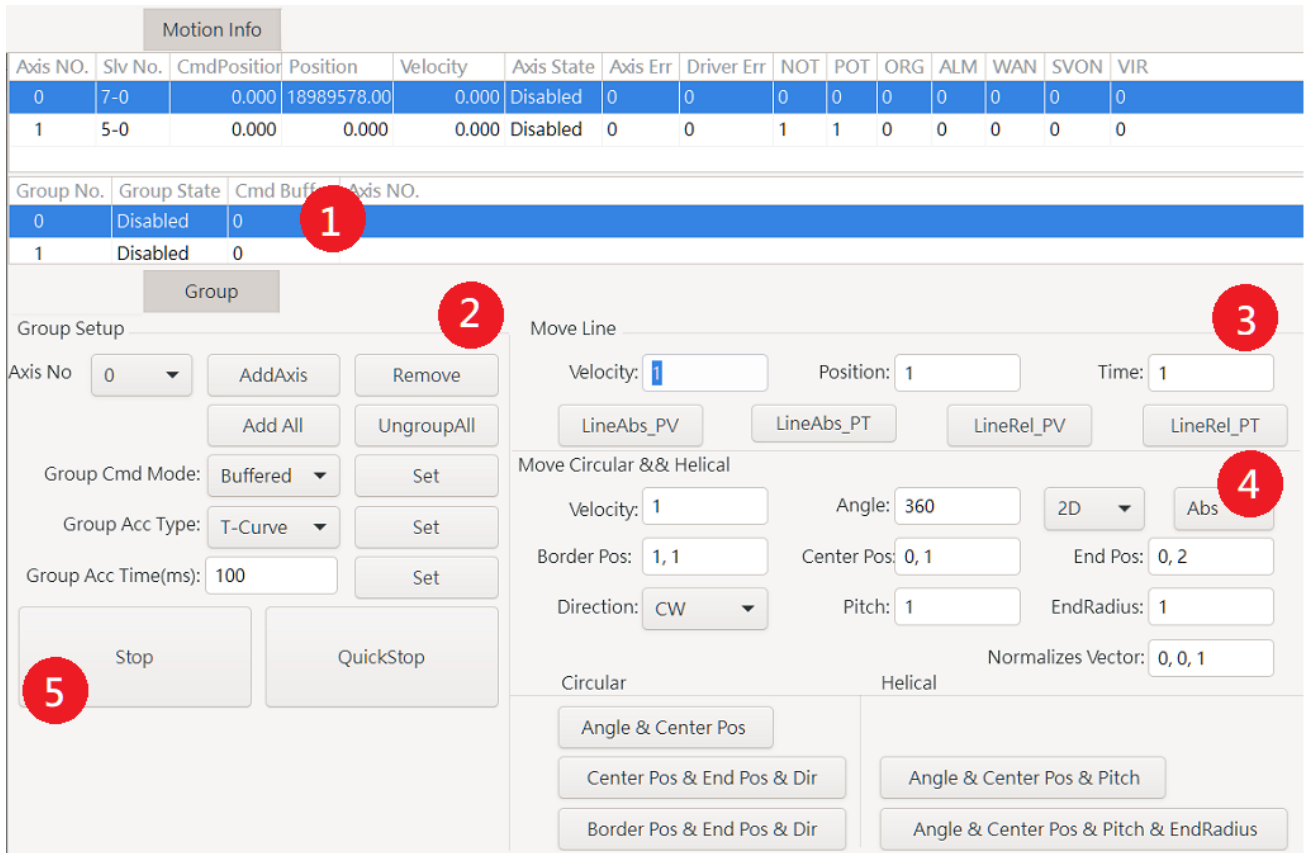


圖 3.21

表 3.15

項目	說明
(1)	群組運動資訊欄 (目前提供四個群組)
(2)	群組參數設定框 (由(1)處選擇要操作的群組)
(3)	群組直線運動運動控制框 (可以做簡易功能測試)
(4)	群組圓弧運動及螺旋運動控制框 (可以做簡易功能測試)
(5)	群組停止運動控制框

## 群組運動資訊欄

1. 群組運動資訊欄內容定義如下:

表 3.16

項目	說明
Group No.	群組編號
GroupState	群組狀態
Cmd Buffer	待被執行的命令數量
Axis No.	已加入群組之軸號

## 群組通用性操作及參數設定群組

1. 使用者可先點選群組運動資訊欄內的群組編號來指定欲操作之群組。
2. 群組在尚未加入任何軸號時其群組狀態為 Disabled，此時可從軸號清單"Axis No"中選擇欲加入之指定軸號，並點選加入"Add"按鈕將指定軸號加入群組內。加入軸的順序也很重要，要注意。
3. 使用者亦可點選移除"Remove"按鈕來將指定軸號從群組中移除。
4. 另外可點選取消群組"Ungroup All"按鈕來移除群組內全部軸號。
5. 使用者可根據相關參數點選設定按鈕"Set"進行參數的設定動作

## 群組直線運動控制

1. 使用者可先點選群組運動資訊欄內的群組編號來指定欲操作之群組。
2. 設定群組直線運動及 Profile 運動控制群組框內速度"Velocity"、位置"Position"等參數，各軸位置請使用逗點進行分隔，例如 2 軸直線運動時，位置欄內請輸入 50,100 即表示 2 軸直線運動的絕對或相對位置。
3. 設定群組運動命令執行模式"Group Cmd Mode"。
4. 點選絕對直線運動"Line Abs"或相對直線運動"Line Rel"按鈕進行群組直線運動控制。
5. 群組運動控制中若想停止，可點選停止按鈕"Stop"或快速停止"QuickStop"來停止群組運動控制。

## 群組圓弧運動控制

1. 使用者可先點選群組運動資訊欄內的群組編號來指定欲操作之群組。
2. 設定群組圓弧運動及螺旋運動控制群組框內速度"Velocity"、中心點位置"Center Position"、

旋轉角度"Angle"等參數，中心點位置請使用逗點進行分隔。

3. 設定群組運動命令執行模式"Group Cmd Mode"。
4. 點選圓弧運動"Angle & Center Pos"按鈕進行群組圓弧運動控制。
5. 群組運動控制中若想停止，可點選停止按鈕"Stop"或快速停止"QuickStop"來停止群組運動控制。

表 3.17

項目	說明
Angle & Center Pos	根據中心位置及旋轉角度，開始指定群組的圓弧補間移動。
Center Pos & End Pos & Dir	根據中心位置及終點位置與旋轉方向，開始指定群組的圓弧補間移動。
Border Pos & End Pos & Dir	根據圓弧上一點位置及終點位置與旋轉方向，開始指定群組的圓弧補間移動。

#### 群組螺旋運動控制

1. 使用者可先點選群組運動資訊欄內的群組編號來指定欲操作之群組。
2. 設定群組圓弧運動及螺旋運動控制群組框內速度"Velocity"、中心點位置"Center Position"、螺旋圈數"Angle"、螺旋螺距"Pitch"等參數，中心點位置請使用逗點進行分隔。
3. 設定群組運動命令執行模式"Group Cmd Mode"。
4. 點選螺旋運動"Angle & Center Pos & Pitch"按鈕進行群組螺旋運動控制。
5. 群組運動控制中若想停止，可點選停止按鈕"Stop"或快速停止"QuickStop"來停止群組運動控制。

表 3.18

項目	說明
Angle & Center Pos & Pitch	開始指定群組的螺旋補間移動。
Angle & Center Pos & Pitch & EndRadius	開始指定群組的圓錐螺旋補間移動。

### 3.6.3. 運動位置顯示頁面說明

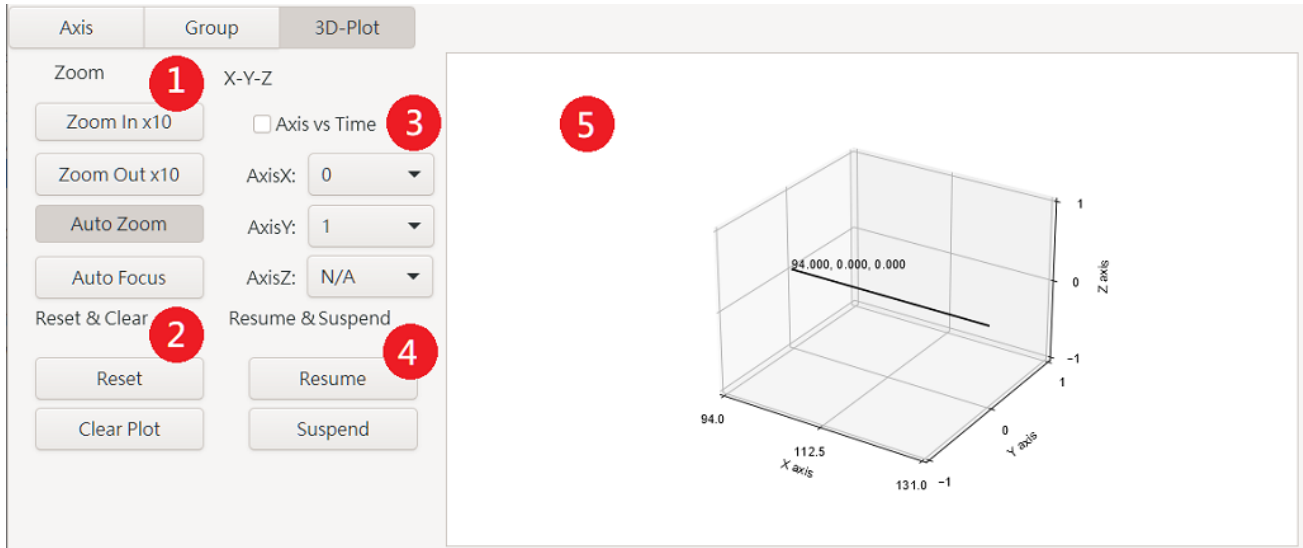


圖 3.22

注:只會顯示 1 個小時內的運動情況，其餘資料會逐漸消失。

表 3.19

項目	說明
(1)	運動位置顯示框縮放功能
(2)	運動位置顯示框清除功能
(3)	X-Y-Z 軸選擇框
(4)	運動位置顯示暫停功能
(5)	運動位置顯示框

#### 運動位置顯示框縮放功能

運動位置顯示框縮放功能，內容定義如下：

表 3.20


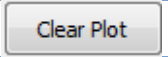
項目	說明
Zoom In x10	將圖放大 10 倍
Zoom Out x10	將圖縮小 10 倍
Auto Zoom	自動縮放至適當大小
Auto Focus	自動將視角移到適當位置



## 運動位置顯示框清除功能

運動位置顯示框清除功能，內容定義如下：

表 3.21

項目	說明
	將運動位置顯示框重置
	將圖清除

## X-Y-Z 軸選擇框

X-Y-Z 軸選擇框，內容定義如下：

表 3.22

項目	說明
	X 軸設定
	Y 軸設定
	Z 軸設定

## 運動位置顯示框

設置 X-Y-Z 軸後即可觀察其位置變化。

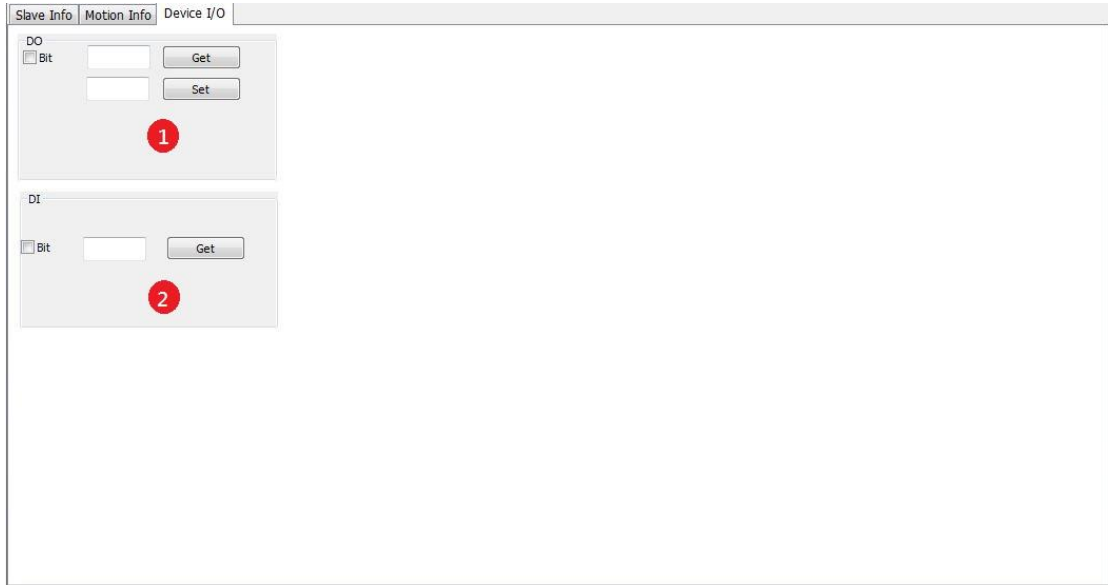
視角平移: 按住滑鼠中鍵後拖拉。

繞 X 軸及 Y 軸旋轉: 按住滑鼠左鍵後拖拉。

縮放: 按住滑鼠右鍵後拖拉。

## 3.7. 裝置 I/O 操作頁面

當使用者完成裝置初始化步驟後，如下圖所示裝置 I/O 操作頁面將致能，各控制項定義如下。



項目	說明
(1)	裝置 DO 控制群組框
(2)	裝置 DI 控制群組框

### 3.7.1. 裝置 DO 控制操作步驟

這是控制卡上的 DO，不是 EtherCAT 連線從站的 DO。

1. 點選"Set"按鈕來寫入全部通道的資料 (十進位方式)。
2. 點選"Get"按鈕則可取得 DO 資料 (十進位方式)。若"Bit"選項勾選時，於文字框內輸入位元編號，點選"Get"按鈕來取得位元編號之位元值。

### 3.7.2. 裝置 DI 控制操作步驟

這是控制卡卡上的 DI，不是 EtherCAT 連線從站的 DI。

1. 點選"Get"按鈕則可取得 DI 資料(十進位方式)。若"Bit"選項勾選時，於文字框內輸入位元編號，點選"Get"按鈕來取得位元編號之位元值。



### 3.8. PID 控制器操作介面說明

進入 OP 後，使用者即可開始 PID 控制器的相關操作

#### 3.8.1. PID 控制器控制操作頁面說明

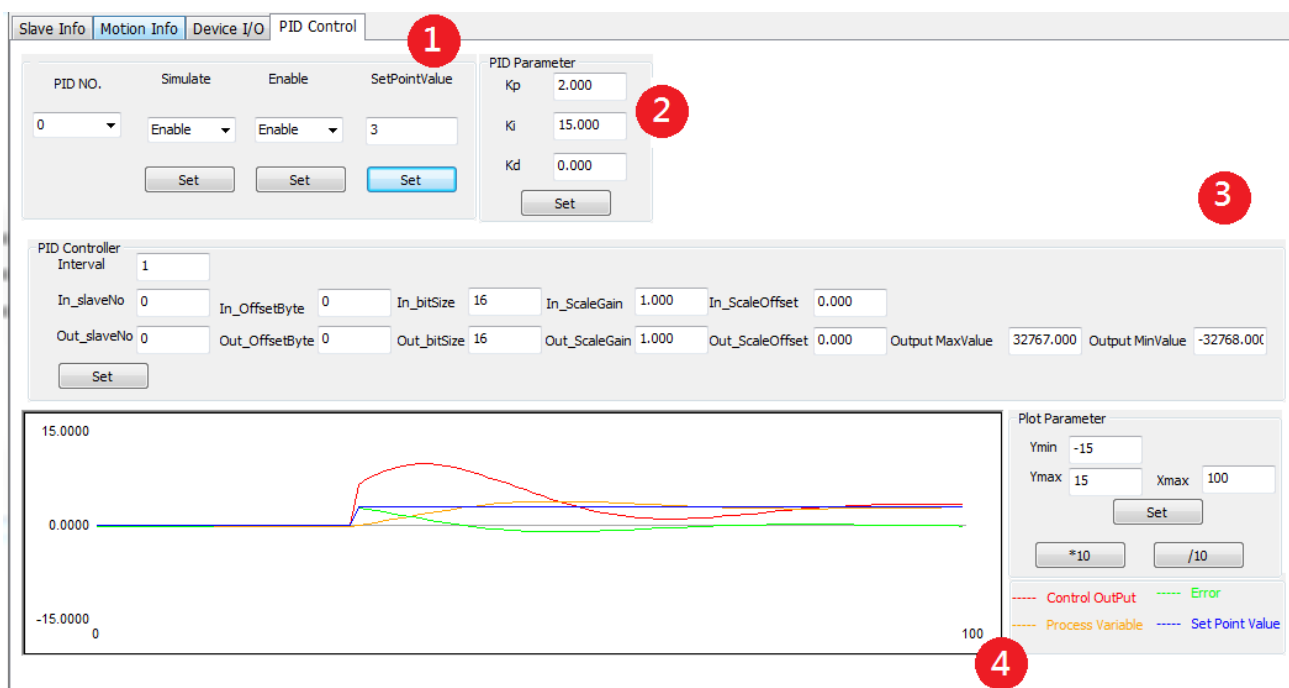


圖 3.23

表 3.23

項目	說明
(1)	PID 狀態設定欄
(2)	PID 控制器參數設定欄
(3)	PID 控制器輸入/輸出設定欄
(4)	PID 控制器圖形顯示控制欄

## PID 狀態設定欄內容定義

項目	說明
PID No.	PID 控制器編號
Simulate	是否啟用模擬模型
Enable	是否啟用 PID 控制器
SetPointValue	PID 控制迴路命令設定值

## PID 狀態設定

1. 選擇要進行操作的 PID 控制器編號。
2. 設定測量模組以及輸出模組。
3. 設定 PID 參數。
4. 設定 **Simulate** 為 Enable 啟用模擬模型，或是設定 **Simulate** 為 disable 啟用測量模組以及輸出模組。
5. 設定 **Enable 控制** 為 Enable 來啟用 PID 控制器。
6. 設定 **SetPointValue** 給予 PID 控制迴路命令設定值。

## PID 輸入/輸出設定欄內容定義

項目	說明
Interval	PID 控制器執行間隔時間，單位為通訊週期
In_slaveNo	PID 控制器測量模組之從站編號
In_OffsetByte	PID 控制器測量模組 TxPdo 偏移量 (用於指定 AD channel)
In_bitSize	PID 控制器測量模組資料大小，單位:bit，(AD 的資料大小)
In_ScaleGain	PID 控制器測量模組輸入增益 (例如，AD channel 將物理量轉成有符號的整數數值，而我們利用此增益與對應的偏移量設定將該有符號的整數數值轉成對應的物理量來提供迴路做計算)
In_ScaleOffset	PID 控制器測量模組輸入偏移量(同上說明，使用來轉換有符號的整數數值成物理量來提供迴路做計算)
Out_slaveNo	PID 控制器輸出模組之從站編號

<b>Out_OffsetByte</b>	PID 控制器輸出模組 RxPdo 偏移量(用於指定 DA channel)
<b>Out_bitSize</b>	PID 控制器輸出模組資料大小，單位:bit，(DA 的資料大小)
<b>Out_ScaleGain</b>	PID 控制器輸出模組輸出增益（例如，控制器算出的控制量要使用 DA channel 來輸出到致動器。控制量是有符號的數值，我們利用此增益與對應的偏移量將該有符號的數值轉換成對應的 DA 輸出來驅動致動器)
<b>Out_ScaleOffset</b>	PID 控制器輸出模組輸出偏移量 (同上說明，使用來轉換有符號的數值成物理量輸出來提供來驅動致動器)
<b>Output MaxValue</b>	PID 控制器輸出模組限制最大值
<b>Output MinValue</b>	PID 控制器輸出模組限制最小值

### 3.9. EtherCAT 診斷

進入 OP 後，使用者即可開始 EtherCAT 診斷的相關操作

每模組都會有 4 個 Port，一般來說，模組的 EtherCAT Port 只會用到兩個，通常 EtherCAT In Port 為 Port0，EtherCAT Out Port 為 Port1

#### 3.9.1. EtherCAT 診斷操作頁面說明

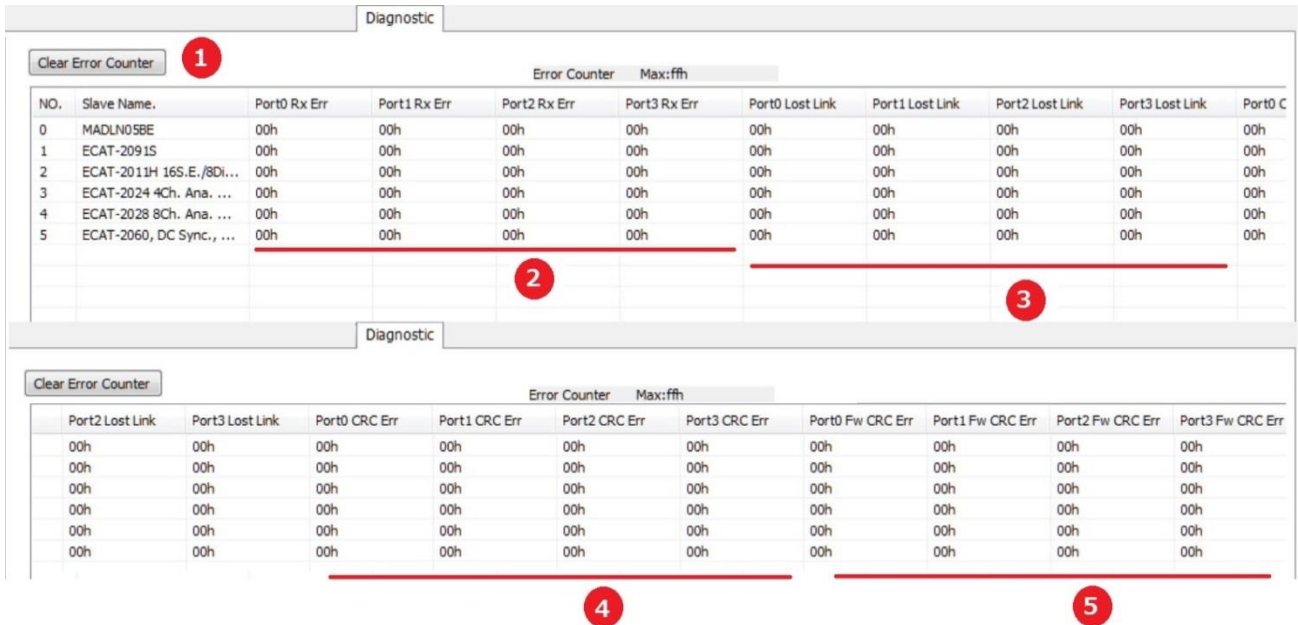


圖 3.24

表 3.24

項目	說明
(1)	清除錯誤計數器，錯誤計數器最大值為 255(Dec) 0xff(Hex)
(2)	無效幀(Rx 接收) 錯誤計數器
(3)	連接丟失 錯誤計數器
(4)	無效幀(CRC 校驗) 錯誤計數器
(5)	轉發 CRC 錯誤計數器

## 3.9.2. EtherCAT 診斷說明

---

### 無效幀(Rx 接收)

### 無效幀(CRC 校驗)

---

一個端口上的數據幀破壞，從站就會把該端口的 Invalid Frame Counter(無效幀計數器) 加 1

有時候，即使硬件信號到達了相鄰的從站，接收到的信號卻與最初發送的不一致

數據包發生損壞，最有可能的原因是：

- 如果該錯誤計數零星增加，極可能是由於外部 EMC 干擾。
- 如果該錯誤計數快速且穩步增加，極可能是由於設備損壞。

### RX Errors:

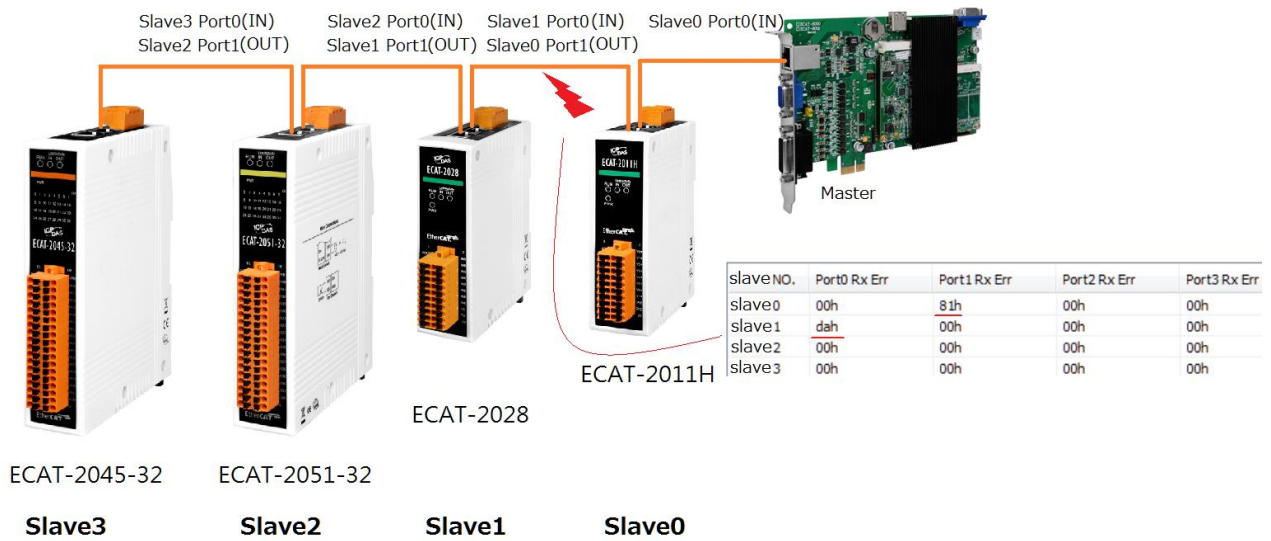
- 個別符號錯誤 (被特定的硬件解碼器識別為無效)。
- 在數據幀內部或者外部都可能發生 (即使是在沒有數據幀傳輸的時候，每個物理接口都會傳輸 idle 即表示空間的符號)

### CRC Errors:

- 整個數據幀的循環冗餘校驗 (CRC) 錯誤。
- 在數據幀內發生 (只有 Ethernet 數據幀才進行校驗)。

無效幀(Rx/CRC)計數器不為零的第一個端口 → 極有可能就是故障點





在故障點採取什麼措施？

- 檢查與上個從站之間的連接電纜：
  - EtherCAT 網線的走向是否靠近電源線或者干擾源？
  - 是否自製電纜的接頭做工很差？
  - 電纜的遮蔽層是否接地良好？
- 檢查故障點的站和前一個從站：
  - 是否供電不足 (供給本站的電流太小，例如：EBUS 電流)？
  - 兩個從站的接地點不是等電位？(不同接地之間的環流可能導致數據損壞)
- 試著替換該從站和前一個從站，或者交換二者的順序，以查看錯誤是跟隨某個從站還是出現在原先的位置。

對於 EMC 干擾，應該會導致 RX 和 CRC 錯誤都有增加 (即使二者增加的機率可能不同)，因為外部干擾不可能與通訊同步，於是在 Ethernet 數據幀內部和外部都會破壞數據。

完全不平衡的計數值 (RX 錯誤很多，而沒有 CRC 錯誤，或者很 CRC 錯誤而沒有 RX 錯誤) 可能意味著其中一個從站的內部硬件錯誤。

---

## 連接丟失

---

一個端口上的物理連接中斷，從站就會把該端口的 Link Lost Counter(連接丟失計數器) 加 1

**Link loss ( 連接丟失 )，最有可能的原因是：**

- 電纜或接頭損壞 ( 電纜連接 )，接觸彈片不夠力或者氧化 ( Ebus 連接 )。
- 一個或者幾個從站掉電

## 4. 函式庫使用流程說明

### 4.1. 裝置操作流程

如圖 4.1 所示,使用者可調用 *GetDeviceCnt* 函式來取得有多少裝置可以建立通訊,並根據 Card ID 調用 *OpenDevice* 函式建立和裝置間的通訊。於建立通訊後裝置並未開始 EtherCAT 操作任務,此時可以進行基本的裝置操作。

首先可調用 *GetDeviceState* 函式來取得目前整個 EtherCAT 網絡的各種狀態,包含目前連接從站數、主站的 AL 狀態 (INIT, PreOP, SafeOP, OP)、網路連接狀態(連線/斷線)等,或是對從站調用 *GetSlaveInfo* 函式取得從站相關資訊。若需讀寫 SDO 物件數值,可調用 *GetSlaveSdoObject*, *SetSlaveSdoObject* 函式來完成。

**當使用者需要開始 EtherCAT 操作任務時,請先使用工具程式編輯 EtherCAT 網絡資訊,並利用工具程式將 EtherCAT 網絡資訊寫入裝置內部。** EtherCAT 網路資訊沒寫入到控制卡內時,該卡無充足的從站資訊,所以無法啟動來做控制。EtherCAT 網路資訊可以規劃數個,有不同編號,可以都下載到控制卡裡面。在這之後,使用者的主程式就可以調用 *StartDeviceOpTask* 函式,在參數中指定某一編號的網路資訊來開始 EtherCAT 操作任務。如果指定的網路資訊與實際掃描出來的連線資訊室一樣時,等一段時間後,裝置進入到 OP 狀態(代表所有從站也都進入 OP 狀態),使用者就可以進行從站 PDO 物件讀寫以及運動控制等等的操作。如果指定的網路資訊與實際掃描出來的連線資訊室不一樣,則會回報錯誤。

調用 *StopDeviceOpTask* 函式可以控制卡內部離開 OP 模式,並且停止操作任務;而進一步調用 *CloseDevice* 函式則可以關閉與該裝置的通訊。

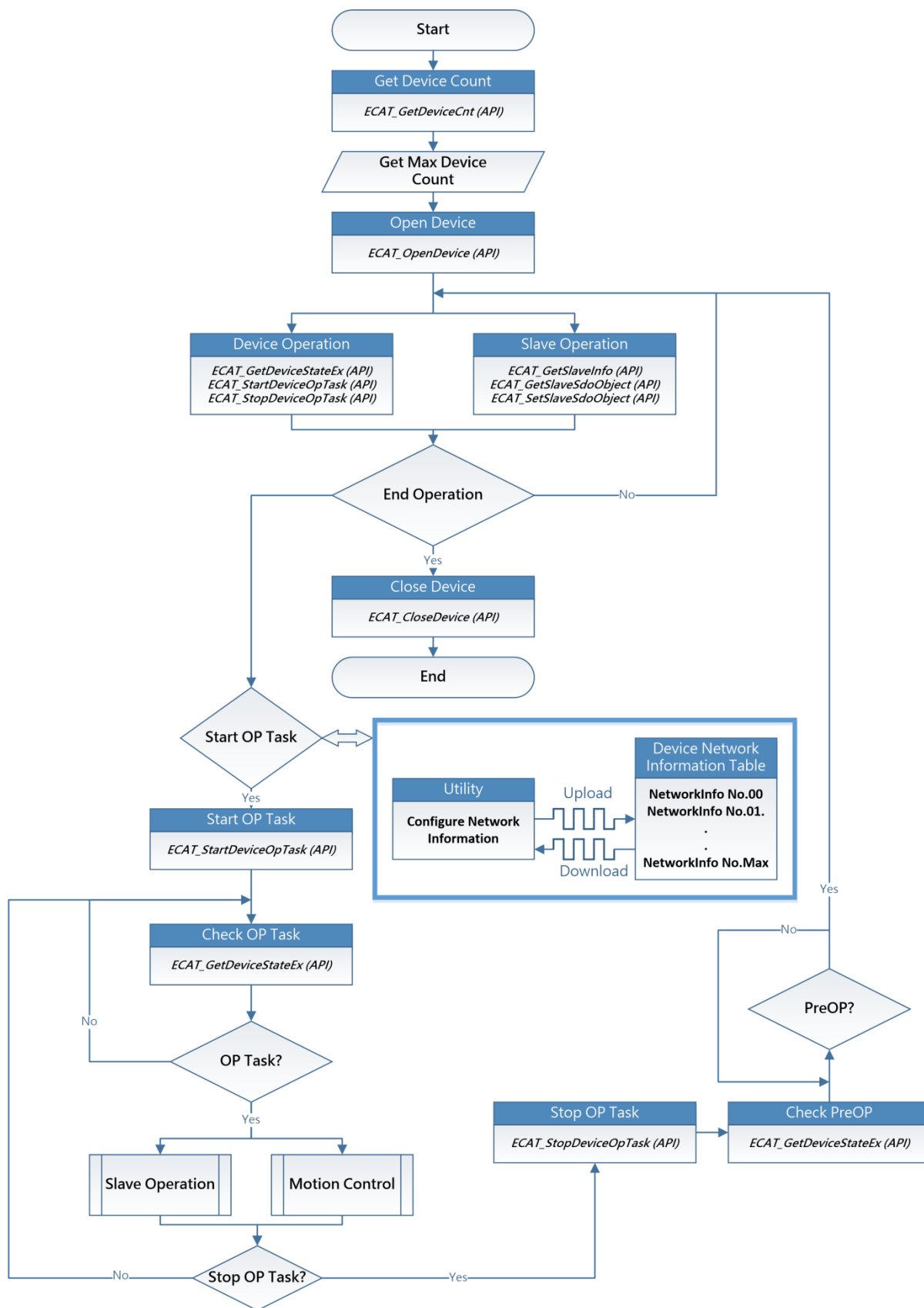


圖 4.1

## 4.2. 從站模組操作流程

如圖 4.2 所示從站模組操作可分成 2 個部分:(1)裝置基本操作時。(2)EtherCAT 操作任務時。裝置基本操作時其裝置尚未開始 EtherCAT 操作任務，只提供調用 GetSlaveInfo, GetSlaveSdoObject, SetSlaveSdoObject 等函式，當裝置切換成 EtherCAT 操作任務時則可調用 RxPdo, TxPdo 等讀寫函式。

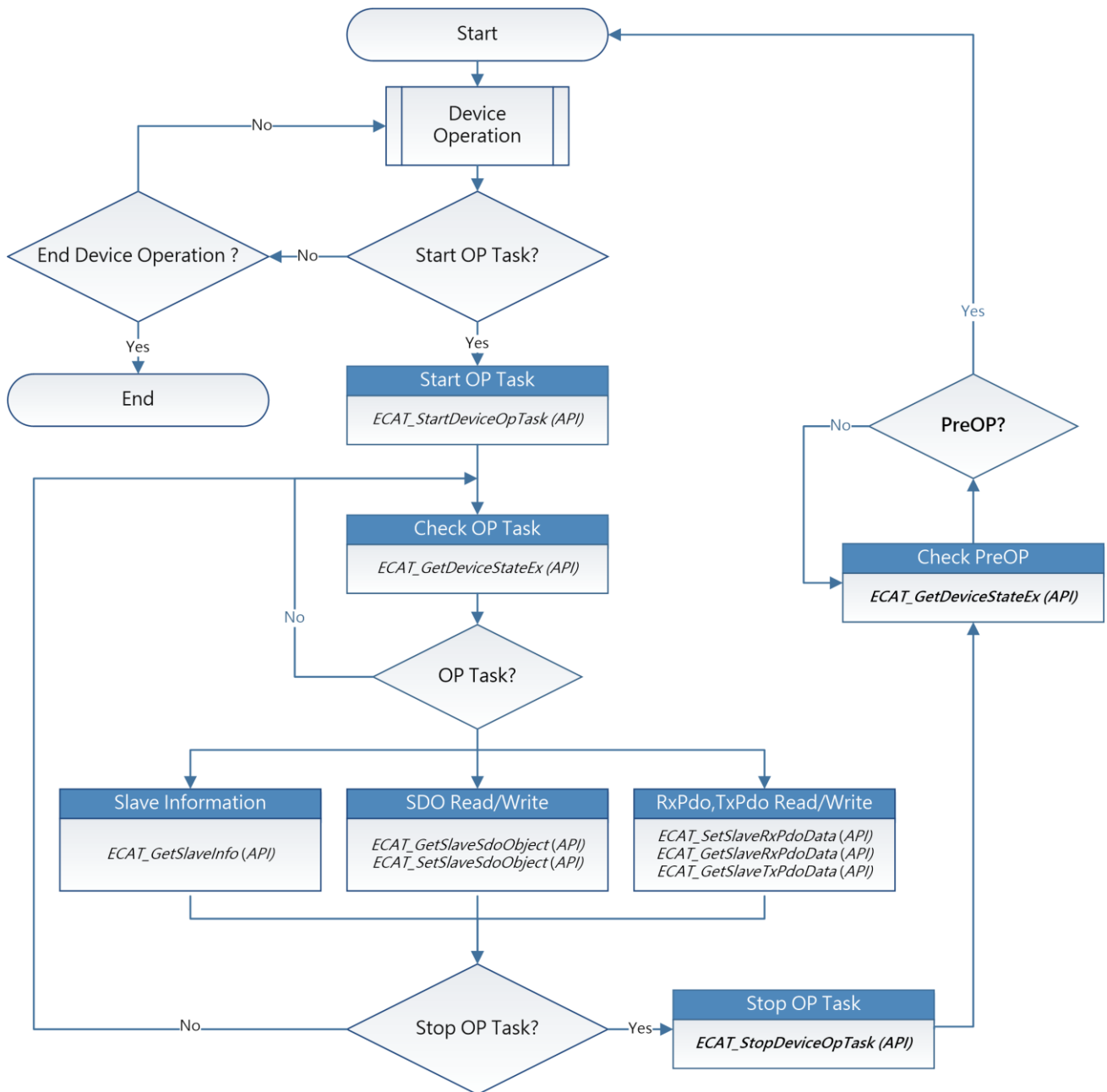


圖 4.2

## 4.3. 運動控制操作流程

### 4.3.1. 運動控制初始化流程

如圖 4.3 所示，在開始運動控制的操作前需先進行初始化動作，初始化會將從站編號轉換成軸號，使得裝置可以針對該軸號進行運動控制 PDO 操作。使用者可調用 *Mclnit* 函式進行運動控制的初始化，當初始化完成後，即可開始運動控制相關操作，包含有單軸原點復歸、單軸運動控制、單軸錯誤狀態處理及群組運動控制等操作

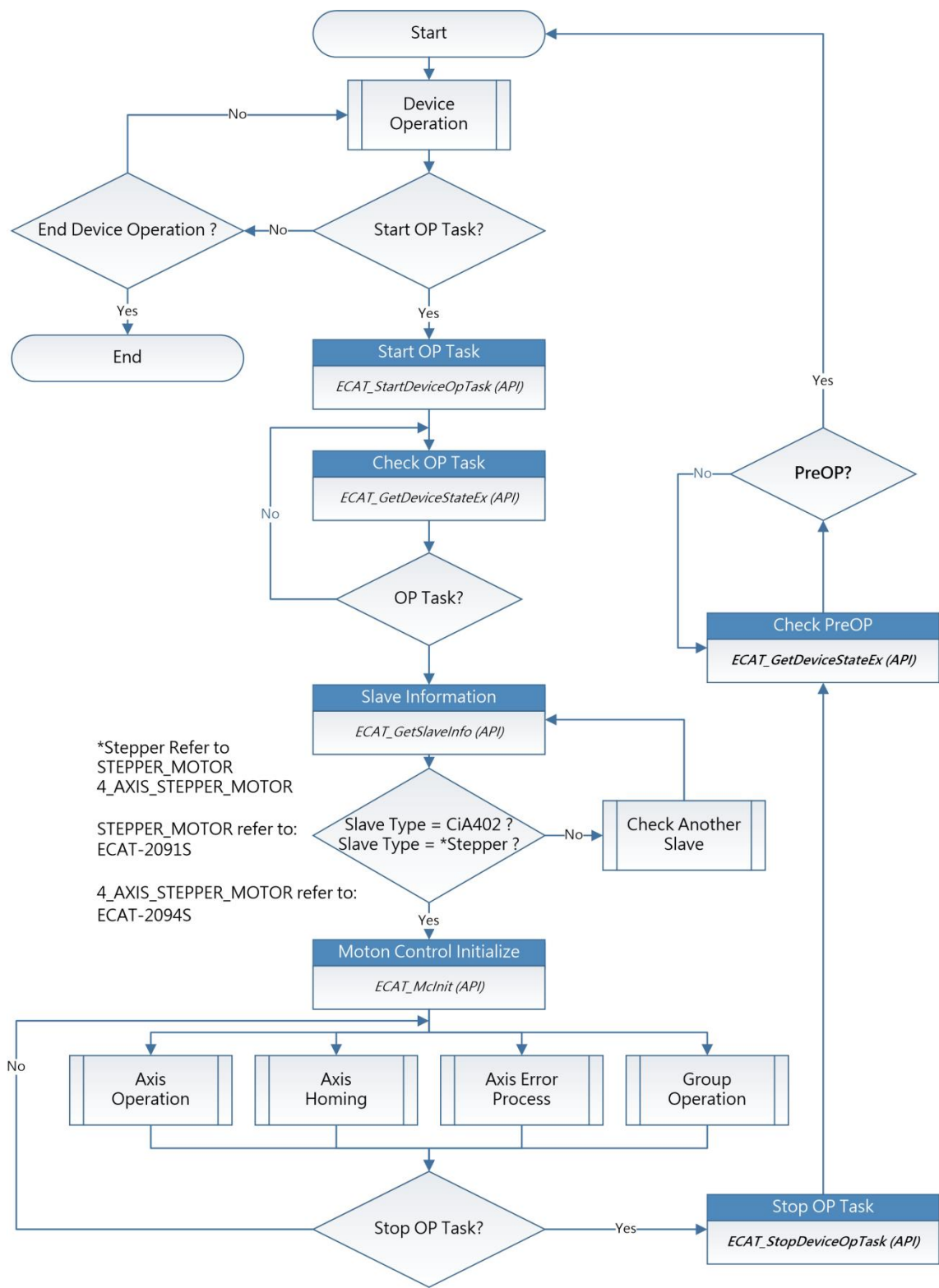


圖 4.3

### 4.3.2. 單軸運動控制操作流程

如圖 4.4 所示，在進行單軸操作之前，使用者需要對各項參數進行配置，在使用者設定完成這些參數後，使用者可調用 *McGetAxisState* 來取得單軸狀態，如果狀態為 Standstill，表示單軸目前處於靜止狀態，準備接收新的運動命令。此時使用者可調用運動控制相關函式，於調用函式成功後軸狀態會從 Standstill 狀態轉換成 Discrete Motion (單一指令運動), Continuous Motion (連續運動), Synchronized Motion (同步運動)等狀態，這表示單軸移動中。

當軸移動中時，使用者可調用停止函式來停止單軸，當軸狀態從移動中狀態轉變成 Standstill 時，即表示停止的命令已完成，並且可以接收新的運動命令。若在移動中發生錯誤時，軸狀態也會從移動中狀態轉換成 Error Stop 狀態，此時使用者需針對錯誤的情況做相關的處理



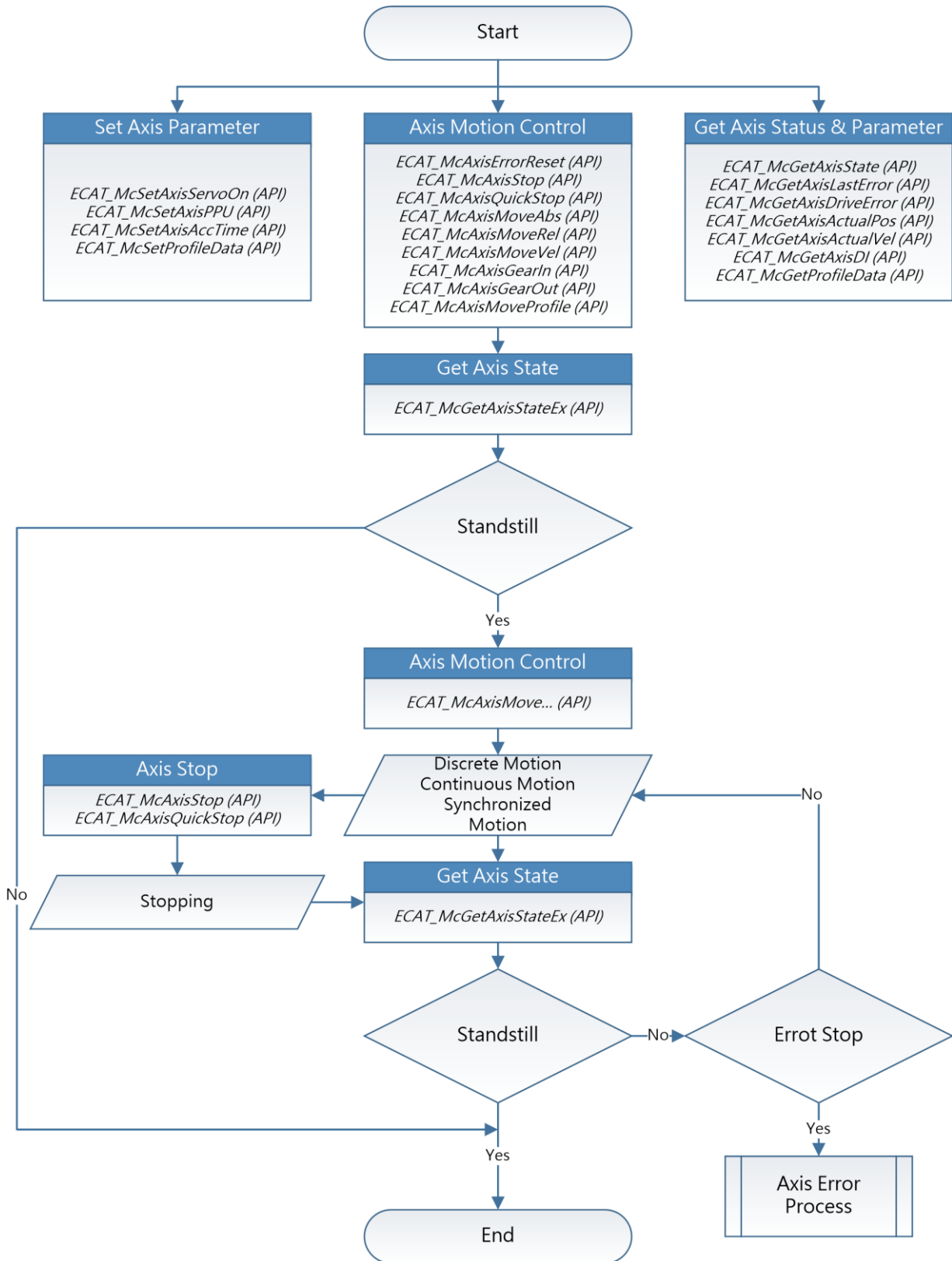


圖 4.4

### 4.3.3. 單軸自動原點復歸流程

如圖 4.5 所示，在開始進行原點復歸前需要針對原點復歸方法、原點復歸速度、原點復歸加速度、原點復歸偏移量等參數進行設定。如同單軸運動控制一樣，使用者可調用 *McGetAxisState* 來取得單軸狀態，如果狀態為 Standstill，表示單軸目前處於靜止狀態，準備接收新的運動命令。此時可調用原點復歸函式，於調用函式成功後軸狀態會從 Standstill 狀態轉換成 Homing 狀態，這表示單軸原點復歸中。

當單軸原點復歸時，使用者可調用停止函式來停止單軸，當軸狀態從原點復歸中狀態轉變成 Standstill 時，即表示停止的命令已完成，並且可以接收新的運動命令。若在原點復歸中發生錯誤時，軸狀態也會從移動中狀態轉換成 Error Stop 狀態，此時使用者需針對錯誤的情況做相關的處理

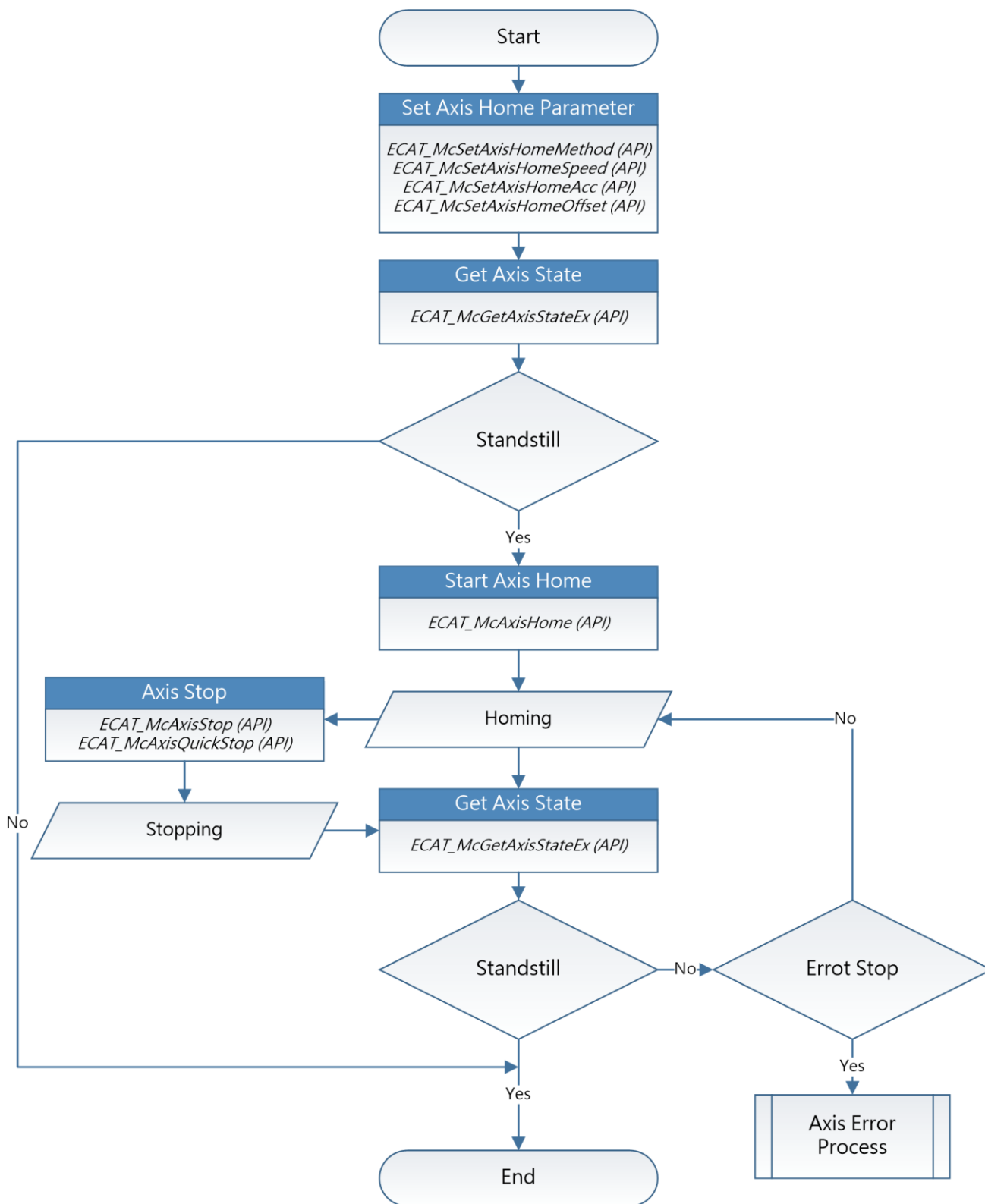


圖 4.5

#### 4.3.4. 單軸錯誤狀態處理流程

如圖 4.6 所示，當單軸狀態轉變成 Error Stop 時，可調用 *McGetAxisLastError* 取得錯誤代碼，由錯誤代碼(-1107 / -1108)來判斷是否為驅動器產生的錯誤。在錯誤處理上包含兩部份:(1)若錯誤不是驅動器所產生的錯誤，使用者可經由錯誤代碼確認錯誤原因後，然後調用 *McAxisErrorReset* 來清除錯誤，軸狀態將從 Error Stop 轉換成 Standstill。(2) 若錯誤是驅動器所產生的，使用者可再調用 *McGetAxisDriveError* 來取得驅動器的錯誤代碼，並從錯誤代碼確認錯誤原因後，再調用 *McAxisErrorReset* 來清除錯誤。

驅動器產生的錯誤有些可以藉由下命令方式清除，有些錯誤則無法由下命令方式清除，當下達清除命令都無法將軸狀態轉變成 Standstill 時，請重啟驅動器來清除驅動器錯誤，重啟驅動器時，會產生通訊錯誤，請參考章節“4.4 通信錯誤處理流程”，通訊恢復後，再調用 *McAxisErrorReset* 來清除錯誤。

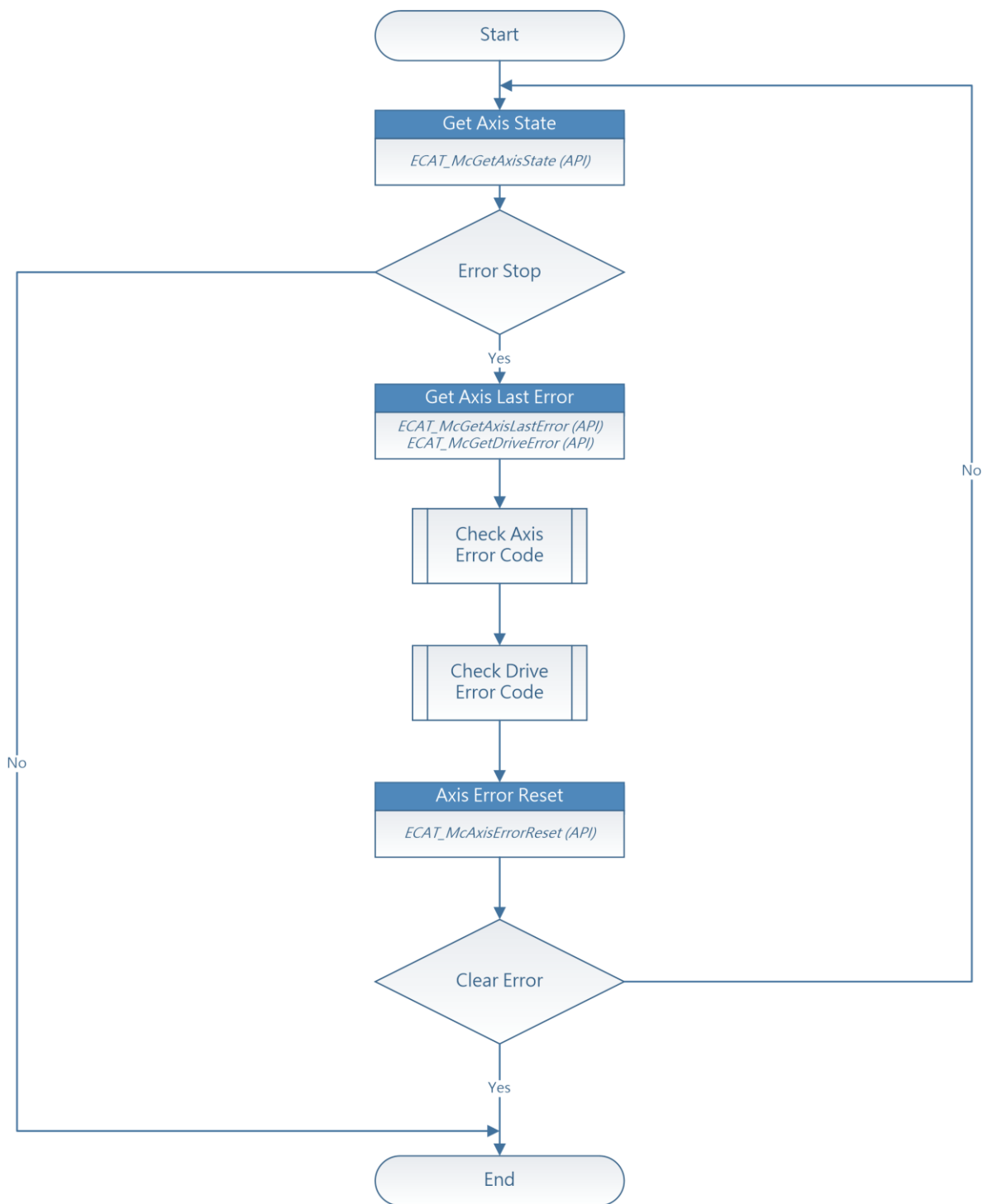


圖 4.6

### 4.3.5. 群組運動控制操作流程

使用者可藉由群組運動控制來達成補間移動，在使用群組運動前，使用者需針對指定群組進行群組化，藉由調用 `McAddAxisToGroup` 來將指定軸號加入到指定群組內，調用 `McRemoveAxisFromGroup` 將指定軸號從指定群組中移除，或者調用 `McUngroupAllAxes` 來將全部軸號從群組中移除，當完成群組化後，即可開始群組運動的各項操作，如圖 4.7 所示。

如同單軸運動控制一樣，使用者可調用 `McGetGroupState` 來取得群組狀態，如果狀態為 `Standby`，表示群組目前處於停止狀態，準備接收新的運動命令。此時針對運動控制功能調用運動控制函式，於調用函式成功後軸狀態會從 `Standby` 狀態轉換成 `Moving`，這表示群組目前移動中。

當群組移動中時，使用者可調用停止函式來停止群組運動，當群組狀態從移動中狀態轉變成 `Standby` 時，即表示停止的命令已完成，並且可以接收新的運動命令。若在移動中發生錯誤時，群組狀態會從移動中狀態轉換成 `Error Stop` 狀態，此時使用者需針對錯誤的情況做相關的處理

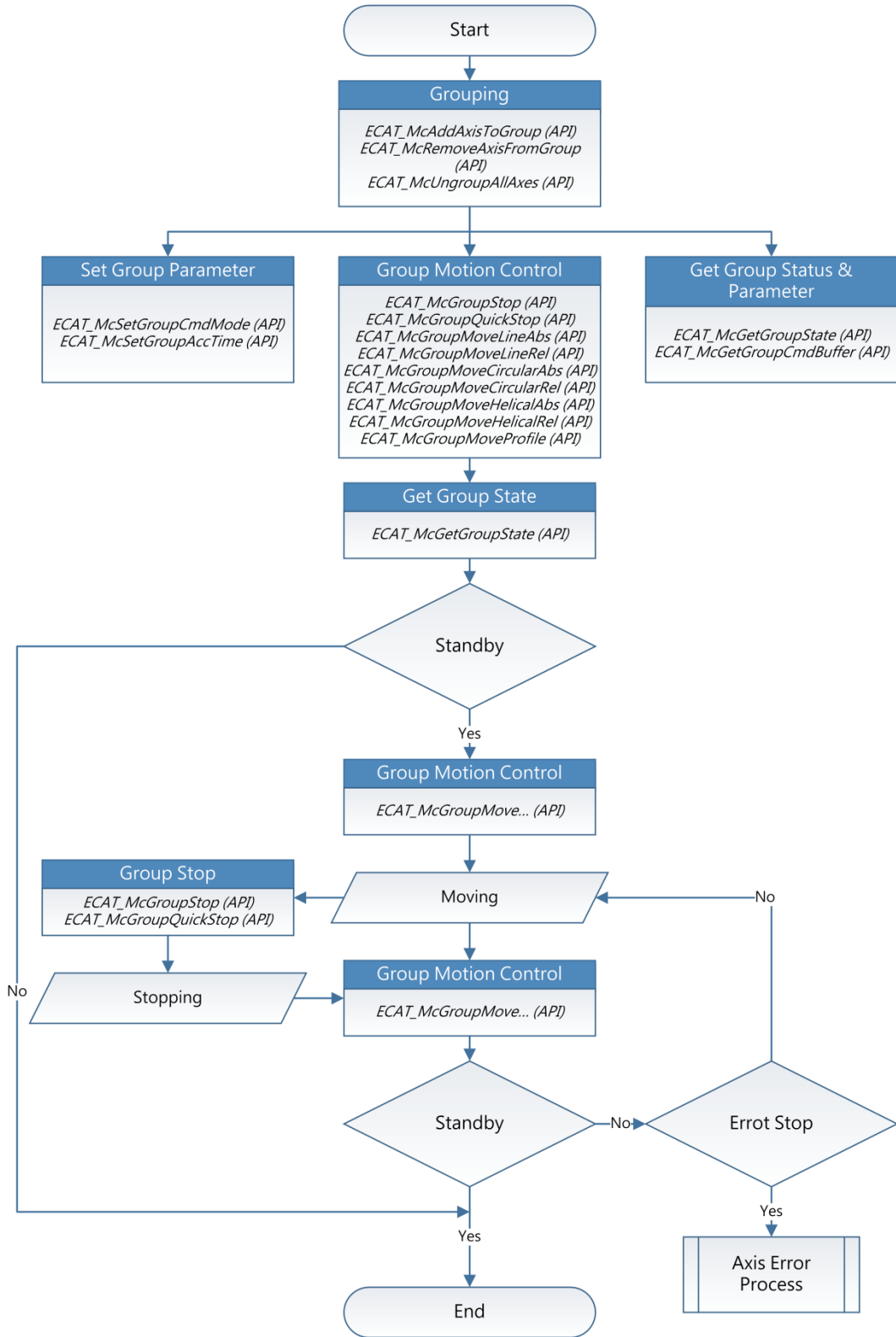


圖 4.7

### 4.4. 通信錯誤處理流程

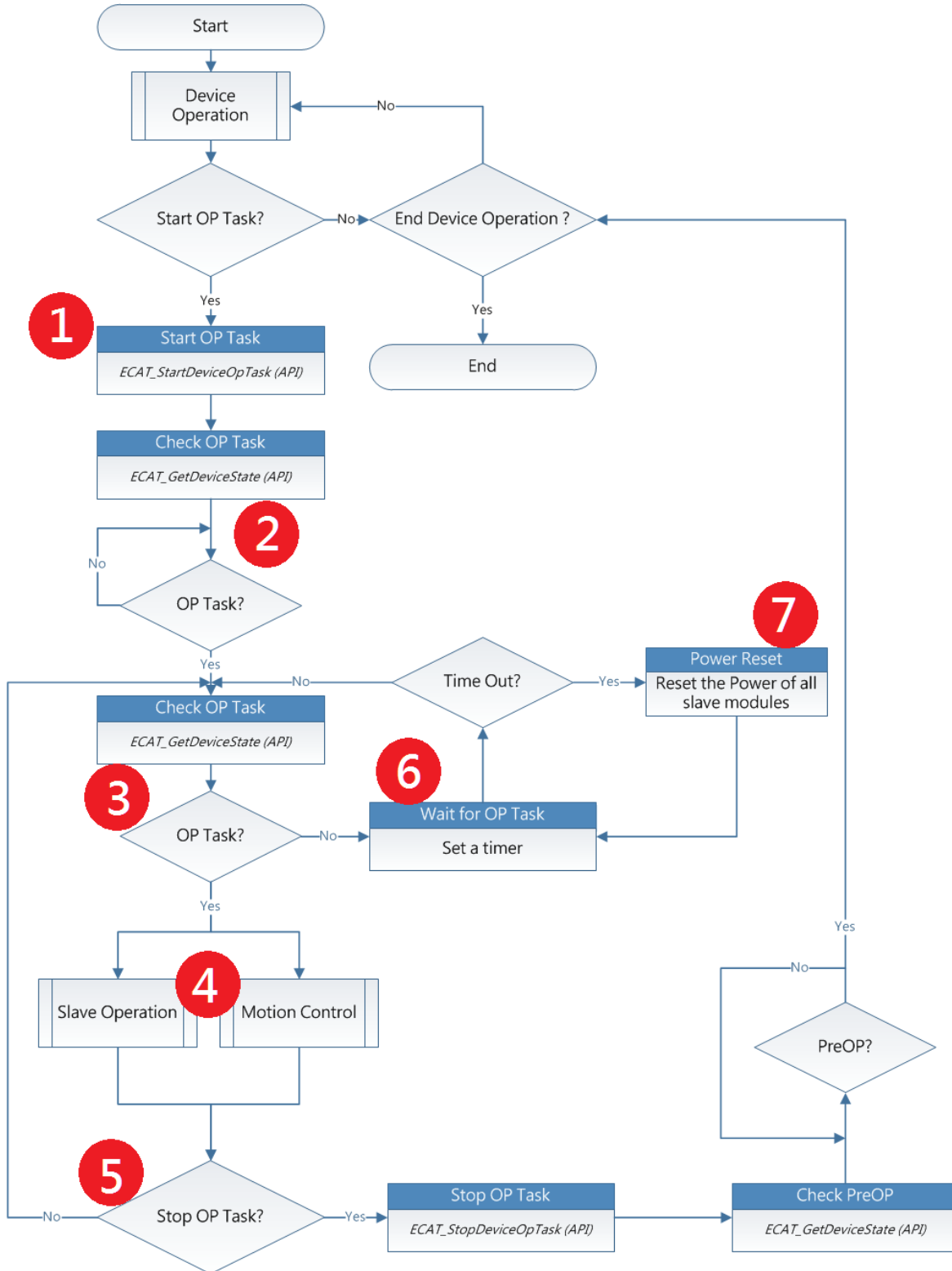


圖 4.8



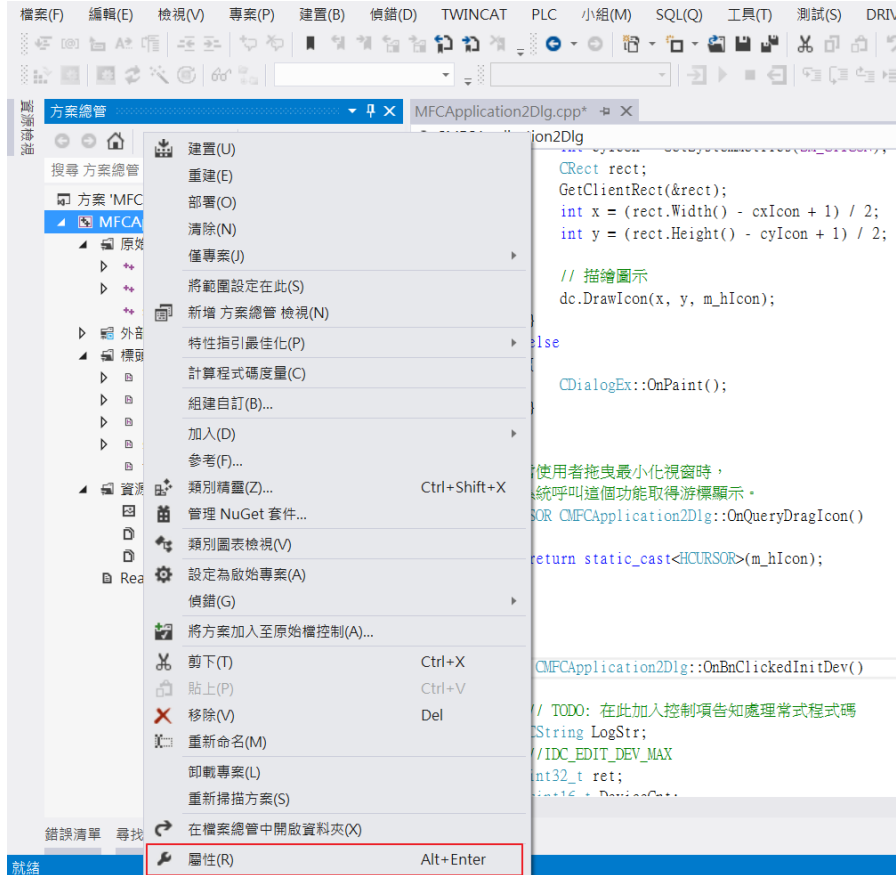
流程:

- 步驟 1: 使用 ECAT\_StartDeviceOpTask 進入 EtherCAT 操作任務。
- 步驟 2: 使用 ECAT\_GetDeviceState 讀取當前 EtherCAT 網路狀態，並且等待至當前 EtherCAT 網路狀態為"OP"狀態，待到"OP"狀態時，記錄當前的 WC(Working Counter)與從站數量。
- 步驟 3: 當網路狀態為"OP"狀態之後，除非使用 ECAT\_StopDeviceOpTask 退回"PreOP"狀態，否則網路狀態都應為"OP"狀態，因此，需要使用 ECAT\_GetDeviceState 定時檢查當前 EtherCAT 網路狀態是否為"OP"狀態，若當前狀態不為"OP"，至步驟 6。  
同時比較當前 WC 是否與在步驟 2 記錄的 WC 相同，若不同，至步驟 6。  
當從站數量與在步驟 2 記錄的從站數量不同，代表有斷線情形，恢復連線後，至步驟 6。
- 步驟 4: 當網路狀態為"OP"狀態之後，使用者可以開始讀寫 DIO 以及使用運動控制函式。
- 步驟 5: 回到步驟 3。
- 步驟 6: 當發生通信問題，當前網路狀態脫離"OP"，或 WC 變化，或從站數量變化，當通信恢復正常，網路狀態會再次變為"OP"，且 WC 會與步驟 2 記錄的 WC 相同，通信恢復正常，至步驟 3，但是有些模組通訊失敗後可能無法回到"OP"狀態，或者 WC 無法回到步驟 2 記錄的 WC，使用者可以設定一計時器，當狀態連續一段時間都無法恢復通信，至步驟 7。  
使用者可使用應用程式測試恢復通信時間。
- 步驟 7: 將從站模組斷電後復電，到步驟 6，等待網路狀態回到"OP"。

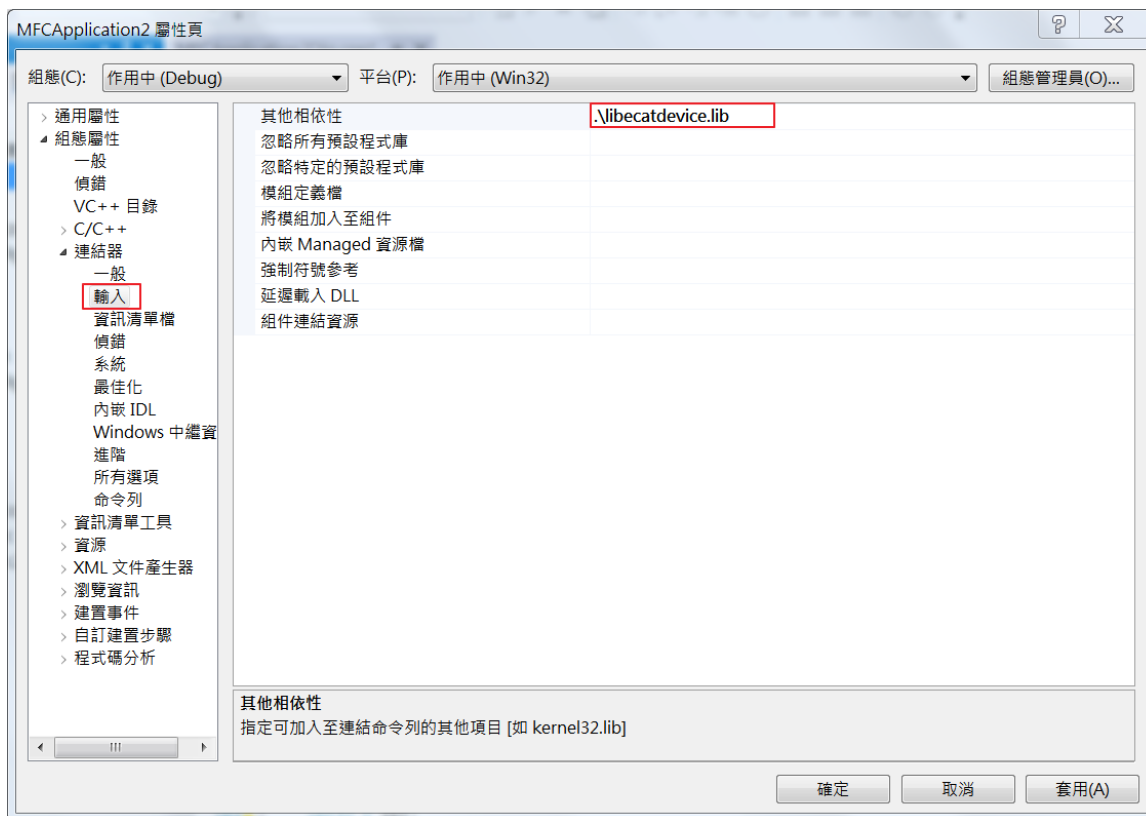
## 4.5. 函式庫在 Windows 程式編程說明

### 4.5.1. Visual Studio 工具使用說明

1. 新建專案，於主選單點選檔案/新增/專案
2. 於建立的專案上點選滑鼠右鍵選單中的屬性，開啟屬性頁面



3. 於屬性頁面下，選擇連結器/輸入/其他相依性，指定 libecatdevice.lib 顯示加載 dll



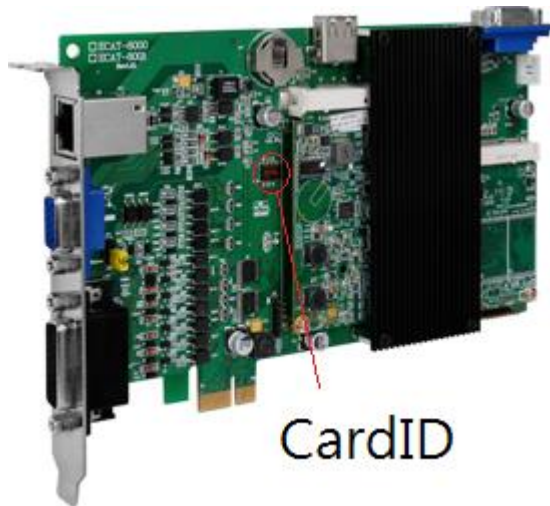
4. 於程式編程時將標頭檔包含進設計文件中

## 5. 裝置操作函式集說明

### 5.1. ECAT\_GetDeviceCnt

**說明:**

取得目前可使用裝置數量



**格式:**

```
int32_t ECAT_GetDeviceCnt (uint16_t *DeviceCnt, uint8_t CardID[])
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceCnt	uint16_t	OUT	可使用裝置數量
CardID	uint8_t *	OUT	各裝置的 Card ID (卡號)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceCnt, i;
uint8_t CardID[CARD_DEVICE_NO_MAX];
ret = ECAT_GetDeviceCnt(&DeviceCnt, CardID);
if(ret < 0)
{
    printf("Failed to get device count:%d\n", ret);
}
else
{
    printf("Device Count%u \n", DeviceCnt);
    for(i=0;i< DeviceCnt;i++)
    {
        printf("CardId[%u] = %u \n", i, CardID[ i ]);
    }
}
```

## 5.2. ECAT\_OpenDevice

### 說明:

開啟指定裝置編號(卡號)來做通訊操作。

注: (1)一張卡片只能被一個程序開啟，若其他程序開卡，回傳 -1304。

(2)若回傳 -1211，代表 PC 可能進入過休眠，或是 PC 了開啟快速開機，請將休眠及快速開機關閉，將 PC 重新啟動再嘗試開卡。

(3)若回傳 -1206，可能是因為控制卡尚未完成初始化，請在 PC 開機 1 分鐘後再嘗試開卡。若一直無法開卡，請將 PC 關機(請確保完成關機程序，不要"重新開機")後再開機，等待 1 分鐘後嘗試開卡，如果還是不行(回傳 -1206)，請連繫客服人員。

### 格式:

```
int32_t ECAT_OpenDevice(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號(卡號)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_OpenDevice(DeviceNo);
if(ret < 0)
{
    printf("Failed to open device:%d\n", ret);
}
else
{
    printf("Open device successfully! \n");
}
```

---

## 5.3. ECAT\_CloseDevice

### 說明:

關閉指定裝置編號(卡號)的通訊操作。

### 格式:

```
int32_t ECAT_CloseDevice(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號(卡號)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_CloseDevice(DeviceNo);
if(ret < 0)
{
    printf("Failed to close device:%d\n", ret);
}
else
{
    printf("Close device successfully! \n");
}
```

---

## 5.4. ECAT\_GetDeviceSerialNo

### 說明:

取得硬體序號。

### 格式:

```
int32_t ECAT_GetDeviceSerialNo(uint16_t DeviceNo, uint8_t *SerialNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號(卡號)
SerialNo	uint8_t *	OUT	硬體序號(8 Bytes 的陣列大小)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t SerialNo[8];
ret = ECAT_GetDeviceSerialNo(DeviceNo, SerialNo);
if(ret < 0)
{
    printf("Failed to get device serial No.:%d\n", ret);
}
else
{
    printf("serial number = %x %x %x %x %x %x %x %x\n",
        SerialNo[0],SerialNo[1],SerialNo[2],SerialNo[3],
        SerialNo[4],SerialNo[5],SerialNo[6],SerialNo[7]);
}
```

## 5.5. ECAT\_GetDllVersion

### 說明:

取得當前使用的 dll 版本資訊。

### 格式:

```
int32_t ECAT_GetDllVersion(char *Version, uint16_t *Size);
```

### 參數:

名稱	類型	IN or OUT	說明
Version	char*	OUT	當前使用的 dll 版本資訊
Size	uint16_t	OUT	回傳的 Version 大小 單位:Byte

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
char Version[512];
uint16_t size;
ret = ECAT_GetDllVersion(Version, &size);
if(ret < 0)
{
    printf("Failed to get dll version:%d\n", ret);
}
else
{
    printf("dll version = %s\n", Version);
}
```

---

## 5.6. ECAT\_GetFirmwareVersion

### 說明:

取得當前使用的韌體版本資訊。

### 格式:

```
int32_t ECAT_GetFirmwareVersion(uint16_t DeviceNo, char *Version, uint16_t *Size);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號(卡號)
Version	char*	OUT	當前使用的韌體版本資訊
Size	uint16_t	OUT	回傳的 Version 大小 單位:Byte

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
char Version[255];
uint16_t size;
ret = ECAT_GetFirmwareVersion(DeviceNo ,Version, &size);
if(ret < 0)
{
    printf("Failed to get firmware version:%d\n", ret);
}
else
{
    printf("firmware version = %s\n", Version);
}
```

---

## 5.7. ECAT\_GetDeviceDI

### 說明:

讀取 ECAT-M801 卡本地端的輸入埠的資料。共 13 點，Sink/Source 可設定，請參考硬體手冊的 JP3 設定。

### 格式:

```
int32_t ECAT_GetDeviceDI(uint16_t DeviceNo, uint32_t *Value)
```

### 參數:

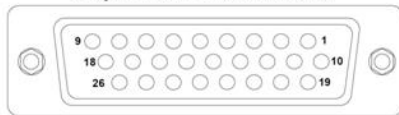
名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Value	uint32_t*	OUT	輸入埠資料 (DI0 ~ DI12)

### 回傳值:

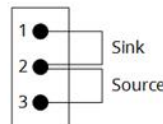
0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

26-pin Female D-sub Connector

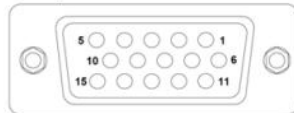


Pin Number	Pin Assignment	Pin Number	Pin Assignment	Pin Number	Pin Assignment
1	DI0	10	DO0	19	DI8
2	DI1	11	DO1	20	DI9
3	DI2	12	DO2	21	DI10
4	DI3	13	DO3	22	DI11
5	DI4	14	DO4	23	DO8
6	DI5	15	DO5	24	DO9
7	DI6	16	DO6	25	DO10
8	DI7	17	DO7	26	DO11
9	EXT. GND	18	EXT. PWR		



JP3

15-pin Female D-sub Connector



Pin Number	Pin Assignment	Pin Number	Pin Assignment	Pin Number	Pin Assignment
1	1A-	6	1A+	11	CMP 1+
2	1B-	7	1B+	12	CMP 1-
3	2A-	8	2A+	13	CMP 2+
4	2B-	9	2B+	14	CMP 2-
5	EXT GND	10	DI12	15	DO12



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
ret = ECAT_GetDeviceDI(DeviceNo, &Value);
if(ret < 0)
{
    printf("Failed to get device DI:%d\n", ret);
}
else
{
    printf("DI:%u! \n", Value);
}
```

---

## 5.8. ECAT\_GetDeviceDIBit

### 說明:

讀取 ECAT-M801 卡本地端的輸入埠指定位元的資料。請參考 *ECAT\_GetDeviceDI* 說明。

### 格式:

```
int32_t ECAT_GetDeviceDIBit(uint16_t DeviceNo, uint16_t BitNo, uint32_t *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
BitNo	uint16_t	IN	指定位元編號 (0 ~ 12)
Value	uint32_t*	OUT	該 DI 值 (0 或是 1)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
uint16_t BitNo = 2;
ret = ECAT_GetDeviceDIBit(DeviceNo, BitNo, &Value);
if(ret < 0)
{
    printf("Failed to get device DI:%d\n", ret);
}
else
{
    printf("DI_Bit[%u]:%u! \n", BitNo,Value);
}
```

---

## 5.9. ECAT\_GetDeviceDO

**說明:**

讀取 ECAT-M801 卡本地端的輸出埠的資料。共 13 點請參考硬體手冊。

**格式:**

```
int32_t ECAT_GetDeviceDO(uint16_t DeviceNo, uint32_t *Value)
```

**參數:**

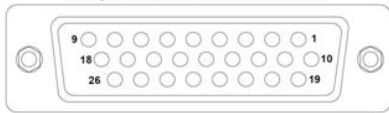
名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Value	uint32_t*	OUT	輸出埠資料 (DO0 ~DO12)

**回傳值:**

0: 函式執行成功。

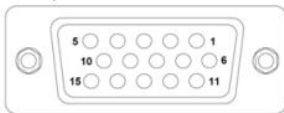
其他: 請參考附錄"函式錯誤回傳代碼"說明。

26-pin Female D-sub Connector



Pin Number	Pin Assignment	Pin Number	Pin Assignment	Pin Number	Pin Assignment
1	DI0	10	DO0	19	DI8
2	DI1	11	DO1	20	DI9
3	DI2	12	DO2	21	DI10
4	DI3	13	DO3	22	DI11
5	DI4	14	DO4	23	DO8
6	DI5	15	DO5	24	DO9
7	DI6	16	DO6	25	DO10
8	DI7	17	DO7	26	DO11
9	EXT. GND	18	EXT. PWR		

15-pin Female D-sub Connector



Pin Number	Pin Assignment	Pin Number	Pin Assignment	Pin Number	Pin Assignment
1	1A-	6	1A+	11	CMP 1+
2	1B-	7	1B+	12	CMP 1-
3	2A-	8	2A+	13	CMP 2+
4	2B-	9	2B+	14	CMP 2-
5	EXT GND	10	DI12	15	DO12

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
ret = ECAT_GetDeviceDO(DeviceNo,&Value);
if(ret < 0)
{
    printf("Failed to get device DO:%d\n",ret);
}
else
{
    printf("DO:%u! \n",Value);
}
```

---

## 5.10. ECAT\_GetDeviceDOBit

### 說明:

讀取 ECAT-M801 卡本地端的輸出埠指定位元的資料。請參考 *ECAT\_GetDeviceDO* 說明。

### 格式:

```
int32_t ECAT_GetDeviceDOBit(uint16_t DeviceNo, uint16_t BitNo, uint32_t *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
BitNo	uint16_t	IN	指定位元 (0 ~ 12)
Value	uint32_t*	OUT	輸出埠資料 (0 或 1)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
uint16_t BitNo = 2;
ret = ECAT_GetDeviceDOBit(DeviceNo, BitNo, &Value);
if(ret < 0)
{
    printf("Failed to get device DO:%d\n", ret);
}
else
{
    printf("DO_Bit[%u]:%u! \n", BitNo, Value);
}
```

---

## 5.11. ECAT\_SetDeviceDO

### 說明:

寫入資料到 ECAT-M801 卡本地端的輸出埠。請參考 *ECAT\_GetDeviceDO* 說明。

### 格式:

```
int32_t ECAT_SetDeviceDO(uint16_t DeviceNo, uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Value	uint32_t	IN	輸出埠寫入資料 (bit0 ~bit12 有效)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value = 0x000F;
ret = ECAT_SetDeviceDO(DeviceNo, Value);
if(ret < 0)
{
    printf("Failed to set device DO:%d\n", ret);
}
else
{
    printf("DO:%u! \n", Value);
}
```

## 5.12. ECAT\_SetDeviceDOBit

### 說明:

寫入資料到 ECAT-M801 卡本地端的指定位元輸出埠。請參考 *ECAT\_GetDeviceDO* 說明。

### 格式:

```
int32_t ECAT_SetDeviceDOBit(uint16_t DeviceNo, uint16_t BitNo, uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
BitNo	uint16_t	IN	指定位元 (0 ~ 12)
Value	uint32_t*	IN	輸出埠寫入資料 (0 或 1)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value = 1;
uint16_t BitNo = 2;
ret = ECAT_SetDeviceDOBit(DeviceNo, BitNo, Value);
if(ret < 0)
{
    printf("Failed to set device DO:%d\n", ret);
}
else
{
    printf("DO_Bit[%u]:%u! \n", BitNo, Value);
}
```

---

## 5.13. ECAT\_SetDeviceEncProperty

### 說明:

設定 ECAT-M801 卡本地端的編碼器介面模式。

### 格式:

int32\_t ECAT\_SetDeviceEncProperty(uint16\_t DeviceNo, uint16\_t EncNo, uint8\_t Mode, uint8\_t InvertCnt, uint8\_t LPF)

### 參數:

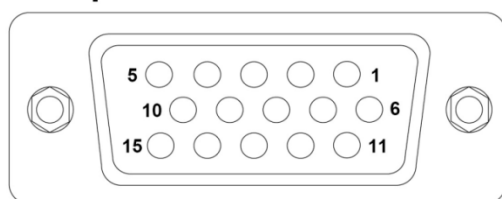
名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EncNo	uint16_t	IN	編碼器介面編號
Mode	uint8_t	IN	編碼器介面輸入模式 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	IN	反相計數設定 (0 或 1)
LPF	uint8_t	IN	低通濾波器設定(如表 5.1 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 15-pin Female D-sub Connector



Pin Number	Pin Assignment	Pin Number	Pin Assignment	Pin Number	Pin Assignment
1	1A-	6	1A+	11	CMP 1+
2	1B-	7	1B+	12	CMP 1-
3	2A-	8	2A+	13	CMP 2+
4	2B-	9	2B+	14	CMP 2-
5	EXT GND	10	DI12	15	DO12

表 5.1: 編碼器低通濾波器定義

標頭檔定義名稱	數值	說明
DEV_ENC_LPF_4_MHZ	0	4MHz
DEV_ENC_LPF_3P6_MHZ	1	3.6MHz
DEV_ENC_LPF_1P8_MHZ	2	1.8MHz
DEV_ENC_LPF_950_KHZ	4	950KHz
DEV_ENC_LPF_480_KHZ	8	480KHz
DEV_ENC_LPF_240_KHZ	16	240KHz
DEV_ENC_LPF_120_KHZ	32	120KHz
DEV_ENC_LPF_60_KHZ	64	60KHz
DEV_ENC_LPF_30_KHZ	128	30KHz

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
uint8_t Mode = 3; //A/B Phase
uint8_t ReverseCnt = 1; //Enable Reverse
uint8_t LPF = 0;
ret = ECAT_SetDeviceEncProperty(DeviceNo,EncNo,Mode,ReverseCnt,LPF);
if(ret != 0)
{
    printf("Failed to set encoder mode:%d\n",ret);
}
else
{
    printf("Set encoder mode successfully! \n");
}
```

---

## 5.14. ECAT\_GetDeviceEncProperty

### 說明:

取得 ECAT-M801 卡本地端的編碼器介面模式。請參考 *ECAT\_SetDeviceEncProperty* 說明。

### 格式:

```
int32_t ECAT_GetDeviceEncProperty(uint16_t DeviceNo, uint16_t EncNo, uint8_t *Mode, uint8_t *InvertCnt, uint8_t *LPF)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EncNo	uint16_t	IN	編碼器介面編號
Mode	uint8_t *	OUT	編碼器介面輸入模式
InvertCnt	uint8_t *	OUT	反相計數狀態
LPF	uint8_t *	OUT	低通濾波器設定(如表 5.1 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
uint8_t Mode;
uint8_t ReverseCnt ;
uint8_t LPF;
ret = ECAT_GetDeviceEncProperty(DeviceNo, EncNo, &Mode, &ReverseCnt, &LPF);
if(ret != 0)
{
    printf("Failed to get encoder mode:%d\n", ret);
}
else
{
    printf("Encoder mode:%u\n", Mode);
    printf("Encoder reverse:%u\n", ReverseCnt);
    printf("Encoder Low Pass Filter:%u\n", LPF);
}
```

---



## 5.15. ECAT\_GetDeviceEncCount

### 說明:

取得 ECAT-M801 卡本地端的編碼器計數器數值。

### 格式:

```
int32_t ECAT_GetDeviceEncCount(uint16_t DeviceNo, uint16_t EncNo, int32_t *Cnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EncNo	uint16_t	IN	編碼器介面編號 (0 或 1)
Cnt	int32_t *	OUT	編碼器計數器數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
int32_t Cnt;
ret = ECAT_GetDeviceEncCount(DeviceNo, EncNo, &Cnt);
if(ret != 0)
{
    printf("Failed to get encoder count:%d\n", ret);
}
else
{
    printf("Encoder count:%d\n", Cnt);
}
```

---

## 5.16. ECAT\_ResetDeviceEncCount

### 說明:

清除 ECAT-M801 卡本地端的編碼器計數器數值。

### 格式:

```
int32_t ECAT_ResetDeviceEncCount(uint16_t DeviceNo, uint16_t EncNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EncNo	uint16_t	IN	編碼器編號 (0 或 1)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
ret = ECAT_ResetDeviceEncCount(DeviceNo, EncNo);
if(ret != 0)
{
    printf("Failed to clear encoder count:%d\n", ret);
}
else
{
    printf("Clear encoder count successfully!\n");
}
```

---

## 5.17. ECAT\_SetDeviceCmpTrigProperty

### 說明:

設定 ECAT-M801 卡本地端的比較器觸發功能相關配置。

### 格式:

```
int32_t ECAT_SetDeviceCmpTrigProperty(uint16_t DeviceNo, uint16_t CmpNo,
uint32_t PulseWidth, uint8_t Source)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CmpNo	uint16_t	IN	比較器編號 (0 或 1)
PulseWidth	uint32_t	IN	觸發時，DO 輸出脈波寬度設定，每一個單位為 0.016us 最大值為 0x7ffffff x 0.016us
Source	uint8_t	IN	DO 輸出通道 0: 比較器 0 比較編碼器 0 輸出 CMP1 通道 比較器 1 比較編碼器 1 輸出 CMP2 通道 1: 比較器 0 比較編碼器 0 輸出 CMP2 通道 比較器 1 比較編碼器 1 輸出 CMP1 通道 2: 比較器 0 比較編碼器 0 輸出 CMP1 通道 比較器 1 比較編碼器 0 輸出 CMP2 通道 3: 比較器 0 比較編碼器 1 輸出 CMP1 通道 比較器 1 比較編碼器 1 輸出 CMP2 通道

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t CmpNo = 0;  
uint32_t PulseWidth = 100000;  
uint8_t Source = 0;  
ret = ECAT_SetDeviceCmpTrigProperty(DeviceNo, CmpNo, PulseWidth, Source);  
if(ret != 0)  
    printf("Failed to set compare trigger property:%d\n", ret);
```

---

## 5.18. ECAT\_GetDeviceCmpTrigProperty

### 說明:

取得 ECAT-M801 卡本地端的比較器觸發功能相關配置。

### 格式:

```
int32_t ECAT_GetDeviceCmpTrigProperty(uint16_t DeviceNo, uint16_t CmpNo,
uint32_t *PulseWidth, uint8_t *Source)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CmpNo	uint16_t	IN	比較器編號
PulseWidth	uint32_t *	OUT	觸發時，DO 輸出脈波寬度設定，每一個單位為 0.016us 最大值為 0x7ffffff x 0.016us
Source	uint8_t *	OUT	DO 輸出通道 0: 比較器 0 比較編碼器 0 輸出 CMP1 通道 比較器 1 比較編碼器 1 輸出 CMP2 通道 1: 比較器 0 比較編碼器 0 輸出 CMP2 通道 比較器 1 比較編碼器 1 輸出 CMP1 通道 2: 比較器 0 比較編碼器 0 輸出 CMP1 通道 比較器 1 比較編碼器 0 輸出 CMP2 通道 3: 比較器 0 比較編碼器 1 輸出 CMP1 通道 比較器 1 比較編碼器 1 輸出 CMP2 通道

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t CmpNo = 0;
uint32_t PulseWidth;
uint8_t Source;
ret = ECAT_GetDeviceCmpTrigProperty(DeviceNo, CmpNo, &PulseWidth, &Source);
if(ret != 0)
{
    printf("Failed to get compare trigger property:%d\n", ret);
}
else
{
    printf("Compare trigger pulse width:%u\n", PulseWidth);
    printf("Compare trigger source:%u\n", Source);
}
```

---



## 5.19. ECAT\_SetDeviceCmpTrigData

### 說明:

開始 ECAT-M801 卡本地端的比較器的單次比較觸發功能。

注:將單次比較數值設定為編碼器計數器數值時，會立刻觸發

### 格式:

```
int32_t ECAT_SetDeviceCmpTrigData(uint16_t DeviceNo, uint16_t CmpNo, int32_t CmpData)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CmpNo	uint16_t	IN	比較器編號 (0 或 1)
CmpData	int32_t	IN	單次比較數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint16_t CmpNo = 0;  
int32_t CmpData = 1000;  
ret = ECAT_SetDeviceCmpTrigData(DeviceNo, CmpNo, CmpData);  
if(ret != 0)  
    printf("Failed to set compare trigger data:%d\n", ret);
```

---

## 5.20. ECAT\_SetDeviceContCmpTrigData

### 說明:

開始 ECAT-M801 卡本地端的比較器的連續、多次比較觸發功能。

### 格式:

int32\_t ECAT\_SetDeviceContCmpTrigData(uint16\_t DeviceNo, uint16\_t CmpNo, int32\_t Start, uint32\_t Interval, uint32\_t Times, uint8\_t Dir)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CmpNo	uint16_t	IN	比較器編號 (0 或 1)
Start	int32_t	IN	開始比較的數值
Interval	uint32_t	IN	每次觸發的間隔數值
Times	uint32_t	IN	設定 0 為連續觸發，大於 0 為多次觸發次數
Dir	uint8_t	IN	比較方法 0: 向上計數時比較 1: 向下計數時比較

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t CmpNo = 0;
int32_t Start = 1000;
uint32_t Interval = 200;
uint32_t Times = 10;
ret = ECAT_SetDeviceContCmpTrigData(DeviceNo, CmpNo, Start, Interval, Times);
if(ret != 0)
    printf("Failed to set continuous compare trigger data:%d\n", ret);
```

---

## 5.21. ECAT\_SetDeviceCmpDisable

### 說明:

停止 ECAT-M801 卡本地端的比較器的比較觸發功能。

### 格式:

```
int32_t ECAT_SetDeviceCmpDisable(uint16_t DeviceNo, uint16_t CmpNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CmpNo	uint16_t	IN	比較器編號 (0 或 1)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint16_t CmpNo = 0;  
ret = ECAT_SetDeviceCmpDisable(DeviceNo, CmpNo);  
if(ret != 0)  
    printf("Failed to disable compare trigger:%d\n", ret);
```

---

## 5.22. ECAT\_SetDeviceEmg

### 說明:

設定控制卡卡的緊急停止信號相關配置。可以使用本地端 DI12 或是指定 EtherCAT 從站模組的 DI。

注: EMP-9000 系列控制卡無本地端 DI12，使用本地端 DI7

### 格式:

int32\_t ECAT\_SetDeviceEmg(uint16\_t DeviceNo, uint8\_t Source, uint8\_t Enable, uint8\_t Logic, uint16\_t SlaveNo, uint16\_t BitNo, uint8\_t ServoOff)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Source	uint8_t	IN	緊急停止信號來源設定(如表 5.2 所示)
Enable	uint8_t	IN	啟用/停用緊急停止功能 0: 停用 (內定值) 1: 啟用
Logic	uint8_t	IN	緊急停止信號邏輯準位 0: Low 1: High
SlaveNo	uint16_t	IN	從站編號 (使用從站 DI 時，有效)
BitNo	uint16_t	IN	位元編號 (使用從站 DI 時，有效)
ServoOff	uint8_t	IN	緊急停止時是否將軸 Servo Off 0: 否 1: 是

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 5.2: 緊急停止信號來源定義

標頭檔定義名稱	數值	說明
DEV_EMG_SOURCE_OB_DI	0	主卡 DI
DEV_EMG_SOURCE_SLAVE_DI	1	從站 DI

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Source, Logic, Enable, ServoOff;
uint16_t SlaveNo, BitNo;
/* On board DI settings*/
Source = DEV_EMG_SOURCE_OB_DI;
Logic = 0; // Low active
Enable = 1;
ServoOff = 0;
ret = ECAT_SetDeviceEmg(DeviceNo, Source, Enable, Logic, 0, 0, ServoOff);
if(ret != 0)
    printf("Failed to enable emergency:%d\n", ret);
/* Slave DI settings*/
Source = DEV_EMG_SOURCE_SLAVE_DI;
Logic = 0; // Low active
Enable = 1;
SlaveNo = 0;
BitNo = 1;
ret = ECAT_SetDeviceEmg(DeviceNo, Source, Enable, Logic, SlaveNo, BitNo, ServoOff);
if(ret != 0)
    printf("Failed to enable emergency:%d\n", ret);

```



## 5.23. ECAT\_GetDeviceEmg

### 說明:

取得控制卡的緊急停止信號相關配置資訊。可能是使用本地端 DI12 或是指定 EtherCAT 從站模組的 DI。

### 格式:

```
int32_t ECAT_GetDeviceEmg(uint16_t DeviceNo, uint8_t *Source, uint8_t *Enable,
uint8_t *Logic, uint16_t *SlaveNo, uint16_t *BitNo, uint8_t *ServoOff)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Source	uint8_t *	OUT	緊急停止信號來源設定(如表 5.2 所示)
Enable	uint8_t *	OUT	啟用/停用緊急停止功能 0: 停用 (內定值) 1: 啟用
Logic	uint8_t *	OUT	緊急停止信號邏輯準位 0: Low 1: High
SlaveNo	uint16_t *	OUT	從站編號 (使用從站 DI 時, 有效)
BitNo	uint16_t *	OUT	位元編號 (使用從站 DI 時, 有效)
ServoOff	uint8_t *	IN	緊急停止時是否將軸 Servo Off 0: 否 1: 是

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Source, Logic, Enable, ServoOff;
uint16_t SlaveNo, BitNo;

ret = ECAT_GetDeviceEmg(DeviceNo, &Source, &Enable, &Logic, &SlaveNo, &BitNo, &ServoOff);
if(ret != 0)
    printf("Failed to get emergency settings:%d\n", ret);
else{
    printf("Emergency source:%d\n", Source);
    printf("Emergency enable:%d\n", Enable);
    printf("Emergency logic:%d\n", Logic);
    printf("Emergency SlaveNo:%d\n", SlaveNo);
    printf("Emergency BitNo:%d\n", BitNo);
    printf("Emergency ServoOff:%d\n", ServoOff);
}
```

## 5.24. ECAT\_GetDeviceEmgStatus

### 說明:

取得裝置緊急停止信號的狀態。緊急停止信號可能是使用本地端 DI12 或是指定 EtherCAT 從站模組的 DI，要事先定義與啟用。

### 格式:

```
int32_t ECAT_GetDeviceEmgStatus(uint16_t DeviceNo, uint8_t *Status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Status	uint8_t *	OUT	緊急停止信號狀態

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Status;

ret = ECAT_GetDeviceEmgStatus(DeviceNo, &Status);
if(ret != 0)
    printf("Failed to get emergency status:%d\n", ret);
else
    printf("Emergency Status:%d\n", Status);
```

---

## 5.25. ECAT\_SetDeviceEmgSoftSig

### 說明:

觸發軟體緊急停止信號。這是由 PC 端下命令方式通知 ECAT-M801 卡做緊急停止動作。

### 格式:

int32\_t ECAT\_SetDeviceEmgSoftSig (uint16\_t DeviceNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Source, Logic, Enable, ServoOff;
uint16_t SlaveNo, BitNo;
/* On board DI settings*/
Source = DEV_EMG_SOURCE_OB_DI;
Logic = 0; // Low active
Enable = 1;
ServoOff = 0;
ret = ECAT_SetDeviceEmg(DeviceNo, Source, Enable, Logic, 0, 0, ServoOff);
if(ret != 0)
    printf("Failed to enable emergency:%d\n", ret);

ret = ECAT_SetDeviceEmgSoftSig (DeviceNo);
if(ret != 0)
    printf("Failed to set emergency software signal:%d\n", ret);
```

## 5.26. ECAT\_SetDeviceMPG

### 說明:

啟用/停用手搖輪功能。手搖輪介面使用 ECAT-M801 本地端的 DI 以及編碼器介面來定義，MPG 信號腳位接線如表 5.3、表 5.4 所示。

### 格式:

```
int32_t ECAT_SetDeviceMPG(uint16_t DeviceNo, uint8_t Enable, uint16_t *AxisNo,
uint16_t AxisCount)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Enable	uint8_t	IN	啟用/停用手搖輪功能 0: 停用 (內定值) 1: 啟用
AxisNo	uint16_t *	IN	初始化手搖輪軸號對應的實際軸號
AxisCount	uint16_t	IN	初始化軸號數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 5.3: CON1 接頭 MPG 信號腳位接線

Pin Number	Pin Assignment	MPG Signal	Pin Number	Pin Assignment	MPG Signal
1	DI0	X	8	DI7	x1
2	DI1	Y	19	DI8	x10
3	DI2	Z	20	DI9	x100
4	DI3	4	9	EXT. GND	0V
5	DI4	5	18	EXT. PWR	+24V
6	DI5	6			
7	DI6	7			

表 5.4: CON2 接頭 MPG 信號腳位接線

Pin Number	Pin Assignment	MPG Signal
1	1A-	$\bar{A}$
6	1A+	A
2	1B-	$\bar{B}$
7	1B+	B

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Enable = 1;
uint16_t AxisNo[4];
uint16_t AxisCount = 4;
AxisNo[0] = 0;
AxisNo[1] = 1;
AxisNo[2] = 2;
AxisNo[3] = 3;

```



---

```
ret = ECAT_SetDeviceMPG(DeviceNo, Enable, AxisNo, AxisCount);  
if (ret != 0)  
{  
    printf("Failed to set device MPG:%d\n", ret);  
}
```

---

## 5.27. ECAT\_GetDeviceMPG

### 說明:

取得手搖輪功能相關配置。手搖輪介面使用 ECAT-M801 本地端的 DI 以及編碼器介面來定義，MPG 信號腳位接線如表 5.3、表 5.4 所示。

### 格式:

```
int32_t ECAT_GetDeviceMPG(uint16_t DeviceNo, uint8_t *Enable, uint16_t *AxisNo,
uint16_t *AxisCount)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Enable	uint8_t	OUT	啟用/停用手搖輪功能 0: 停用 (內定值) 1: 啟用
AxisNo	uint16_t *	OUT	初始化手搖輪軸號對應的實際軸號
AxisCount	uint16_t	OUT	初始化軸號數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Enable;
uint16_t AxisNo[4];
uint16_t i, AxisCount;

ret = ECAT_GetDeviceMPG(DeviceNo, &Enable, AxisNo, &AxisCount);
if (ret != 0){
    printf("Failed to get device MPG:%d\n", ret);
}
else{
    printf("MPG enable:%d\n", Enable);
    for (i = 0; i < AxisCount; i++)
        printf("MPG axis number[%d]:%d\n", i, AxisNo[i]);
}
```

---

## 5.28. ECAT\_GetDeviceState

### 說明:

取得指定裝置編號 EtherCAT 網絡狀態。

### 格式:

```
int32_t ECAT_GetDeviceState(uint16_t DeviceNo, uint32_t *LinkUp, uint32_t *
SlavesResp, uint32_t *AIStates, uint32_t *Wc)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
LinkUp	uint32_t*	OUT	顯示乙太網路孔連接狀態 0: 主站無連接任何從站 1: 主站已連接任何從站
SlavesResp	uint32_t*	OUT	顯示目前已連接從站數量
AIStates	uint32_t*	OUT	顯示目前整個網絡 EtherCAT SubDevice 狀態 (如表 5.5 所示)
Wc	uint32_t*	OUT	顯示 EtherCAT 工作計數器數值 (Working Counter)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 5.5: EtherCAT AL 狀態

標頭檔定義名稱	數值	說明
ECAT_AS_INIT	0x00	Init 狀態
ECAT_AS_PREOP	0x02	Pre-Operational 狀態
ECAT_AS_SAFEOP	0x04	Safe-Operational 狀態
ECAT_AS_OP	0x08	Operational 狀態

## 使用範例:

## [C/C++]

```

int32_t ret;
char buffer[1024];
char StrAlState[255];
uint16_t DeviceNo = 0;
uint32_t LinkUp, SlavesResp, AlStates, Wc;
ret = ECAT_GetDeviceState(DeviceNo, &LinkUp, &SlavesResp, &AlStates, &Wc);
if(ret < 0)
{
    printf("Failed to get device state:%d\n", ret);
}
else
{
    if(AlStates == ECAT_AS_INIT)
        sprintf(StrAlState, "INIT");
    else if(AlStates == ECAT_AS_PREOP)
        sprintf(StrAlState, "PREOP");
    else if(AlStates == ECAT_AS_SAFEOP)
        sprintf(StrAlState, "SAFEOP");
    else if(AlStates == ECAT_AS_OP)
        sprintf(StrAlState, "OP");
    else
        sprintf(StrAlState, "Invalid");
    sprintf(buffer, "Slave(s):%u | AL States:%s | Link is :%s | Wc:%-u "
                ,SlavesResp, StrAlState, LinkUp? "up" : "down", Wc);
    printf("%s\n", buffer);
}

```

---

}

---

## 5.29. ECAT\_GetDeviceStateEx

### 說明:

取得指定裝置編號 EtherCAT 網絡狀態。

AIStates 在 ECAT\_AS\_OP 時，若 EtherCAT 通訊有異常，Wc 會有變化，此時 AIStates 可能還是會維持在 ECAT\_AS\_OP，此函式與 ECAT\_GetDeviceState 不同的是，當 Wc 有異常時，會將 AIStates 的 bit4 修改為 1

### 格式:

```
int32_t ECAT_GetDeviceStateEx(uint16_t DeviceNo, uint32_t *LinkUp, uint32_t *
SlavesResp, uint32_t *AIStates, uint32_t *Wc)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
LinkUp	uint32_t*	OUT	顯示乙太網路孔連接狀態 0: 主站無連接任何從站 1: 主站已連接任何從站
SlavesResp	uint32_t*	OUT	顯示目前已連接從站數量
AIStates	uint32_t*	OUT	顯示目前整個網絡 EtherCAT SubDevice 狀態 (如表 5.5 所示)
Wc	uint32_t*	OUT	顯示 EtherCAT 工作計數器數值 (Working Counter)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 5.6: EtherCAT AL 狀態

標頭檔定義名稱	數值	說明
ECAT_AS_INIT	0x00	Init 狀態
ECAT_AS_PREOP	0x02	Pre-Operational 狀態
ECAT_AS_SAFEOP	0x04	Safe-Operational 狀態
ECAT_AS_OP	0x08	Operational 狀態

## 使用範例:

## [C/C++]

```

int32_t ret;
char buffer[1024];
char StrAlState[255];
uint16_t DeviceNo = 0;
uint32_t LinkUp, SlavesResp, AlStates, Wc;
ret = ECAT_GetDeviceStateEx(DeviceNo, &LinkUp, &SlavesResp, &AlStates, &Wc);
if(ret < 0)
{
    printf("Failed to get device state:%d\n", ret);
}
else
{
    if(AlStates == ECAT_AS_INIT)
        sprintf(StrAlState, "INIT");
    else if(AlStates == ECAT_AS_PREOP)
        sprintf(StrAlState, "PREOP");
    else if(AlStates == ECAT_AS_SAFEOP)
        sprintf(StrAlState, "SAFEOP");
    else if(AlStates == ECAT_AS_OP)
        sprintf(StrAlState, "OP");
    else
        sprintf(StrAlState, "Invalid");
    sprintf(buffer, "Slave(s):%u | AL States:%s | Link is :%s | Wc:%-u "
                ,SlavesResp, StrAlState, LinkUp? "up" : "down", Wc);
    printf("%s\n", buffer);
}

```



---

}

---

## 5.30. ECAT\_StartDeviceOpTask

### 說明:

開始指定裝置編號 EtherCAT 操作任務。

### 格式:

```
int32_t ECAT_StartDeviceOpTask(uint16_t DeviceNo, uint16_t NetworkInfoNo,
uint8_t EnumCycleTime, uint32_t WcErrCnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
NetworkInfoNo	uint16_t	IN	網絡架構編號 (由 Utility 建立並寫入的網路資訊檔案編號)
EnumCycleTime	uint8_t	IN	EtherCAT 通訊週期編號 (如表 5.7 所示)
WcErrCnt	uint32_t	IN	設定裝置網絡狀態出錯時，裝置可容許連續出錯次數，當發生次數到達設定值時才發出錯誤告知使用者端。

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 5.7: 從站通訊週期編號

標頭檔定義名稱	數值	說明
DEV_OP_CYCLE_TIME_500US	252	通訊週期 0.5ms
DEV_OP_CYCLE_TIME_1MS	0	通訊週期 1ms
DEV_OP_CYCLE_TIME_2MS	1	通訊週期 2ms
DEV_OP_CYCLE_TIME_3MS	2	通訊週期 3ms
DEV_OP_CYCLE_TIME_4MS	3	通訊週期 4ms
DEV_OP_CYCLE_TIME_5MS	4	通訊週期 5ms
DEV_OP_CYCLE_TIME_6MS	5	通訊週期 6ms
DEV_OP_CYCLE_TIME_7MS	6	通訊週期 7ms
DEV_OP_CYCLE_TIME_8MS	7	通訊週期 8ms
DEV_OP_CYCLE_TIME_9MS	8	通訊週期 9ms
DEV_OP_CYCLE_TIME_10MS	9	通訊週期 10ms
DEV_OP_CYCLE_TIME_11MS	10	通訊週期 11ms
DEV_OP_CYCLE_TIME_12MS	11	通訊週期 12ms
DEV_OP_CYCLE_TIME_13MS	12	通訊週期 13ms
DEV_OP_CYCLE_TIME_14MS	13	通訊週期 14ms
DEV_OP_CYCLE_TIME_15MS	14	通訊週期 15ms
DEV_OP_CYCLE_TIME_16MS	15	通訊週期 16ms

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t NetworkInfoNo = 0;
uint8_t EnumCycleTime = DEV_OP_CYCLE_TIME_1MS;
uint32_t WcErrCnt = 3;
ret = ECAT_StartDeviceOpTask(DeviceNo, NetworkInfoNo, EnumCycleTime, WcErrCnt);
if(ret < 0)
{
    printf("Failed to start device op task:%d\n", ret);
}

```

---

```
}  
else  
{  
    printf("Start device op task successfully! \n");  
}
```

---

## 5.31. ECAT\_StopDeviceOpTask

### 說明:

停止指定裝置編號 EtherCAT 操作任務。

### 格式:

```
int32_t ECAT_StopDeviceOpTask(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	指定裝置編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t NetworkInfoNo = 0;
ret = ECAT_StopDeviceOpTask(DeviceNo);
if(ret < 0)
{
    printf("Failed to stop device op task:%d\n",ret);
}
else
{
    printf("stop device op task successfully! \n");
}
```

---

## 5.32. ECAT\_SetTimer

### 說明:

設定一個定時器。當控制卡進入 OP 狀態後內部會有一個定時工作的通訊週期時間，此週期時間由函式 *ECAT\_StartDeviceOpTask* 所定義。所以此函式與其它相關函式是利用此特性來產生的，可以給 PC 端一個可以使用的計時函式。本函式只有啟動 Timer 固定時間送出一個 event，使用者如果想要利用這定時產生的 event，則必須使用 *ECAT\_WaitforTimer* 函式來等 event。*ECAT\_WaitforTimer* 函式是一個 blocking 函式，內部等待直到 event 來才離開。如果時間未到，呼叫 *ECAT\_WaitforTimer* 函式的程序會休息釋放 CPU 控制權。

相關函式還有 *ECAT\_SetTimerStop* 函式，它會先送出一個 event，然後停止系統定時送出 event 的動作。

### 格式:

```
int32_t ECAT_SetTimer (uint16_t DeviceNo, uint32_t Interval)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	指定裝置編號
Interval	uint32_t	IN	時間間隔，單位為通訊週期

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Interval = 100;

ret = ECAT_SetTimer(DeviceNo, Interval);
if(ret < 0)
{
    printf("Failed to Set Timer:%d\n",ret);
}
else
{
    printf("Set Timer successfully! \n");
}
while(1)
{
    ret = ECAT_WaitforTimer(DeviceNo);
    if (ret == 0)
    {
        //do something ...
    }
    else
    {
        // do something ...
        break;
    }
}
```



## 5.33. ECAT\_SetTimerStop

### 說明:

解除定時器。請參考函式 *ECAT\_SetTimer* 說明。

### 格式:

```
int32_t ECAT_SetTimerStop(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	指定裝置編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_SetTimerStop(DeviceNo);
if(ret < 0)
{
    printf("Failed to Set Timer Stop:%d\n", ret);
}
else
{
    printf("Set Timer Stop successfully! \n");
}
```

---

## 5.34. ECAT\_WaitforTimer

### 說明:

等待定時器直到到達設定時間。請參考函式 *ECAT\_SetTimer* 說明。如果此 timer 已經使用 *ECAT\_SetTimerStop* 函式終結了，則再執行就會發生錯誤。所以檢查是否回傳錯誤可以判斷是否要終結該迴路，請參考範例的用法。

### 格式:

```
int32_t ECAT_WaitforTimer(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	指定裝置編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Interval = 100;

ret = ECAT_SetTimer(DeviceNo, Interval);
if(ret < 0)
{
    printf("Failed to Set Timer:%d\n",ret);
}
else
{
    printf("Set Timer successfully! \n");
}
while(1)
{
    ret = ECAT_WaitforTimer(DeviceNo);
    if (ret == 0)
    {
        //do something ...
    }
    else
    {
        // do something ...
        break;
    }
}
```

## 5.35. ECAT\_GetProcessTime

### 說明:

讀取 EtherCAT 通訊所花費的時間。注:1000 次花費時間的平均，單位為 micro second。

注意:根據從站數量的多寡，以及各種函式的使用，都有可能增加或減少 EtherCAT 通訊所花費的時間，請保持此數值在通訊週期(EnumCycleTime ,[ECAT\\_StartDeviceOpTask](#) 中設定)的 50%以下。

### 格式:

```
int32_t ECAT_GetProcessTime(uint16_t DeviceNo, double *Time)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Time	double*	OUT	EcatCAT 通訊所花費的時間

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
double Time;
ret = ECAT_GetProcessTime(DeviceNo,&Time);
if(ret < 0)
{
    printf("Failed to get Process Time:%d\n",ret);
}
else
{
    printf("Process Time:%f \n", Time);
}
```

---

## 5.36. ECAT\_SetHeartBeat

### 說明:

設定心跳偵測數值。

在進入 OP 且執行 [ECAT\\_McInIt](#) 後，若超過設定的心跳時間未執行任何命令，將會觸發軟體緊急停止信號

注：若啟用此功能，且 PC 端程式在進行斷點除錯或單步執行時，在心跳時間內未執行任何命令，將會觸發軟體緊急停止信號

### 格式:

```
int32_t ECAT_SetHeartBeat(uint16_t DeviceNo, uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	指定裝置編號
Value	uint32_t	IN	心跳時間，單位為通訊週期

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value= 1000;

ret = ECAT_SetHeartBeat(DeviceNo, Value);
if(ret < 0)
{
    printf("Failed to Set heartbeat:%d\n",ret);
}
else
{
    printf("Set heartbeat successfully! \n");
}
```

---



## 5.37. ECAT\_SetHeartBeatStatus

### 說明:

設定心跳偵測狀態。

在進入 OP 且執行 [ECAT\\_McInIt](#) 後，若超過設定的心跳時間未執行任何命令，將會觸發軟體緊急停止信號

注：若啟用此功能，且 PC 端程式在進行斷點除錯或單步執行時，在心跳時間內未執行任何命令，將會觸發軟體緊急停止信號

### 格式:

```
int32_t ECAT_SetHeartBeatStatus(uint16_t DeviceNo, uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	指定裝置編號
Value	uint32_t	IN	狀態 1:啟用 0:禁用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value= 1000;

ret = ECAT_SetHeartBeat(DeviceNo, Value);
if(ret < 0)
    printf("Failed to Set heartbeat:%d\n",ret);
else
    printf("Set heartbeat successfully! \n");

ret = ECAT_SetHeartBeatStatus(DeviceNo, 1); //Enable
if(ret < 0)
{
    printf("Failed to Set heartbeat status:%d\n",ret);
}
else
{
    printf("Set heartbeat status successfully! \n");
}
```

---

## 5.38. ECAT\_SetDeviceIgnoreWC

### 說明:

啟用/停用忽略 Working Counter 檢查功能。

在進入 OP 後，主站與從站會開始進行週期通訊，若之間的通訊有失敗的情形，Working Counter 就會有缺失，此時 DIO 會無法操控，運動軸會停止當前運動，並將狀態切為 MC\_AS\_ERRORSTOP，Last error 為-1004。

當狀態為 OP 且移除/接入其他模組時，會有一小段時間(幾個ms)會有通訊有失敗的情形。此功能主要是在有需要移除/接入其他模組時，先將忽略功能啟用，再將模組移除/接入，此時就不會因為通訊失敗造成軸狀態切為 MC\_AS\_ERRORSTOP。

### 格式:

```
int32_t ECAT_SetDeviceIgnoreWC(uint16_t DeviceNo, int8_t Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Enable	int8_t	IN	啟用/停用忽略 Working Counter 檢查功能 0: 停用 (內定值) 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
int8_t Enable = 1;
```

```
ret = ECAT_SetDeviceIgnoreWC(DeviceNo, Enable);
```

```
if (ret != 0)
```

```
{
```

```
    printf("Failed to set device Ignore wc:%d\n", ret);
```

```
}
```

---

## 5.39. ECAT\_GetDeviceIgnoreWC

### 說明:

取得啟用/停用忽略 Working Counter 檢查功能。

### 格式:

```
int32_t ECAT_GetDeviceIgnoreWC(uint16_t DeviceNo, int8_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Enable	int8_t	OUT	啟用/停用忽略 Working Counter 檢查功能 0: 停用 (內定值) 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
int8_t Enable;

ret = ECAT_GetDeviceIgnoreWC(DeviceNo, &Enable);
if (ret != 0){
    printf("Failed to get ignore wc:%d\n", ret);
}
else{
    printf("ignore wc enable:%d\n", Enable);
}
```

---

## 5.40. ECAT\_SetCheckSlaveCnt

### 說明:

啟用/停用進入 OP 時網絡架構檢查。

### 格式:

int32\_t ECAT\_SetCheckSlaveCnt(uint16\_t DeviceNo, int32\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Enable	int8_t	IN	啟用/停用網絡架構檢查 0: 停用 1: 啟用(預設)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
int8_t Enable = 0;
```

```
ret = ECAT_SetCheckSlaveCnt(DeviceNo, Enable);
```

```
if (ret != 0)
```

```
{
```

```
    printf("Failed to set device slave count check:%d\n", ret);
```

```
}
```

---



## 6. 從站操作函式集說明

### 6.1. ECAT\_SetSlaveNoType

**說明:**

定義從站編號，  
 當從站編號類型為 SLAVE\_NO\_TYPE\_POSITION 時，從站編號為模組的位置；  
 當從站編號類型為 SLAVE\_NO\_TYPE\_ALIAS 時，從站編號為模組別名(Alias)。

以圖 6.1 為例:

當從站編號類型為 SLAVE\_NO\_TYPE\_POSITION 時，從站編號“1”指的是 ECAT-2028  
 當從站編號類型為 SLAVE\_NO\_TYPE\_ALIAS 時，從站編號“1”指的是 ECAT-2011H

**注:**

- (1) 模組的位置指的是模組在 EtherCAT 網路架構中的位置(MainDevice-Module 0-Module 1...)
- (2) 模組別名(Alias)，不受模組連接順序影響，可自行設定，詳情請見 3.1.2. 網路架構編輯步驟。

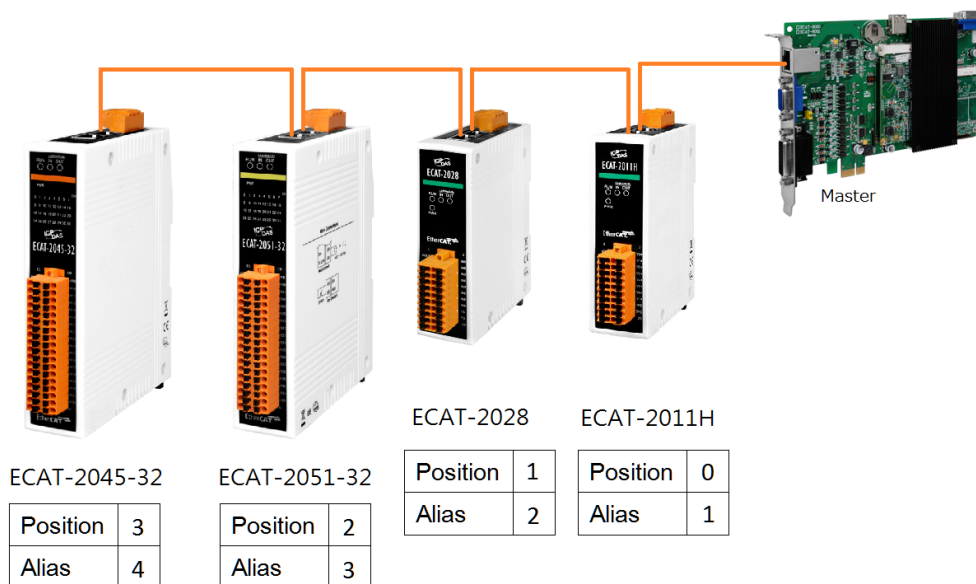


圖 6.1

**格式:**

int32\_t ECAT\_SetSlaveNoType(uint16\_t DeviceNo, uint16\_t Type)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Type	uint16_t	IN	從站編號類型，預設為 SLAVE_NO_TYPE_POSITION

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.1: 從站編號種類編號

標頭檔定義名稱	數值	說明
SLAVE_NO_TYPE_POSITION	0	位置型
SLAVE_NO_TYPE_ALIAS	1	別名型

## 使用範例:

## [C/C++]

```
int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t Type = SLAVE_NO_TYPE_ALIAS;
ret = ECAT_SetSlaveNoType(DeviceNo, Type);
if(ret < 0)
{
    printf("Failed to set slaveno type:%d\n", ret);
}
else
{
    printf("Set slaveno type successfully!\n");
}
```

## 6.2. ECAT\_GetSlaveNoType

**說明:**

取得從站編號定義，  
 當從站編號類型為 SLAVE\_NO\_TYPE\_POSITION 時，從站編號為模組的位置；  
 當從站編號類型為 SLAVE\_NO\_TYPE\_ALIAS 時，從站編號為模組別名(Alias)。

以圖 6.1 為例:

當從站編號類型為 SLAVE\_NO\_TYPE\_POSITION 時，從站編號“1”指的是 ECAT-2028  
 當從站編號類型為 SLAVE\_NO\_TYPE\_ALIAS 時，從站編號“1”指的是 ECAT-2011H

**注:**

- (1) 模組的位置指的是模組在 EtherCAT 網路架構中的位置(MainDevice-Module 0-Module 1...)
- (2) 模組別名(Alias)，不受模組連接順序影響，可自行設定，詳情請見 3.1.2. 網路架構編輯步驟。

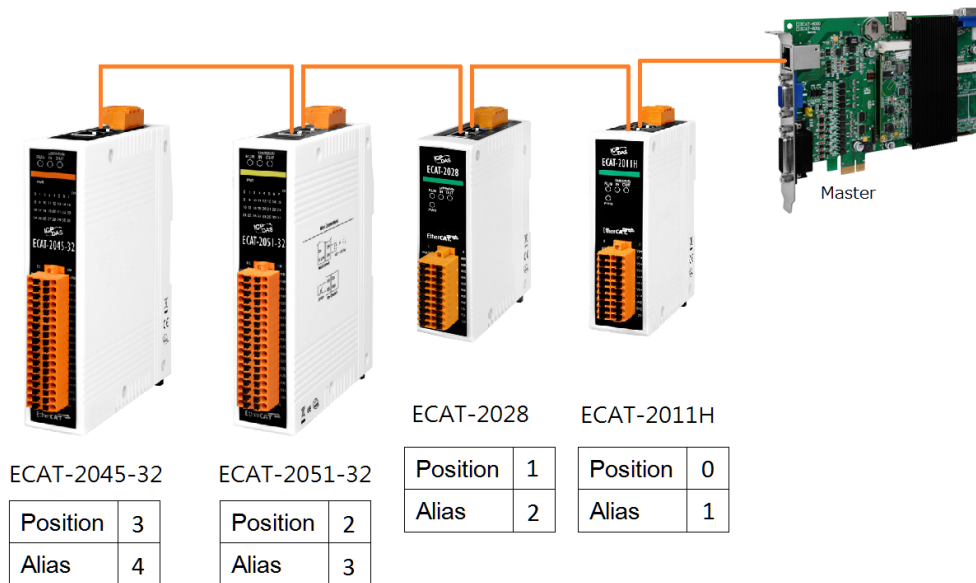


圖 6.2

**格式:**

```
int32_t ECAT_GetSlaveNoType(uint16_t DeviceNo, uint16_t *Type)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Type	uint16_t*	OUT	從站編號類型

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.2: 從站編號種類編號

標頭檔定義名稱	數值	說明
SLAVE_NO_TYPE_POSITION	0	位置型
SLAVE_NO_TYPE_ALIAS	1	別名型

## 使用範例:

## [C/C++]

```
int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t Type;
ret = ECAT_GetSlaveNoType(DeviceNo, &Type);
if(ret < 0)
{
    printf("Failed to set slaveno type:%d\n", ret);
}
else
{
    printf("Get slaveno type successfully!\n");
}
```

## 6.3. ECAT\_GetSlaveInfo

### 說明:

取得從站模組相關資訊。

當 Slavename(從站名稱)為空白時，代表在 PreOP 時讀不到名稱，OP 後會讀出從 ESI 獲取的名稱。(V1.0.18 或以上)

### 格式:

```
int32_t ECAT_GetSlaveInfo(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t *Alias,
uint32_t *ProductCode, uint32_t *VendorID, uint32_t *RevisionNo, uint32_t *SerialNo,
uint8_t *AIState, uint32_t *SlaveType, char *Slavename)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Alias	uint16_t*	OUT	從站 Alias 數值
ProductCode	uint32_t*	OUT	從站 Product Code 數值
VendorID	uint32_t*	OUT	從站 Vendor ID 數值
RevisionNo	uint32_t*	OUT	從站 Revision No 數值
SerialNo	uint32_t*	OUT	從站 Serial No 數值
AIState	uint8_t*	OUT	從站 EtherCAT 狀態
SlaveType	uint32_t*	OUT	從站種類(如表 6.3 所示)，若尚未開始 EtherCAT 操作任務時此值永遠為 0
Slavename	char*	OUT	從站名稱

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.3: 從站種類編號

標頭檔定義名稱	數值	說明
SLAVE_TYPE_GENERIC	0	通用種類
SLAVE_TYPE_CiA402	1	CiA 402 種類
SLAVE_TYPE_STEPPER_MOTOR	2	單軸步進馬達種類
SLAVE_TYPE_4_AXIS_STEPPER_MOTOR	3	4 軸步進馬達種類

## 使用範例:

## [C/C++]

```

int32_t ret;
int16_t i;
uint16_t SlaveCnt;
uint16_t DeviceNo = 0;
uint16_t Alias;
uint32_t ProductCode, VendorID, RevisionNo, SerialNo, SlaveType;
uint8_t AIState;
ret = ECAT_OpenDevice(DeviceNo);
char Slavename[MAX_SLAVE_NAME_LENGTH];
if(ret < 0)
{
    printf("Failed to open device:%d\n",ret);
}
else
{
    for(i=0;i<SlaveCnt;i++)
    {
        ret = ECAT_GetSlaveInfo(DeviceNo, i, &Alias, &ProductCode,
            &VendorID, &RevisionNo, &SerialNo, &AIState, &SlaveType, Slavename);
        if(ret < 0)
        {
            printf("Failed to get slave information:%d\n",ret);
        }
        else
        {

```



---

```
printf("Slave(%u)-+\n"  
      "|-ProductCode:0x%X\n"  
      "|-VendorID:0x%X\n"  
      "|-RevisionNo:0x%X\n"  
      "|-SerialNo:0x%X\n"  
      "|-SlaveType:%d\n"  
      "\n"  
      , i, ProductCode, VendorID, RevisionNo, SerialNo, SlaveType);  
}  
}  
}
```

---

## 6.4. ECAT\_GetSlaveSdoObject

### 說明:

取得從站模組 SDO 資料 (或是說, 使用 SDO 通訊取得指定物件資料)。

### 格式:

```
int32_t ECAT_GetSlaveSdoObject(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Index,
uint8_t SubIndex, uint16_t DataSize, uint32_t *ObjectVal, uint32_t *AbortCode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Index	uint16_t	IN	索引值
SubIndex	uint8_t	IN	子索引值
DataSize	uint16_t	IN	SDO 資料大小
ObjectVal	uint32_t*	OUT	取得的 SDO 資料
AbortCode	uint32_t*	OUT	SDO 操作錯誤終止代碼(請參考附錄 "SDO 操作錯誤終止代碼"說明)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t i;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint16_t Index = 0x607C; //Home Offset Index
uint8_t SubIndex = 0x00; //Home Offset SubIndex
uint16_t DataSize = 4; //4 byte
uint32_t ObjectVal = 0;
ret = ECAT_GetSlaveSdoObject(DeviceNo, SlaveNo, Index, SubIndex,
    DataSize, &ObjectVal, &AbortCode);
if(ret < 0)
{
    printf("Failed to get sdo object:%d\n", ret);
}
else
{
    printf("Get sdo object successfully!\n");
}
```

## 6.5. ECAT\_SetSlaveSdoObject

### 說明:

設定從站模組 SDO 資料 (或是說, 使用 SDO 通訊設定指定物件資料)。

### 格式:

```
int32_t ECAT_SetSlaveSdoObject(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Index,
uint8_t SubIndex, uint16_t DataSize, uint32_t ObjectVal, uint32_t *AbortCode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Index	uint16_t	IN	索引值
SubIndex	uint8_t	IN	子索引值
DataSize	uint16_t	IN	SDO 資料大小
ObjectVal	uint32_t	IN	設定的 SDO 資料
AbortCode	uint32_t*	OUT	SDO 操作錯誤終止代碼(請參考附錄 "SDO 操作錯誤終止代碼"說明)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t i;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint16_t Index = 0x607C; //Home Offset Index
uint8_t SubIndex = 0x00; //Home Offset SubIndex
uint16_t DataSize = 4; //4 byte
uint32_t ObjectVal = 100;
ret = ECAT_SetSlaveSdoObject(DeviceNo, SlaveNo, Index, SubIndex,
    DataSize, ObjectVal, &AbortCode);
if(ret < 0)
{
    printf("Failed to set sdo object:%d\n",ret);
}
else
{
    printf("set sdo object successfully!\n");
}
```

## 6.6. ECAT\_SetSlaveRxPdoData\_Ex

### 說明:

取代舊的函式"ECAT\_SetSlaveRxPdoData"

設定從站模組 RxPdo 資料 (或是說, 使用 PDO 通訊設定指定物件資料)。比如說, 設定 DO 模組的 DO 輸出資料, 該資料一般就是 RxPdo 資料。AO 模組也是類似的用法。

### 格式:

```
int32_t ECAT_SetSlaveRxPdoData_Ex(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint8_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 RxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t	IN	設定的資料大小 (最大可設定資料定義於標頭檔名稱 RW_PDO_DATA_SIZE_MAX)
Data	uint8_t*	IN	設定的位元組資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055-32
OffsetByte = 0;
DataSize = 2; //2 bytes
Data[0] = 0xFF;
Data[1] = 0xAA;

ret = ECAT_SetSlaveRxPdoData_Ex(DeviceNo, SlaveNo, OffsetByte, DataSize, Data);
if(ret < 0)
{
    printf("Failed to set RxPdo data:%d\n", ret);
}
else
{
    printf("Set RxPdo data successfully!\n");
}

```

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055

typedef struct ecat_2055_t // 8DI 8DO
{
    uint8_t DI;

```

```
uint8_t DO;
}ecat_2055_st;

ecat_2055_st E2055;
OffsetByte = 0;
DataSize = 1; //1 bytes
E2055.DO = 0xFF;

ret = ECAT_SetSlaveRxPdoData_Ex(DeviceNo, SlaveNo, OffsetByte, DataSize, (uint8_t *)& E2055.DO);
if(ret < 0)
{
    printf("Failed to set RxPdo data:%d\n", ret);
}
else
{
    printf("Set RxPdo data successfully!\n");
}
```

---



## 6.7. ECAT\_GetSlaveRxPdoData\_Ex

### 說明:

取代舊的函式"ECAT\_GetSlaveRxPdoData"

取得從站模組 RxPdo 資料 (或是說, 使用 PDO 通訊取得指定物件資料)。比如說, 讀取 DO 模組的 DO 輸出資料, 該資料一般就是 RxPdo 資料。AO 模組也是類似的用法。

### 格式:

```
int32_t ECAT_GetSlaveRxPdoData_Ex(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint8_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 RxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t	IN	取得的資料大小 (最大可設定資料定義於標頭檔名稱 RW_PDO_DATA_SIZE_MAX)
Data	uint8_t*	OUT	取得的位元組資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t i;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055-32
OffsetByte = 0;
DataSize = 2; //2 bytes

ret = ECAT_GetSlaveRxPdoData_Ex(DeviceNo, SlaveNo, OffsetByte, DataSize, Data);
if(ret < 0)
{
    printf("Failed to get RxPdo data:%d\n",ret);
}
else
{
    for(i=0;i<DataSize;i++)
    {
        printf("Data[%u]:0x%X\n", i, Data[i]);
    }
}

```

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t i;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2024

```

---

```
typedef struct ecat_2024_t
{
    unsigned int Output      :16;
    unsigned int Gap        :16;
}ecat_2024_st;
ecat_2024_st E2024;

OffsetByte = 0; //VOUT 0
// OffsetByte = sizeof(E2024) * 1; VOUT 1
DataSize = sizeof(E2024);

ret = ECAT_GetSlaveRxPdoData_Ex(DeviceNo, SlaveNo, OffsetByte, DataSize, (uint8_t *)&E2024);
if(ret < 0)
{
    printf("Failed to get RxPdo data:%d\n",ret);
}
else
{
    printf("AO Data: %u  \n", E2024.Output);
}
```

---

## 6.8. ECAT\_GetSlaveTxPdoData\_Ex

### 說明:

取代舊的函式"ECAT\_GetSlaveTxPdoData"

取得從站模組 TxPdo 資料 (或是說, 使用 PDO 通訊取得指定物件資料)。比如說, 讀取 DI 模組的 DI 輸入資料, 該資料一般就是 TxPdo 資料。AI 模組也是類似的用法。

### 格式:

```
int32_t ECAT_GetSlaveTxPdoData_Ex(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint8_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t	IN	取得的資料大小 (最大可設定資料定義於標頭檔名稱 RW_PDO_DATA_SIZE_MAX)
Data	uint8_t*	OUT	取得的位元組資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t i;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055-32
OffsetByte = 0;
DataSize = 2; //2 bytes

ret = ECAT_GetSlaveTxPdoData_Ex(DeviceNo, SlaveNo, OffsetByte, DataSize, Data);
if(ret < 0)
{
    printf("Failed to get TxPdo data:%d\n", ret);
}
else
{
    for(i=0;i<DataSize;i++)
    {
        printf("Data[%u]:0x%X\n", i, Data[i]);
    }
}

```

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t i;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2011-H

```

---

```
typedef struct ecat_2011h_t
{
    unsigned int Underrange      : 1;
    unsigned int Overrange      : 1;
    unsigned int Limit1         : 2;
    unsigned int Limit2         : 2;
    unsigned int Error          : 1;
    unsigned int Gap1           : 1;
    unsigned int Gap2           : 6;
    unsigned int TxPDO_State    : 1;
    unsigned int TxPDO_Toggle   : 1;
    unsigned int Value          : 16;
}ecat_2011h_st;
ecat_2011h_st E2011H;

OffsetByte = 0; // V0
// OffsetByte = sizeof(E2011H) * 1; V 1
DataSize = sizeof( E2011H );

ret = ECAT_GetSlaveTxPdoData_Ex(DeviceNo, SlaveNo, OffsetByte, DataSize, (uint8_t *)&E2011H);
if(ret < 0)
{
    printf("Failed to get TxPdo data:%d\n", ret);
}
else
{
    printf("AI Data: %u \n", E2011H.Value);
}
```

---

## 6.9. ECAT\_SetSlaveDIMap

### 說明:

啟用\_Directly 相關函式，請在 ECAT\_StartDeviceOpTask 前使用此函式

使用 ECAT\_StopDeviceOpTask 與 ECAT\_OpenDevice 時會自動禁用此功能

\_Directly 相關函式可以減少一般函式所花費的時間，但是有模組數上的限制，需要使用此函式設定模組，沒有被設定的模組無法使用\_Directly 相關函式

### 格式:

```
int32_t ECAT_SetSlaveDIMap(uint16_t DeviceNo, uint16_t ChannelCnt, uint16_t
*SlaveNo, uint16_t *Dir)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelCnt	uint16_t	IN	要設定的通道數量
SlaveNo	uint16_t []	IN	從站編號
Dir	uint32_t []	IN	0: 輸入埠資料 1: 輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [SLAVE_DIO_MAP_MAX];
uint32_t Dir[SLAVE_DIO_MAP_MAX];
uint16_t ChannelCnt;

/*
MainDevice
|- ECAT-2057 DO module
|- ECAT-2051 DI module
L ECAT-2055 DI& DO module
*/
// ECAT-2057 DO
SlaveNo[0] = 0;
Dir[0] = 1;

// ECAT-2051 DI
SlaveNo[1] = 1;
Dir[1] = 0;

// ECAT-2055 DI&DO
SlaveNo[2] = 2;
Dir[2] = 0;//DI

SlaveNo[3] = 2;
Dir[3] = 1;//DO

ChannelCnt= 4;

ret = ECAT_SetSlaveDIOMap(DeviceNo, ChannelCnt, SlaveNo, Dir);
if(ret < 0)
    printf("Failed to set slave DIO Map:%d\n", ret);
else
    printf("Set slave DIO Map successfully! \n");

```





## 6.10. ECAT\_SetSlaveDIMap\_16bit

### 說明:

啟用\_Directly 相關函式，請在 ECAT\_StartDeviceOpTask 前使用此函式

使用 ECAT\_StopDeviceOpTask 與 ECAT\_OpenDevice 時會自動禁用此功能

\_Directly 相關函式可以減少一般函式所花費的時間，但是有通道數上的限制，需要使用此函式設定模組，沒有被設定的模組無法使用\_Directly 相關函式

註: (1)若大部分的模組的 DI 數都小於 16 點，可以使用此函示，並且通道數上的限制為 ECAT\_SetSlaveDIMap 的 2 倍

(2)不可與 ECAT\_SetSlaveDIMap 同時使用

### 格式:

int32\_t ECAT\_SetSlaveDIMap\_16bit(uint16\_t DeviceNo, uint16\_t ChannelCnt, uint16\_t\* SlaveNo, uint16\_t\* Offset, uint16\_t\* Dir)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelCnt	uint16_t	IN	要設定的通道數量
SlaveNo	uint16_t []	IN	從站編號
Offset	uint16_t []	IN	點位偏移量
Dir	uint32_t []	IN	0: 輸入埠資料 1: 輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [SLAVE_DIO_MAP_MAX_EX];
uint16_t Offset [SLAVE_DIO_MAP_MAX_EX];
uint32_t Dir[SLAVE_DIO_MAP_MAX_EX];
uint16_t ChannelCnt = 0;
```

```
/*
```

```
  MainDevice
```

```
  |- ECAT-2057 DO module
```

```
  |- ECAT-2051 DI module
```

```
  |- ECAT-2055 DI& DO module
```

```
  L ECAT-2051-32 DI module
```

```
*/
```

```
// ECAT-2057 DO
```

```
SlaveNo[ChannelCnt] = 0;
```

```
Offset[ChannelCnt] = 0;
```

```
Dir[ChannelCnt] = 1;
```

```
ChannelCnt++;
```

```
// ECAT-2051 DI
```

```
SlaveNo[ChannelCnt] = 1;
```

```
Offset[ChannelCnt] = 0;
```

```
Dir[ChannelCnt] = 0;
```

```
ChannelCnt++;
```

```
// ECAT-2055 DI&DO
```

```
SlaveNo[ChannelCnt] = 2;
```

```
Offset[ChannelCnt] = 0;
```

```
Dir[ChannelCnt] = 0;//DI
```

```
ChannelCnt++;
```

```
SlaveNo[ChannelCnt] = 2;
```

```
Offset[ChannelCnt] = 0;
```

```
Dir[ChannelCnt] = 1;//DO
```

```
ChannelCnt++;
```

```
//for 32bit modules ECAT-2051-32 DI
```

```
SlaveNo[ChannelCnt] = 3;
```

```
Offset[ChannelCnt] = 0; //DI bit 0~15
```

```
Dir[ChannelCnt] = 0;
```

```
ChannelCnt++;
```

```
SlaveNo[ChannelCnt] = 3;
```

```
Offset[ChannelCnt] = 2; //DI bit 16~31
```

```
Dir[ChannelCnt] = 0;
```

```
ChannelCnt= 5;
```

```
ret = ECAT_SetSlaveDIOMap_16bit(DeviceNo, ChannelCnt, SlaveNo, Offset, Dir);
```

```
if(ret < 0)
```

```
    printf("Failed to set slave DIO Map:%d\n", ret);
```

```
else
```

```
    printf("Set slave DIO Map successfully! \n");
```

---

## 6.11. ECAT\_GetSlaveDI

### 說明:

取得從站模組 DI 資料。這裡假設 DI 是 32-bit 或以下，地址由 0 開始且連續定義。這是 *ECAT\_GetSlaveTxPdoData\_Ex* 函式對 DI 模組處理的簡化版。

### 格式:

```
int32_t ECAT_GetSlaveDI(uint16_t DeviceNo, uint16_t SlaveNo, uint32_t *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Value	uint32_t*	OUT	輸入埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint32_t Value;
```

```
ret = ECAT_GetSlaveDI(DeviceNo, SlaveNo, &Value);
```

```
if(ret < 0)
```

```
    printf("Failed to get slave DI:%d\n", ret);
```

```
else
```

```
    printf("DI:%u! \n", Value);
```

---

## 6.12. ECAT\_GetSlaveDI\_Directly

### 說明:

降低使用 ECAT\_GetSlaveDI 所花費的時間

注:需先啟用 ECAT\_SetSlaveDIMap

## 6.13. ECAT\_GetSlaveDI\_Directly\_16bit

### 說明:

取得從站模組 DI 資料。

注:需先啟用 ECAT\_SetSlaveDIMap\_16Bit

### 格式:

```
int32_t ECAT_GetSlaveDI_Directly_16bit(uint16_t DeviceNo, uint16_t* Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Value	uint32_t*	OUT	輸入埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [SLAVE_DIO_MAP_MAX_EX];
uint16_t Offset [SLAVE_DIO_MAP_MAX_EX];
uint16_t Value[SLAVE_DIO_MAP_MAX_EX];
uint32_t Dir[SLAVE_DIO_MAP_MAX_EX];
uint16_t ChannelCnt = 0;

/*
MainDevice
|- ECAT-2057 DO module
|- ECAT-2051 DI module
|- ECAT-2055 DI& DO module
L ECAT-2051-32 DI module

*/
// ECAT-2057 DO
SlaveNo[ChannelCnt] = 0;
Offset[ChannelCnt] = 0;
Dir[ChannelCnt] = 1;

ChannelCnt++;

// ECAT-2051 DI
SlaveNo[ChannelCnt] = 1;
Offset[ChannelCnt] = 0;
Dir[ChannelCnt] = 0;

ChannelCnt++;

// ECAT-2055 DI&DO
SlaveNo[ChannelCnt] = 2;
Offset[ChannelCnt] = 0;
Dir[ChannelCnt] = 0;//DI

```

```
ChannelCnt++;
```

```
SlaveNo[ChannelCnt] = 2;
```

```
Offset[ChannelCnt] = 0;
```

```
Dir[ChannelCnt] = 1;//DO
```

```
ChannelCnt++;
```

```
//for 32bit modules ECAT-2051-32 DI
```

```
SlaveNo[ChannelCnt] = 3;
```

```
Offset[ChannelCnt] = 0; //DI bit 0~15
```

```
Dir[ChannelCnt] = 0;
```

```
ChannelCnt++;
```

```
SlaveNo[ChannelCnt] = 3;
```

```
Offset[ChannelCnt] = 2; //DI bit 16~31
```

```
Dir[ChannelCnt] = 0;
```

```
ChannelCnt= 5;
```

```
ret = ECAT_SetSlaveDI_Map_16bit(DeviceNo, ChannelCnt, SlaveNo, Offset, Dir);
```

```
if(ret < 0)
```

```
    printf("Failed to set slave DIO Map:%d\n", ret);
```

```
else
```

```
    printf("Set slave DIO Map successfully! \n");
```

```
//after start op task
```

```
ret = ECAT_GetSlaveDI_Directly_16bit(DeviceNo, Value);
```

```
if(ret < 0)
```

```
    printf("Failed to get slave Di:%d\n", ret);
```

```
else
```

```
    for(uint16_t i = 0, i < ChannelCnt, i++)
```

```
    {
```

```
        printf("Value[%u]:%f\n", i, Value[ i ]);
```

```
    }
```



## 6.14. ECAT\_GetSlaveDIBit

### 說明:

取得從站模組 DI 指定位元資料。這裡假設 DI 是 32-bit 或以下, 地址由 0 開始且連續定義。這是 *ECAT\_GetSlaveTxPdoData\_Ex* 函式對 DI 模組處理的簡化版再加上 bit 處理。

### 格式:

```
int32_t ECAT_GetSlaveDIBit(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t BitNo,
uint32_t *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
BitNo	uint16_t	IN	指定位元
Value	uint32_t*	OUT	輸入埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 2;
uint32_t Value;

ret = ECAT_GetSlaveDIBit(DeviceNo, SlaveNo, BitNo, &Value);
if(ret < 0)
    printf("Failed to get slave DI:%d\n", ret);
else
    printf("DI_Bit[%u]:%u! \n", BitNo, Value);
```

---

## 6.15. ECAT\_GetSlaveDIBit\_Directly

### 說明:

降低使用 ECAT\_GetSlaveDIBit 所花費的時間

注:需先啟用 ECAT\_SetSlaveDIMap

## 6.16. ECAT\_GetSlaveDO

### 說明:

取得從站模組 DO 資料。這裡假設 DO 是 32-bit 或以下，地址由 0 開始且連續定義。這是 *ECAT\_GetSlaveRxPdoData\_Ex* 函式對 DO 模組處理的簡化版。

### 格式:

```
int32_t ECAT_GetSlaveDO(uint16_t DeviceNo, uint16_t SlaveNo, uint32_t *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Value	uint32_t*	OUT	輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint32_t Value;

ret = ECAT_GetSlaveDO(DeviceNo, SlaveNo, &Value);
if(ret < 0)
    printf("Failed to get slave DO:%d\n",ret);
else
    printf("DO:%u! \n",Value);
```

---



## 6.17. ECAT\_GetSlaveDO\_Directly

### 說明:

降低使用 ECAT\_GetSlaveDI 所花費的時間

注:需先啟用 ECAT\_SetSlaveDIMap

## 6.18. ECAT\_GetMultiSlaveDO\_Ex

### 說明:

取代舊的函式"ECAT\_GetMultiSlaveDO"

取得從站模組 DO 資料。此函式可以同時讀取多個模組的資料。

### 格式:

```
int32_t ECAT_GetMultiSlaveDO_Ex(uint16_t DeviceNo, uint16_t SlaveNo[], uint32_t
*Value, uint16_t SlaveCnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t []	IN	從站編號
Value	uint32_t*	OUT	輸出埠資料
SlaveCnt	uint16_t	IN	模組數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [MULTI_SLAVE_DO_MAX];
uint32_t Value[MULTI_SLAVE_DO_MAX];
uint16_t SlaveCnt;

SlaveNo[0] = 0;
SlaveNo[1] = 1;
SlaveNo[2] = 2;

SlaveCnt = 3;

ret = ECAT_GetMultiSlaveDO_Ex(DeviceNo, SlaveNo, Value, SlaveCnt);
if(ret < 0)
    printf("Failed to get slave DO:%d\n",ret);
else
{
    printf("DO[0]:%x, DO[1]:%x, DO[2]:%x \n", Value[0] , Value[1] , Value[2]);
}
```

## 6.19. ECAT\_GetSlaveDOBit

### 說明:

取得從站模組 DO 指定位元資料。這裡假設 DO 是 32-bit 或以下，地址由 0 開始且連續定義。這是 *ECAT\_GetSlaveRxPdoData\_Ex* 函式對 DO 模組處理的簡化版再加上 bit 處理。

### 格式:

```
int32_t ECAT_GetSlaveDOBit(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t BitNo,
uint32_t *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
BitNo	uint16_t	IN	指定位元
Value	uint32_t*	OUT	輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 2;
uint32_t Value;

ret = ECAT_GetSlaveDOBit(DeviceNo, SlaveNo, BitNo, &Value);
if(ret < 0)
    printf("Failed to get slave DO bit:%d\n",ret);
else
    printf("DO_Bit[%u]:%u! \n", BitNo, Value);
```

---

## 6.20. ECAT\_GetSlaveDOBit\_Directly

### 說明:

降低使用 ECAT\_GetSlaveDOBit 所花費的時間

注:需先啟用 ECAT\_SetSlaveDIMap

## 6.21. ECAT\_SetSlaveDO

### 說明:

設定從站模組 DO 資料。這裡假設 DO 是 32-bit 或以下，地址由 0 開始且連續定義。這是 *ECAT\_SetSlaveRxPdoData* 函式對 DO 模組處理的簡化版。

### 格式:

```
int32_t ECAT_SetSlaveDO(uint16_t DeviceNo, uint16_t SlaveNo, uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Value	uint32_t	IN	輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint32_t Value = 255;
```

```
ret = ECAT_SetSlaveDO(DeviceNo, SlaveNo, Value);
```

```
if(ret < 0)
```

```
    printf("Failed to set slave DO:%d\n", ret);
```

```
else
```

```
    printf("Set slave DO successfully! \n");
```

---



## 6.22. ECAT\_SetMultiSlaveDO\_Ex

### 說明:

取代舊的函式"ECAT\_SetMultiSlaveDO"

設定從站模組 DO 資料。類似 ECAT\_SetSlaveDO, 此函式可以同時對多個模組下達指令。

### 格式:

```
int32_t ECAT_SetMultiSlaveDO_Ex(uint16_t DeviceNo, uint16_t SlaveNo[], uint32_t Value[], uint16_t SlaveCnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t []	IN	從站編號
Value	uint32_t []	IN	輸出埠資料
SlaveCnt	uint16_t	IN	要設定的模組數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [MULTI_SLAVE_DO_MAX];
uint32_t Value[MULTI_SLAVE_DO_MAX];
uint16_t SlaveCnt;

/*
MainDevice
|- ECAT-2057
|- ECAT-2057
L ECAT-2057
*/

SlaveNo[0] = 0;
Value[0] = 0xFFFF;

SlaveNo[1] = 1;
Value[1] = 0xFFFF;

SlaveNo[2] = 2;
Value[2] = 0xFFFF;

SlaveCnt = 3;

ret = ECAT_SetMultiSlaveDO_Ex(DeviceNo, SlaveNo, Value, SlaveCnt);
if(ret < 0)
    printf("Failed to set slave DO:%d\n", ret);
else
    printf("Set slave DO successfully! \n");
```

---

## 6.23. ECAT\_SetMultiSlaveDO\_AutoOff\_Ex

### 說明:

取代舊的函式"ECAT\_SetMultiSlaveDO\_AutoOff"

設定從站模組 DO 資料。類似 ECAT\_SetSlaveDO, 此函式可以同時對多個模組下達指令。  
在指定的時間(Width)之後, 會將指定的 DO bit(Mask)關閉。

### 格式:

```
int32_t ECAT_SetMultiSlaveDO_AutoOff_Ex(uint16_t DeviceNo, uint16_t SlaveNo[],
uint32_t Value[], uint32_t Width[], uint32_t Mask[], uint16_t SlaveCnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t []	IN	從站編號
Value	uint32_t []	IN	輸出埠資料
Width	uint32_t []	IN	時間, EtherCAT 通訊週期的倍數
Mask	uint32_t []	IN	遮罩, 僅在有遮罩設定的 BIT 會被關閉
SlaveCnt	uint16_t	IN	要設定的模組數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [MULTI_SLAVE_DO_MAX];
uint32_t Value[MULTI_SLAVE_DO_MAX];
uint32_t Width[MULTI_SLAVE_DO_MAX];
uint32_t Mask[MULTI_SLAVE_DO_MAX];
uint16_t SlaveCnt;

/*
MainDevice
|- ECAT-2057
|- ECAT-2057
L ECAT-2057
*/

SlaveNo[0] = 0;
Value[0] = 0xFFFF;
Width [0] = 30; //if EtherCAT cycle time is 1ms, then Pulse Width is 1ms*30 = 30ms
Mask [0] = 0xFFFF; //after 30ms(Width), 16 bits of DO will be turned off

SlaveNo[1] = 1;
Value[1] = 0xFFFF;
Width [1] = 30;
Mask [1] = 0x00FF; //after 30ms(Width), the first 8 bits of DO will be turned off, and the last 8bits of DO will
remain on

SlaveNo[2] = 2;
Value[2] = 0xFFFF;
Width [2] = 30;
Mask [2] = 0xFF00; //after 30ms(Width), the last 8 bits of DO will be turned off, and the first 8bits of DO will
remain on

SlaveCnt = 3;

ret = ECAT_SetMultiSlaveDO_AutoOff_Ex(DeviceNo, SlaveNo, Value, Width, Mask, SlaveCnt);
if(ret < 0)

```

---

```
printf("Failed to set slave DO:%d\n", ret);  
else  
printf("Set slave DO successfully! \n");
```

---

## 6.24. ECAT\_SetSlaveDOBit

### 說明:

設定從站模組 DO 指定位元資料。這裡假設 DO 是 32-bit 或以下，地址由 0 開始且連續定義。這是 *ECAT\_SetSlaveRxPdoData* 函式對 DO 模組處理的簡化版再加上 bit 處理。

### 格式:

```
int32_t ECAT_SetSlaveDOBit(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t BitNo,
uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
BitNo	uint16_t	IN	指定位元
Value	uint32_t	IN	輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 2;
uint32_t Value = 1;

ret = ECAT_GetSlaveDOBit(DeviceNo, SlaveNo, BitNo, Value);
if(ret < 0)
    printf("Failed to set slave DI bit:%d\n", ret);
else
    printf("Set slave DO bit successfully! \n");
```

---

## 6.25. ECAT\_do\_cfg\_save

### 說明:

儲存從站模組 DO 資料到 PC 上。

路徑為 C:\icpdas\Ecat-M801\ECAT\_DO\_Config.xml

### 格式:

```
int32_t ECAT_do_cfg_save(uint16_t DeviceNo, uint16_t SlaveNo[], uint16_t Offset[],
uint16_t Value[], uint16_t ChannelCnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t []	IN	從站編號
Offset	uint16_t []	IN	點位偏移量
Value	uint32_t []	IN	輸出埠資料
ChannelCnt	uint16_t	IN	要設定的通道數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [SLAVE_DIO_MAP_MAX_EX];
uint16_t Offset [SLAVE_DIO_MAP_MAX_EX];
uint16_t Value[SLAVE_DIO_MAP_MAX_EX];
uint32_t Dir[SLAVE_DIO_MAP_MAX_EX];
uint16_t ChannelCnt = 0;

/*
MainDevice
L - ECAT-2045K-32 DO module

*/
// ECAT-2045K-32 DO
SlaveNo[ChannelCnt] = 0;
Offset[ChannelCnt] = 0; // bit 0~15
Value[ChannelCnt] = 0xFFFF;

ChannelCnt++;

SlaveNo[ChannelCnt] = 3;
Offset[ChannelCnt] = 2; // bit 16~31
Value [ChannelCnt] = 0xFFFF;

ChannelCnt++;

ChannelCnt= 2;

ret = ECAT_do_cfg_save(DeviceNo, SlaveNo, Offset, Value, ChannelCnt);
if(ret < 0)
    printf("Failed to save do config:%d\n", ret);
else
    printf("save do config successfully! \n");

```

## 6.26. ECAT\_do\_cfg\_load

### 說明:

讀取 ECAT\_DO\_Config.xml 檔並設定模組 DO 資料。

路徑為 C:\icpdas\Ecat-M801\ECAT\_DO\_Config.xml

在 ECAT\_StartDeviceOpTask 之後，DO 的預設值為 0，此函示可以在 ECAT\_StartDeviceOpTask 之前設定模組 DO 資料，並在 OP 後保持 DO 資料

注:支援 ECAT-2045K-32

### 格式:

int32\_t ECAT\_do\_cfg\_load(uint16\_t DeviceNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
ret = ECAT_do_cfg_load(DeviceNo);
```

```
if(ret < 0)
```

```
    printf("Failed to load do config:%d\n", ret);
```

```
else
```

```
    printf("load do config successfully! \n");
```

---

## 6.27. ECAT\_ClearDoQueue

### 說明:

設定模組 DO 預設值為 0。

注:支援 ECAT-2045K-32

### 格式:

```
int32_t ECAT_ClearDoQueue(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
  
ret = ECAT_ClearDoQueue(DeviceNo);  
if(ret < 0)  
    printf("Failed to clear do config:%d\n", ret);  
else  
    printf("Clear do config successfully! \n");
```

---

## 6.28. ECAT\_PreSetSlaveDO

### 說明:

設定從站模組 DO 資料。

在 ECAT\_StartDeviceOpTask 之後，DO 的預設值為 0，此函示可以在 ECAT\_StartDeviceOpTask 之前設定模組 DO 資料，並在 OP 後保持 DO 資料

注:支援 ECAT-2045K-32

### 格式:

int32\_t ECAT\_PreSetSlaveDO(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t Offset, uint16\_t Value)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Offset	uint16_t	IN	點位偏移量
Value	uint32_t	IN	輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t Offset = 0;
uint32_t Value = 255;

ret = ECAT_ClearDoQueue(DeviceNo);
if(ret < 0)
    printf("Failed to clear do config:%d\n", ret);
else
    printf("Clear do config successfully! \n");

ret = ECAT_PreSetSlaveDO(DeviceNo, SlaveNo, Offset, Value);
if(ret < 0)
    printf("Failed to set slave DO:%d\n", ret);
else
    printf("Set slave DO successfully! \n");
```

---

## 6.29. ECAT\_SetSlaveAoProperty

### 說明:

設定從站模組類比輸出的參數值。此函式內部使用 SDO 通訊來處理，指定輸出範圍。

注:支援 ECAT-2024 / ECAT-2028

### 格式:

```
int32_t ECAT_SetSlaveAoProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, uint8_t Range)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Range	uint8_t	IN	類比輸出通道的電壓配置碼(如表 6.4 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



表 6.4: 類比輸出通道的電壓配置碼定義

標頭檔定義名稱	數值	說明
SLAVE_AO_UNI_5V	0	電壓範圍 0 ~ 5V
SLAVE_AO_BI_5V	1	電壓範圍 $\pm 5V$
SLAVE_AO_UNI_10V	2	電壓範圍 0 ~ 10V
SLAVE_AO_BI_10V	3	電壓範圍 $\pm 10V$

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range = SLAVE_AO_UNI_10V;

ret = ECAT_SetSlaveAoProperty(DeviceNo, SlaveNo, ChannelNo, Range);
if(ret != 0)
    printf("Failed to set slave AO settings:%d\n", ret);
else
    printf("Set slave AO settings successfully! \n");

```

## 6.30. ECAT\_GetSlaveAoProperty

### 說明:

取得從站模組類比輸出的參數值。此函式內部使用 SDO 通訊來處理，讀取輸出範圍設定。

注:支援 ECAT-2024 / ECAT-2028

### 格式:

```
int32_t ECAT_GetSlaveAoProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t *ChannelNo, uint8_t *Range)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Range	uint8_t *	OUT	類比輸出通道的電壓配置碼(如表 6.4 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range;

ret = ECAT_GetSlaveAoProperty(DeviceNo, SlaveNo, ChannelNo, &Range);
if(ret != 0)
    printf("Failed to get slave AO settings:%d\n", ret);
else
    printf("AO range:%d\n", Range);
```

---

## 6.31. ECAT\_SetSlaveAoRawData

### 說明:

使用者用給定 16 位元數值方式來設定從站模組類比輸出的電壓值。

注:支援 ECAT-2024 / ECAT-2028

### 格式:

```
int32_t ECAT_SetSlaveAoRawData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, int16_t Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	int16_t	IN	16 位元電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
int16_t Data = 0xFF;
```

```
ret = ECAT_SetSlaveAoRawData(DeviceNo, SlaveNo, ChannelNo, Data);
```

```
if(ret != 0)
```

```
    printf("Failed to set slave AO raw data:%d\n", ret);
```

```
else
```

```
    printf("Set slave AO raw data successfully! \n");
```

---

## 6.32. ECAT\_GetSlaveAoRawData

### 說明:

取得從站模組類比輸出的電壓值，以 16 位元數值方式來表示。

注:支援 ECAT-2024 / ECAT-2028

### 格式:

```
int32_t ECAT_GetSlaveAoRawData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, int16_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	int16_t *	OUT	16 位元電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
int16_t Data;
```

```
ret = ECAT_GetSlaveAoRawData(DeviceNo, SlaveNo, ChannelNo, &Data);
```

```
if(ret != 0)
```

```
    printf("Failed to get slave AO raw data:%d\n", ret);
```

```
else
```

```
    printf("AO raw data:%d\n", Data);
```

---

## 6.33. ECAT\_SetSlaveAoVoltData

### 說明:

使用者以輸入電壓範圍內的浮點數的方式來設定從站模組類比輸出的電壓值。

注:支援 ECAT-2024 / ECAT-2028

### 格式:

```
int32_t ECAT_SetSlaveAoVoltData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	double	IN	浮點電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
double Data = 5.5;

ret = ECAT_SetSlaveAoVoltData(DeviceNo, SlaveNo, ChannelNo, Data);
if(ret != 0)
    printf("Failed to set slave AO volt data:%d\n", ret);
else
    printf("Set slave AO volt data successfully! \n");
```

---

## 6.34. ECAT\_GetSlaveAoVoltData

### 說明:

取得從站模組類比輸出的電壓值，其值用輸出電壓範圍內的浮點數方式來表示。

注:支援 ECAT-2024 / ECAT-2028

### 格式:

```
int32_t ECAT_GetSlaveAoVoltData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	Double *	OUT	浮點電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
double Data;
```

```
ret = ECAT_GetSlaveAoVoltData(DeviceNo, SlaveNo, ChannelNo, &Data);
```

```
if(ret != 0)
```

```
    printf("Failed to get slave AO volt data:%d\n", ret);
```

```
else
```

```
    printf("AO volt data:%d\n", Data);
```

---

## 6.35. ECAT\_SetSlaveAiProperty

### 說明:

設定從站模組類比輸入的參數值。此函式內部使用 SDO 通訊來處理，指定輸入範圍。

注:(1)支援 ECAT-2011H、ECAT-2012H

(2)修改其中一個 channel 的數值，其餘 channel 的數值也會一起被改變。

### 格式:

```
int32_t ECAT_SetSlaveAiProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, uint8_t Range)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Range	uint8_t	IN	類比輸入通道的電壓配置碼(如表 6.5 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.5: 類比輸入通道的電壓/電流配置碼定義

標頭檔定義名稱	數值	說明
SLAVE_AI_BI_10V	0	電壓範圍 $\pm 10V$
SLAVE_AI_BI_5V	1	電壓範圍 $\pm 5V$
SLAVE_AI_BI_2_5V	2	電壓範圍 $\pm 2.5V$
SLAVE_AI_UNI_10V	3	電壓範圍 $0 \sim 10V$
SLAVE_AI_UNI_20mA	4	電流範圍 $0 \sim 20mA$
SLAVE_AI_UNI_4_20mA	5	電流範圍 $4 \sim 20mA$
SLAVE_AI_BI_20mA	6	電流範圍 $\pm 0 \sim 20mA$
SLAVE_AI_BI_4_20mA	7	電流範圍 $\pm 4 \sim 20mA$
SLAVE_AI_BI_10V_UNI_20mA	8	CH0~3 電壓範圍 $\pm 10V$ , CH4~7 電流範圍 $0 \sim 20mA$
SLAVE_AI_BI_10V_UNI_4_20mA	9	CH0~3 電壓範圍 $\pm 10V$ , CH4~7 電流範圍 $4 \sim 20mA$
SLAVE_AI_BI_10V_BI_20mA	10	CH0~3 電壓範圍 $\pm 10V$ , CH4~7 電流範圍 $\pm 0 \sim 20mA$
SLAVE_AI_BI_10V_BI_4_20mA	11	CH0~3 電壓範圍 $\pm 10V$ , CH4~7 電流範圍 $\pm 4 \sim 20mA$

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range = SLAVE_AI_BI_10V;

ret = ECAT_SetSlaveAiProperty(DeviceNo, SlaveNo, ChannelNo, Range);
if(ret != 0)
    printf("Failed to set slave AI settings:%d\n", ret);
else
    printf("Set slave AI settings successfully! \n");

```



## 6.36. ECAT\_GetSlaveAiProperty

### 說明:

取得從站模組類比輸入的參數值。此函式內部使用 SDO 通訊來處理，讀取輸入範圍設定。

注:支援 ECAT-2011H、ECAT-2012H

### 格式:

```
int32_t ECAT_GetSlaveAiProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t
*ChannelNo, uint8_t *Range)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Range	uint8_t *	OUT	類比輸入通道的電壓/電流配置碼(如表 6.5 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
uint8_t Range;
```

```
ret = ECAT_GetSlaveAiProperty(DeviceNo, SlaveNo, ChannelNo, &Range);
```

```
if(ret != 0)
```

```
    printf("Failed to get slave AI settings:%d\n", ret);
```

```
else
```

```
    printf("AI range:%d\n", Range);
```

---



## 6.37. ECAT\_GetSlaveAiRawData

### 說明:

取得從站模組類比輸入的電壓/電流值，以 16 位元數值方式來表示。

注:支援 ECAT-2011H、ECAT-2012H

### 格式:

```
int32_t ECAT_GetSlaveAiRawData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, int16_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	int16_t *	OUT	16 位元電壓/電流值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
int16_t Data;

ret = ECAT_GetSlaveAiRawData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AI raw data:%d\n", ret);
else
    printf("AI raw data:%d\n", Data);
```

---

## 6.38. ECAT\_GetSlaveAiVoltData

### 說明:

取得從站模組類比輸入的電壓值，其值用輸入電壓範圍內的浮點數方式來表示。

注:支援 ECAT-2011H、ECAT-2012H

### 格式:

```
int32_t ECAT_GetSlaveAiVoltData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	Double *	OUT	浮點電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
double Data;

ret = ECAT_GetSlaveAiVoltData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AI volt data:%d\n", ret);
else
    printf("AI volt data:%d\n", Data);
```

---

## 6.39. ECAT\_GetSlaveAimAData

### 說明:

取得從站模組類比輸入的電流值，其值用輸入電流範圍內的浮點數方式來表示。

注:支援 ECAT-2011H、ECAT-2012H

### 格式:

```
int32_t ECAT_GetSlaveAimAData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	Double *	OUT	浮點電流值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
double Data;

ret = ECAT_GetSlaveAimAtData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AI mA data:%d\n", ret);
else
    printf("AI mA data:%d\n", Data);
```

---

## 6.40. ECAT\_Set\_ECAT2016\_AiProperty

### 說明:

設定從站模組類比輸入的參數值。此函式內部使用 SDO 通訊來處理，指定輸入範圍。

注:(1)支援 ECAT-2016N、ECAT-2016-3

(2)修改其中一個 channel 的數值，其餘 channel 的數值也會一起被改變。

### 格式:

```
int32_t ECAT_Set_ECAT2016_AiProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t ChannelNo, uint8_t Range, uint32_t *AbortCode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Range	uint8_t	IN	類比輸入通道的電壓配置碼(如表 6.6 所示)
AbortCode	uint32_t*	OUT	SDO 操作錯誤終止代碼(請參考附錄 "SDO 操作錯誤終止代碼"說明)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.6: 類比輸入通道的電壓/電流配置碼定義

標頭檔定義名稱	數值	說明
ECAT2016_AI_BI_10V	0	電壓範圍 ±10V
ECAT2016_AI_BI_5V	1	電壓範圍 ±5V
ECAT2016_AI_BI_2_5V	2	電壓範圍 ±2.5V
ECAT2016_AI_BI_1V	3	電壓範圍 ±1V
ECAT2016_AI_BI_500mV	4	電壓範圍 ±500mV
ECAT2016_AI_BI_100mV	5	電壓範圍 ±100mV
ECAT2016_AI_BI_50mV	6	電壓範圍 ±50mV
ECAT2016_AI_BI_25mV	7	電壓範圍 ±25mV
ECAT2016_AI_BI_20mV	8	電壓範圍 ±20mV
ECAT2016_AI_BI_16mV	9	電壓範圍 ±16mV
ECAT2016_AI_BI_15mV	10	電壓範圍 ±15mV

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range = ECAT2016_AI_BI_10V;
uint32_t Abortcode;

ret = ECAT_Set_ECAT2016_AiProperty(DeviceNo, SlaveNo, ChannelNo, Range, &Abortcode);
if(ret != 0)
    printf("Failed to set slave AI settings:%d\n", ret);
else
    printf("Set slave AI settings successfully! \n");

```



## 6.41. ECAT\_Get\_ECAT2016\_AiProperty

### 說明:

取得從站模組類比輸入的參數值。此函式內部使用 SDO 通訊來處理，讀取輸入範圍設定。

注:支援 ECAT-2016N、ECAT-2016-3

### 格式:

```
int32_t ECAT_Get_ECAT2016_AiProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t *ChannelNo, uint8_t *Range, uint32_t *AbortCode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Range	uint8_t *	OUT	類比輸入通道的電壓/電流配置碼(如表 6.6 所示)
AbortCode	uint32_t*	OUT	SDO 操作錯誤終止代碼(請參考附錄 "SDO 操作錯誤終止代碼"說明)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
uint8_t Range;
```

```
uint32_t Abortcode;
```

```
ret = ECAT_Get_EC2016_AiProperty(DeviceNo, SlaveNo, ChannelNo, &Range, &Abortcode);
```

```
if(ret != 0)
```

```
    printf("Failed to get slave AI settings:%d\n", ret);
```

```
else
```

```
    printf("AI range:%d\n", Range);
```

---

## 6.42. ECAT\_Get\_ECAT2016\_AiRawData

### 說明:

取得從站模組類比輸入的電壓值，以 16 位元數值方式來表示。

注:支援 ECAT-2016N、ECAT-2016-3

### 格式:

```
int32_t ECAT_Get_ECAT2016_AiRawData(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t ChannelNo, int16_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	int16_t *	OUT	16 位元電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
int16_t Data;
```

```
ret = ECAT_Get_ECATA2016_AiRawData(DeviceNo, SlaveNo, ChannelNo, &Data);
```

```
if(ret != 0)
```

```
    printf("Failed to get slave AI raw data:%d\n", ret);
```

```
else
```

```
    printf("AI raw data:%d\n", Data);
```

---

## 6.43. ECAT\_Get\_ECAT2016\_AiVoltData

### 說明:

取得從站模組類比輸入的電壓值，其值用輸入電壓範圍內的浮點數方式來表示。

注:支援 ECAT-2016N、ECAT-2016-3

### 格式:

```
int32_t ECAT_Get_ECAT2016_AiVoltData(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t ChannelNo, double *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	通道編號
Data	Double *	OUT	浮點電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t ChannelNo = 0;
```

```
double Data;
```

```
ret = ECAT_Get_ECATAI2016_AiVoltData(DeviceNo, SlaveNo, ChannelNo, &Data);
```

```
if(ret != 0)
```

```
    printf("Failed to get slave AI volt data:%d\n", ret);
```

```
else
```

```
    printf("AI volt data:%d\n", Data);
```

## 6.44. ECAT\_SetSlaveEncProperty

### 說明:

設定從站模組編碼器參數值。

注:支援 ECAT-2093, ECAT-2092T

### 格式:

```
int32_t ECAT_SetSlaveEncProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Mode, uint8_t InvertCnt, uint8_t LPF)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Mode	uint8_t	IN	編碼器輸入模式 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	IN	反相計數
LPF	uint8_t	IN	低通濾波器設定(如表 6.7 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.7: 低通濾波器設定值定義

Low pass filter number	Maximum Input Frequency	
	Pulse/Direction counting mode Clockwise/Counterclockwise mode	Quadrant counting mode
0	4MHz (filter disabled)	6MHz (filter disabled)
1	4MHz	1MHz
2	2MHz	500KHz
3	1MHz	250KHz
4	640KHz	160KHz
5	320KHz	80KHz
6	160KHz	40Hz
7	80KHz	20KHz
8	40KHz	10KHz

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Mode = 3; //A/B Phase
uint8_t InvertCnt = 1; //Enable Reverse
uint8_t LPF = 0;
ret = ECAT_SetSlaveEncProperty(DeviceNo, SlaveNo, EncNo, Mode, InvertCnt, LPF);
if(ret != 0)
    printf("Failed to set encoder property:%d\n",ret);
else
    printf("Set encoder property successfully! \n");

```



## 6.45. ECAT\_GetSlaveEncProperty

### 說明:

取得從站模組編碼器參數值。

注:支援 ECAT-2093 , ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveEncProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t *Mode, uint8_t *InvertCnt, uint8_t *LPF)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Mode	uint8_t *	OUT	編碼器輸入模式 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t *	OUT	反相計數
LPF	uint8_t *	OUT	低通濾波器設定(如表 6.7 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Mode;
uint8_t ReverseCnt ;
uint8_t LPF;
ret = ECAT_GetSlaveEncProperty(DeviceNo, SlaveNo, EncNo, &Mode, &ReverseCnt, &LPF);
if(ret != 0)
{
    printf("Failed to get encoder property:%d\n", ret);
}
else
{
    printf("Encoder mode:%u\n", Mode);
    printf("Encoder reverse:%u\n", ReverseCnt);
    printf("Encoder Low Pass Filter:%u\n", LPF);
}
```

## 6.46. ECAT\_GetSlaveEncCount

### 說明:

取得從站模組編碼器計數器數值。

注:支援 ECAT-2093, ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveEncCount(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t *Cnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Cnt	int32_t *	OUT	編碼器計數器數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Cnt;
ret = ECAT_GetSlaveEncCount(DeviceNo, SlaveNo , EncNo, &Cnt);
if(ret != 0)
{
    printf("Failed to get encoder count:%d\n",ret);
}
else
{
    printf("Encoder count:%d\n",Cnt);
}
```

---

## 6.47. ECAT\_ResetSlaveEncCount

### 說明:

清除從站模組編碼器計數器數值。

注:支援 ECAT-2093, ECAT-2092T

### 格式:

```
int32_t ECAT_ResetSlaveEncCount(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
ret = ECAT_ResetSlaveEncCount(DeviceNo, SlaveNo , EncNo);
if(ret != 0)
{
    printf("Failed to reset encoder count:%d\n",ret);
}
else
{
    printf("Reset encoder count successfully!\n");
}
```

---

## 6.48. ECAT\_SetSlaveEncCount

### 說明:

設定從站模組編碼器計數器數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_SetSlaveEncCount(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo,
int32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Value	int32_t	IN	編碼器計數器數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Value = 100;

ret = ECAT_SetSlaveEncCount(DeviceNo, SlaveNo , EncNo, Value);
if(ret != 0)
{
    printf("Failed to set encoder count:%d\n",ret);
}
else
{
    printf("Set encoder count successfully!\n");
}
```

---



## 6.49. ECAT\_SetSlaveEncIdxLatchProperty

### 說明:

設定從站模組索引輸入鎖存器(Index Latch)功能參數值。

注:(1)僅支援 ECAT-2093

(2) ECAT-2092T 索引輸入鎖存器始終保持啟用狀態且無法禁用

### 格式:

```
int32_t ECAT_SetSlaveEncIdxLatchProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, uint8_t Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Enable	uint8_t	IN	啟用/停用鎖存器

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Enable = 1;

ret = ECAT_SetSlaveEncIdxLatchProperty(DeviceNo, SlaveNo, EncNo, Enable);
if(ret != 0)
{
    printf("Failed to set encoder index latch property:%d\n",ret);
}
else
{
    printf("Set index latch property successfully!\n");
}
```

---

## 6.50. ECAT\_GetSlaveEncIdxLatchProperty

### 說明:

取得從站模組索引輸入鎖存器(Index Latch)功能參數值。

注:支援 ECAT-2093, ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveEncIdxLatchProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, uint8_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Enable	uint8_t *	OUT	啟用/停用鎖存器

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
uint8_t Enable;
```

```
ret = ECAT_GetSlaveEncIdxLatchProperty (DeviceNo, SlaveNo, EncNo, &Enable);
```

```
if(ret != 0)
```

```
{
```

```
    printf("Failed to get index latch property:%d\n",ret);
```

```
}
```

```
else
```

```
{
```

```
    printf("Index latch enable:%u\n", Enable);
```

```
}
```

## 6.51. ECAT\_GetSlaveEncIdxLatchCnt

### 說明:

取得從站模組索引輸入鎖存器(Index Latch)數值。

注:支援 ECAT-2093, ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveEncIdxLatchCnt(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t *Cnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Cnt	int32_t *	OUT	鎖存器數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
int32_t Cnt;
```

```
ret = ECAT_GetSlaveEnclIdxLatchCnt(DeviceNo, SlaveNo, EncNo, &Cnt);
```

```
if(ret != 0)
```

```
{
```

```
    printf("Failed to get index latch count:%d\n",ret);
```

```
}
```

```
else
```

```
{
```

```
    printf("Index latch count:%u\n", Cnt);
```

```
}
```

## 6.52. ECAT\_ResetSlaveEncIdxLatchCnt

### 說明:

清除從站模組索引輸入鎖存器(Index Latch)數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_ResetSlaveEncIdxLatchCnt(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
ret = ECAT_ResetSlaveEnclIdxLatchCnt(DeviceNo, SlaveNo, EncNo);
```

```
if(ret != 0)
```

```
{
```

```
    printf("Failed to reset index latch count:%d\n",ret);
```

```
}
```

```
else
```

```
{
```

```
    printf("Reset index latch count successfully!\n");
```

```
}
```

---



## 6.53. ECAT\_SetSlaveEncIdxLatchBufferEnable

### 說明:

設定從站模組索引輸入鎖存器(Index Latch)緩衝區(FIFO)功能的狀態。

注: (1)僅支援 ECAT-2093

(2)鎖存頻率與 EtherCAT 通訊週期相關。Ex: 通訊週期若為 1ms , 則鎖存頻率最高為 1kHz。

### 格式:

int32\_t ECAT\_SetSlaveEncIdxLatchBufferEnable(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t EncNo, uint8\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Enable	uint8_t	IN	啟用/停用鎖存器緩衝區功能

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
uint8_t Enable = 1;
```

```
ret = ECAT_SetSlaveEncIdxLatchBufferEnable(DeviceNo, SlaveNo, EncNo, Enable);
```

```
if(ret != 0)
```

```
{
```

```
    printf("Failed to set encoder index latch buffer enable:%d\n",ret);
```

```
}
```

```
else
```

```
{
```

```
    printf("Set index latch property successfully!\n");
```

```
}
```

## 6.54. ECAT\_GetSlaveEncIdxLatchBufferEnable

### 說明:

取得從站模組索引輸入鎖存器(Index Latch)緩衝區(FIFO)狀態。

注:支援 ECAT-2093

### 格式:

```
int32_t ECAT_GetSlaveEncIdxLatchBufferEnable(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, uint8_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Enable	uint8_t *	OUT	啟用/停用鎖存器緩衝區

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Enable;

ret = ECAT_GetSlaveEnclIdxLatchBufferEnable(DeviceNo, SlaveNo, EncNo, &Enable);
if(ret != 0)
{
    printf("Failed to get index latch buffer enable:%d\n",ret);
}
else
{
    printf("Index latch buffer enable:%u\n", Enable);
}
```

---

## 6.55. ECAT\_GetSlaveEncIdxLatchBuffer

### 說明:

取得從站模組索引輸入鎖存器(Index Latch)緩衝區(FIFO)數值。

注: (1)支援 ECAT-2093

(2)讀取資料後該資料會從緩衝區移出

(3)當緩衝區滿時，緩衝區不再更新資料，直到緩衝區有空位

### 格式:

```
int32_t ECAT_GetSlaveEncIdxLatchBuffer(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, int32_t GetCount, int32_t *ActualGetCount, int32_t *Data)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
GetCount	int32_t	IN	要取得的數量
ActualGetCount	int32_t *	OUT	實際取得的數量
Data	int32_t *	OUT	鎖存器數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t GetCount = ENC_BUFFER_MAX;
int32_t ActualGetCount;
int32_t Data[ENC_BUFFER_MAX];

ret = ECAT_GetSlaveEnclIdxLatchBuffer(DeviceNo, SlaveNo, EncNo, GetCount, &ActualGetCount, Data);
if(ret != 0)
{
    printf("Failed to get index latch buffer data:%d\n",ret);
}
else
{
    for(uint16_t i = 0, i < ActualGetCount, i++)
    {
        printf("Data[%u]:%f\n", i, Data[ i ]);
    }
}
```

---

## 6.56. ECAT\_ResetSlaveEncIdxLatchBuffer

### 說明:

清除從站模組索引輸入鎖存器(Index Latch)緩衝區(FIFO)。

注:僅支援 ECAT-2093

### 格式:

```
int32_t ECAT_ResetSlaveEncIdxLatchBuffer(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;

ret = ECAT_ResetSlaveEnclIdxLatchBuffer(DeviceNo, SlaveNo, EncNo);
if(ret != 0)
{
    printf("Failed to reset index latch buffer:%d\n",ret);
}
else
{
    printf("Reset index latch buffer successfully!\n");
}
```

---



## 6.57. ECAT\_SetSlaveEncExtLatchProperty

### 說明:

設定從站模組外部輸入鎖存器(External Latch)功能參數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_SetSlaveEncExtLatchProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, uint8_t Enable, uint8_t Mode, uint8_t Logic)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Enable	uint8_t	IN	啟用/停用鎖存器
Mode	uint8_t	IN	鎖存器模式 0: 重置編碼計數器 1: 鎖存編碼計數器
Logic	uint8_t	IN	觸發邏輯準位 0: 下降緣觸發(active low) 1: 上升緣觸發(active high)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
uint8_t Enable = 1;
```

```
uint8_t Mode = 0;
```

```
uint8_t Logic = 0;
```

```
ret = ECAT_SetSlaveEncExtLatchProperty(DeviceNo, SlaveNo, EncNo, Enable, Mode, Logic);
```

```
if(ret != 0)
```

```
{
```

```
    printf("Failed to set external latch property:%d\n",ret);
```

```
}
```

```
else
```

```
{
```

```
    printf("Set external latch property successfully!\n");
```

```
}
```

## 6.58. ECAT\_GetSlaveEncExtLatchProperty

### 說明:

取得從站模組外部輸入鎖存器(External Latch)功能參數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveEncExtLatchProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, uint8_t *Enable, uint8_t *Mode, uint8_t *Logic)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Enable	uint8_t *	OUT	啟用/停用鎖存器
Mode	uint8_t	IN	鎖存器模式 0: 重置編碼計數器 1: 鎖存編碼計數器
Logic	uint8_t	IN	觸發邏輯準位 0: 下降緣觸發(active low) 1: 上升緣觸發(active high)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Enable, Mode, Logic;

ret = ECAT_GetSlaveEncExtLatchProperty(DeviceNo, SlaveNo, EncNo, &Enable, &Mode, &Logic);
if(ret != 0)
{
    printf("Failed to get external latch property:%d\n",ret);
}
else
{
    printf("External latch enable:%u\n", Enable);
    printf("External latch mode:%u\n", Mode);
    printf("External latch logic:%u\n", Logic);
}
```

## 6.59. ECAT\_GetSlaveEncExtLatchCnt

### 說明:

取得從站模組外部輸入鎖存器(External Latch)數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveEncExtLatchCnt(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t *Cnt_Rising, int32_t *Cnt_Falling)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Cnt_Rising	int32_t *	OUT	上緣觸發之鎖存器數值
Cnt_Falling	int32_t *	OUT	下緣觸發之鎖存器數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Cnt_Rising;
int32_t Cnt_Falling;

ret = ECAT_GetSlaveEncExtLatchCnt(DeviceNo, SlaveNo, EncNo, &Cnt_Rising, &Cnt_Falling);
if(ret != 0)
{
    printf("Failed to get external latch count:%d\n",ret);
}
else
{
    printf("External latch count:%d, %d\n", Cnt_Rising, Cnt_Falling);
}
```

---

## 6.60. ECAT\_ResetSlaveEncExtLatchCnt

### 說明:

清除從站模組外部輸入鎖存器(External Latch)數值。

注:僅支援 ECAT-2092T

### 格式:

int32\_t ECAT\_ResetSlaveEncExtLatchCnt(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t EncNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;

ret = ECAT_ResetSlaveEncExtLatchCnt(DeviceNo, SlaveNo, EncNo);
if(ret != 0)
{
    printf("Failed to reset external latch count:%d\n",ret);
}
else
{
    printf("Reset external latch count successfully!\n");
}
```

---



## 6.61. ECAT\_SetSlaveCmpTrigProperty

### 說明:

設定從站模組比較觸發功能參數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_SetSlaveCmpTrigProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint16_t PulseWidth)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
PulseWidth	uint16_t	IN	比較觸發脈波寬度(如表 6.8 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.8: 比較觸發脈波寬度設定值定義

脈波寬度設定值	實際脈波寬度(微秒)
2	2
...	...
50 (default)	50
...	...
32766	32766
32767	32767

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint16_t PulseWidth = 40;
ret = ECAT_SetSlaveCmpTrigProperty(DeviceNo, SlaveNo, EncNo, PulseWidth);
if(ret != 0)
    printf("Failed to set compare trigger property:%d\n", ret);
```

## 6.62. ECAT\_GetSlaveCmpTrigProperty

### 說明:

取得從站模組比較觸發功能參數值。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_GetSlaveCmpTrigProperty(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, uint16_t *PulseWidth)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
PulseWidth	uint16_t *	OUT	比較觸發脈波寬度(如表 6.8 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint16_t PulseWidth;

ret = ECAT_GetSlaveCmpTrigProperty(DeviceNo, SlaveNo, EncNo, &PulseWidth);
if(ret != 0)
{
    printf("Failed to get compare triger property:%d\n", ret);
}
else
{
    printf("Compare triger pulse width:%u\n", PulseWidth);
}
```

---

## 6.63. ECAT\_SetSlaveCmpTrigData

### 說明:

開始從站模組單次比較觸發功能。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_SetSlaveCmpTrigData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t CmpData)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
CmpData	int32_t	IN	單次比較數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t CmpData = 1000;
ret = ECAT_SetSlaveCmpTrigData(DeviceNo, SlaveNo, EncNo, CmpData);
if(ret != 0)
    printf("Failed to set compare triger data:%d\n", ret);
```

---

## 6.64. ECAT\_SetSlaveContCmpTrigData

### 說明:

開始從站模組連續比較觸發功能。

注:僅支援 ECAT-2092T

### 格式:

```
int32_t ECAT_SetSlaveContCmpTrigData(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t EncNo, int32_t Start, uint32_t Interval, uint8_t Dir)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Start	int32_t	IN	開始比較的數值
Interval	uint32_t	IN	每次觸發的間隔數值
Dir	uint8_t	IN	比較方向 0: 向上計數時比較 1: 向下計數時比較

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Start = 1000;
uint32_t Interval = 200;
uint8_t Dir = 0;
ret = ECAT_SetSlaveContCmpTrigData(DeviceNo, SlaveNo, EncNo, Start, Interval, Dir);
if(ret != 0)
    printf("Failed to set continus compare trigger data:%d\n", ret);
```

---



## 6.65. ECAT\_SetSlaveArrCmpPos

### 說明:

設定陣列比較位置，啟用陣列比較觸發功能後，會從 index 0 所儲存的比較位置開始比較，當編碼器達到比較位置，就會觸發數位輸出。

注: (1) 僅支援 ECAT-2092T。

(2) 第一個要比較的值必須存放在 Index 0，第二個要比較的值必須存放在 Index 1，以此類推。

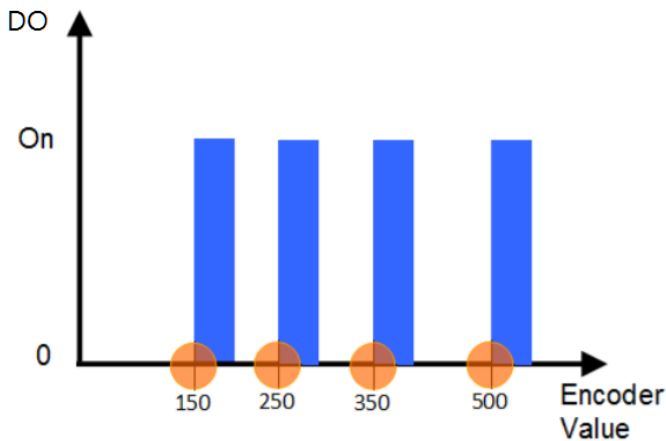
(3) 當 Index 0 的數值被比較過後，才會開始比較 Index 1 的數值，以此類推。

例:將表 6.9 設為陣列比較值

表 6.9

Index	0	1	2	3
Position	150	250	350	500

DO 實際輸出情形:



### 格式:

```
int32_t ECAT_SetSlaveArrCmpPos(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Index, int32_t Position)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號

SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Index	uint8_t	IN	索引值 範圍: 0~199
Position	int32_t	IN	比較的數值

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);
```

## 6.66. ECAT\_GetSlaveArrCmpPos

### 說明:

讀取陣列比較位置。

注: 僅支援 ECAT-2092T。

### 格式:

```
int32_t ECAT_GetSlaveArrCmpPos(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Index, int32_t *Position)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Index	uint8_t	IN	索引值 範圍: 0~199
Position	int32_t*	OUT	比較的數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
uint8_t Index;
```

```
int32_t Position;
```

```
Index = 0;
```

```
ret = ECAT_GetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, &Position);
```

```
if (ret != 0)
```

```
    printf("Failed to get array compare position:%d\n", ret);
```

```
else
```

```
    printf("Index:%u, position:%d \n" , Index, Position);
```

## 6.67. ECAT\_SetSlaveArrCmpEnable

### 說明:

設定陣列比較數位輸出啟用/禁用，啟用陣列比較觸發功能後，會從 index 0 所儲存的比較位置開始比較，當編碼器達到比較位置時，會根據設定而決定是否要觸發數位輸出。

注: (1) 僅支援 ECAT-2092T。

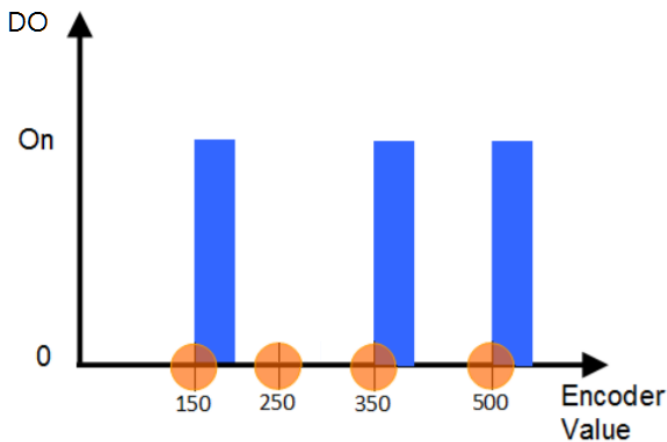
(2) 即使該 Index 的數位輸出被禁用，其比較值還是需要被比較; 例如以下範例，index 1 的數位輸出被禁用，但是 Index 1 的數值被比較過後(不觸發數位輸出)，才會開始比較 Index 2 的數值。

例:將表 6.10 設為陣列比較值及數位輸出設定

表 6.10

Index	0	1	2	3
Position	150	250	350	500
Enable	1	0	1	1

DO 實際輸出情形:



### 格式:

```
int32_t ECAT_SetSlaveArrCmpEnable(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Index, uint8_t Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
----	----	-----------	----

DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Index	uint8_t	IN	索引值 範圍: 0~199
Enable	uint8_t	IN	啟用/禁用 0: 禁用 1: 啟用

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;
uint8_t Enable;

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 0;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
```



```
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);

Index= 1;
Enable= 0;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);

Index= 2;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);

Index= 3;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);
```

---

## 6.68. ECAT\_GetSlaveArrCmpEnable

### 說明:

讀取陣列數位輸出啟用/禁用設定值。

注: 僅支援 ECAT-2092T。

### 格式:

```
int32_t ECAT_GetSlaveArrCmpEnable(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Index, uint8_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
Index	uint8_t	IN	索引值 範圍: 0~199
Enable	uint8_t*	OUT	啟用/禁用 0: 禁用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
uint8_t Index;
```

```
uint8_t Enable;
```

```
Index = 0;
```

```
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, &Enable);
```

```
if (ret != 0)
```

```
    printf("Failed to set array compare enable:%d\n", ret);
```

```
else
```

```
    printf("Index:%u, enable:%u \n" , Index, Enable);
```

## 6.69. ECAT\_SetSlaveArrCmpEndIdx

**說明:**

設定陣列比較範圍, 啟用陣列比較觸發功能後, 會從 index 0 所儲存的比較位置開始比較, 比較至設定的 index 值後, 下一個比較位置將會從 index 0 所儲存的比較位置重新開始比較。

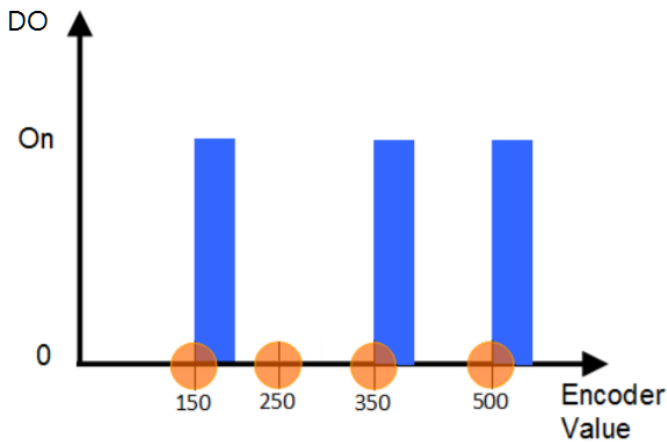
注: 僅支援 ECAT-2092T。

例: 將表 6.11 設為陣列比較值、數位輸出設定及陣列比較範圍

表 6.11

Index	0	1	2	3
Position	150	250	350	500
Enable	1	0	1	1
End index	3			

DO 實際輸出情形:



**格式:**

```
int32_t ECAT_SetSlaveArrCmpEndIdx(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint16_t EndIndex)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號

SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
EndIndex	uint16_t	IN	索引值範圍 範圍: 0~199

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;
uint8_t Enable;
uint16_t End_Index;

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 0;
Enable= 1;
```

---

```
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
Index= 1;  
Enable= 0;  
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
Index= 2;  
Enable= 1;  
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
Index= 3;  
Enable= 1;  
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
End_Index= 3;  
ret = ECAT_SetSlaveArrCmpEndIdx(DeviceNo, SlaveNo, EncNo, End_Index);  
if(ret != 0)  
    printf("Failed to set array compare end index:%d\n", ret);
```

---

## 6.70. ECAT\_GetSlaveArrCmpEndIdx

### 說明:

讀取陣列比較範圍。

注: 僅支援 ECAT-2092T。

### 格式:

```
int32_t ECAT_GetSlaveArrCmpEndIdx(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint16_t *EndIndex)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號
EndIndex	uint16_t*	OUT	索引值範圍 範圍: 0~199

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t EncNo = 0;
```

```
uint16_t End_Index;
```

```
ret = ECAT_GetSlaveArrCmpEndIdx(DeviceNo, SlaveNo, EncNo, &End_Index);
```

```
if(ret != 0)
```

```
    printf("Failed to set array compare end index:%d\n", ret);
```

```
else
```

```
    printf("End Index:%u\n" , End_Index);
```

---

## 6.71. ECAT\_SetSlaveArrCmpTrig

### 說明:

啟用陣列比較觸發功能。

注: (1) 支援 ECAT-2092T。

(2) 需先設定 ECAT\_SetSlaveArrCmpPos、ECAT\_SetSlaveArrCmpEnable 與 ECAT\_SetSlaveArrCmpEndIdx。

### 格式:

```
int32_t ECAT_SetSlaveArrCmpTrig(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
EncNo	uint16_t	IN	編碼器編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;
uint8_t Enable;
uint16_t End_Index;

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 0;
Enable= 1;
```

---

```
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
Index= 1;  
Enable= 0;  
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
Index= 2;  
Enable= 1;  
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
Index= 3;  
Enable= 1;  
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);  
if(ret != 0)  
    printf("Failed to set array compare enable:%d\n", ret);
```

```
End_Index= 3;  
ret = ECAT_SetSlaveArrCmpEndIdx(DeviceNo, SlaveNo, EncNo, End_Index);  
if(ret != 0)  
    printf("Failed to set array compare end index:%d\n", ret);
```

```
ret = ECAT_SetSlaveArrCmpTrig(DeviceNo, SlaveNo, EncNo);  
if(ret != 0)  
    printf("Failed to set array compare trigger:%d\n", ret);
```

---

## 6.72. ECAT\_SetSlaveSaveArrCmpData

### 說明:

將陣列比較值與陣列數位輸出啟用/禁用設定值儲存至該模組的 EEPROM 中，斷電時資料就不會遺失。

注: (1) 支援 ECAT-2092T。

(2) 需要在 PreOP 的狀態下設定，若該模組在 OP 狀態，會返回錯誤。

(3) 需先設定 ECAT\_SetSlaveArrCmpPos 與 ECAT\_SetSlaveArrCmpEnable。

(4) 此函式為阻塞函式，執行後約 10 秒後返回，若回傳值不為零，請將該模組重新上電並且重新設定資料。

### 格式:

```
int32_t ECAT_SetSlaveSaveArrCmpData(uint16_t DeviceNo, uint16_t SlaveNo);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint16_t SlaveNo = 0;  
ret = ECAT_SetSlaveSaveArrCmpData(DeviceNo, SlaveNo);  
if(ret != 0)  
    printf("Failed to save array compare data:%d\n", ret);
```

---

## 6.73. ECAT\_SetTxPdoBufParam

### 說明:

設定一個 TxPdo 緩衝區,此緩衝區會存入最近 PDO\_BUFFER\_DATA\_MAX 個指定 TxPdo 的數值。當讀取緩衝區時,緩衝區內部的資料會被清除。

### 格式:

```
int32_t ECAT_SetTxPdoBufParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint16_t Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_BUFFER_CHANNEL_MAX 個通道
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t	IN	資料大小 ( 最大為 4Bytes )
Enable	uint16_t	IN	0:禁用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 0;
ret = ECAT_SetTxPdoBufParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
    printf("Failed to set TxPdo Buffer parameters:%d\n", ret);
```

---



## 6.74. ECAT\_GetTxPdoBufParam

### 說明:

讀取一個 TxPdo 緩衝區的設定。

### 格式:

```
int32_t ECAT_GetTxPdoBufParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *SlaveNo, uint16_t *OffsetByte, uint16_t *DataSize, uint16_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_BUFFER_CHANNEL_MAX 個通道
SlaveNo	uint16_t*	OUT	從站編號
OffsetByte	uint16_t*	OUT	相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t*	OUT	資料大小 ( 最大為 4Bytes )
Enable	uint16_t*	OUT	0:禁用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 0;
uint16_t Enable= 0;
ret = ECAT_GetTxPdoBufParam(DeviceNo, ChannelNo, &SlaveNo, &OffsetByte, &DataSize, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo Buffer parameters:%d\n", ret);
```

---

## 6.75. ECAT\_SetTxPdoBufEnable

### 說明:

設定一個 TxPdo 緩衝區的啟用/禁用。

### 格式:

int32\_t ECAT\_SetTxPdoBufEnable(uint16\_t DeviceNo, uint16\_t ChannelNo, uint16\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_BUFFER_CHANNEL_MAX 個通道
Enable	uint16_t	IN	0:禁用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint16_t ChannelNo= 0;  
uint16_t Enable= 0;  
ret = ECAT_SetTxPdoBufEnable(DeviceNo, ChannelNo, Enable);  
if(ret != 0)  
    printf("Failed to set TxPdo Buffer Enable:%d\n", ret);
```

---

## 6.76. ECAT\_GetTxPdoBufEnable

### 說明:

讀取一個 TxPdo 緩衝區的啟用/禁用設定。

### 格式:

```
int32_t ECAT_GetTxPdoBufEnable(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_BUFFER_CHANNEL_MAX 個通道
Enable	uint16_t*	OUT	0:禁用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_GetTxPdoBufEnable(DeviceNo, ChannelNo, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo Buffer enable:%d\n", ret);
```

---

## 6.77. ECAT\_GetTxPdoBufValue

### 說明:

讀取一個 TxPdo 緩衝區的資料，此緩衝區會存入最近 PDO\_BUFFER\_DATA\_MAX 個指定 TxPdo 的數值。當讀取緩衝區時，緩衝區內部的資料會被清除。

### 格式:

```
int32_t ECAT_GetTxPdoBufValue(uint16_t DeviceNo, uint16_t ChannelNo, float *Data, uint16_t Size, uint16_t *ActualGetSize)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_BUFFER_CHANNEL_MAX 個通道
Data	float*	OUT	緩衝區中的資料，一次最多可讀取 PDO_BUFFER_DATA_MAX 個資料
Size	uint16_t	IN	要讀取的資料大小
ActualGetSize	uint16_t*	OUT	實際上讀取到的資料數量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 1;
float Data[PDO_BUFFER_DATA_MAX];
uint16_t Size = sizeof( Data );
uint16_t ActualGetSize;

ret = ECAT_SetTxPdoBufParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
    printf("Failed to set TxPdo Buffer parameters:%d\n", ret);

ret = ECAT_GetTxPdoBufValue(DeviceNo, ChannelNo, Data, Size , &ActualGetSize);
if(ret != 0)
    printf("Failed to get TxPdo Buffer Value:%d\n", ret);
else
{
    for(uint16_t i = 0, i < ActualGetSize, i++)
    {
        printf("Data[%u]:%f\n", i, Data[ i ]);
    }
}
```



## 6.78. ECAT\_SetAiFilterParam

### 說明:

設定一個 TxPdo 濾波器，可設定二階點阻濾波器、二階高通濾波器與二階低通濾波器。  
各種濾波器可同時運行。

### 格式:

int32\_t ECAT\_SetAiFilterParam(uint16\_t DeviceNo, uint16\_t ChannelNo, uint16\_t SlaveNo, uint16\_t OffsetByte, uint16\_t DataSize, uint16\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t	IN	資料大小 ( 最大為 4Bytes )
Enable	uint16_t	IN	0:禁用 7:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 7;
ret = ECAT_SetAiFilterParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
    printf("Failed to set TxPdo filter parameters:%d\n", ret);
```

---

## 6.79. ECAT\_GetAiFilterParam

### 說明:

讀取一個 TxPdo 濾波器的設定。

### 格式:

```
int32_t ECAT_GetAiFilterParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *SlaveNo, uint16_t *OffsetByte, uint16_t *DataSize, uint16_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
SlaveNo	uint16_t*	OUT	從站編號
OffsetByte	uint16_t*	OUT	相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t*	OUT	資料大小 ( 最大為 4Bytes )
Enable	uint16_t*	OUT	0:禁用 7:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 0;
uint16_t Enable= 0;
ret = ECAT_GetAiFilterParam(DeviceNo, ChannelNo, &SlaveNo, &OffsetByte, &DataSize, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo filter parameters:%d\n", ret);
```

---

## 6.80. ECAT\_SetAiFilterEnable

### 說明:

設定一個 TxPdo 濾波器的啟用/禁用。

### 格式:

int32\_t ECAT\_SetAiFilterEnable(uint16\_t DeviceNo, uint16\_t ChannelNo, uint16\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
Enable	uint16_t	IN	0:禁用 7:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 7;
ret = ECAT_SetAiFilterEnable(DeviceNo, ChannelNo, Enable);
if(ret != 0)
    printf("Failed to set TxPdo filter Eanble:%d\n", ret);
```

---

## 6.81. ECAT\_GetAiFilterEnable

### 說明:

讀取一個 TxPdo 濾波器的啟用/禁用設定。

### 格式:

```
int32_t ECAT_GetAiFilterEnable(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
Enable	uint16_t*	OUT	0:禁用 7:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_GetAiFilterEnable(DeviceNo, ChannelNo, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo filter enable:%d\n", ret);
```

---



## 6.82. ECAT\_SetAiFilterFreq

### 說明:

設定一個 TxPdo 濾波器的頻率。

### 格式:

int32\_t ECAT\_SetAiFilterFreq(uint16\_t DeviceNo, uint16\_t ChannelNo, uint16\_t FilterType, double Frequency)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
FilterType	uint16_t	IN	1:點阻濾波器 2:低通濾波器 4:高通濾波器
Frequency	float	IN	頻率 點阻濾波器:中心頻率 低通濾波器:截止頻率 高通濾波器:截止頻率 設 0 代表該 FilterType 不使用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t FilterType= FILTER_LOW_PASS; // 2
double Frequency = 60;

ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);

FilterType = FILTER_NOTCH ;//1
Frequency = 0;
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);

FilterType = FILTER_HIGH_PASS;//4
Frequency = 0;
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);
```

---

## 6.83. ECAT\_GetAiFilterFreq

### 說明:

讀取一個 TxPdo 濾波器的頻率。

### 格式:

```
int32_t ECAT_GetAiFilterFreq(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t
FilterType, double *Frequency)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
FilterType	uint16_t	IN	1:點阻濾波器 2:低通濾波器 4:高通濾波器
Frequency	float*	OUT	頻率 點阻濾波器:中心頻率 低通濾波器:截止頻率 高通濾波器:截止頻率

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t FilterType= FILTER_LOW_PASS;
double Frequency;
ret = ECAT_GetAiFilterFreq(DeviceNo, ChannelNo, FilterType, &Frequency);
if(ret != 0)
    printf("Failed to get TxPdo filter Frequency:%d\n", ret);
```

---

## 6.84. ECAT\_GetAiFilterOutput

### 說明:

讀取一個 TxPdo 濾波器的輸出。

### 格式:

```
int32_t ECAT_GetAiFilterOutput(uint16_t DeviceNo, uint16_t ChannelNo, int32_t
*Output)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: AI_FILTER_CHANNEL_MAX 個通道
Output	int32_t*	OUT	濾波器的輸出

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t ChannelNo= 0;
```

```
uint16_t SlaveNo = 0;
```

```
uint16_t OffsetByte= 0;
```

```
uint16_t DataSize= 2;
```

```
uint16_t Enable= 7;
```

```
uint16_t FilterType;
```

```
double Frequency;
```

```
int32_t Output;
```

```
FilterType= FILTER_LOW_PASS;
```

```
Frequency = 60;
```

```
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
```

```
if(ret != 0)
```

```
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);
```

```
FilterType= FILTER_NOTCH;
```

```
Frequency = 0;
```

```
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
```

```
if(ret != 0)
```

```
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);
```

```
FilterType= FILTER_HIGH_PASS;
```

```
Frequency = 0;
```

```
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
```

```
if(ret != 0)
```

```
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);
```

```
ret = ECAT_SetAiFilterParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
```

```
if(ret != 0)
```

```
    printf("Failed to set TxPdo filter parameters:%d\n", ret);
```

```
ret = ECAT_GetAiFilterOutput(DeviceNo, ChannelNo, &Output);
```

```
if(ret != 0)
```

---

```
printf("Failed to set TxPdo filter parameters:%d\n", ret);  
else  
printf("Failed to set TxPdo filter parameters:%d\n", ret);
```

---

## 6.85. ECAT\_SetPdoInToOutParam

### 說明:

設定一個 Pdo 輸入輸出，每個通訊週期會將 TxPDO 資料(Input)寫至 RxPDO(Output)。

### 格式:

```
int32_t ECAT_SetPdoInToOutParam(uint16_t DeviceNo, uint16_t ChannelNo,
uint16_t SlaveNoIn, uint16_t OffsetByteIn, uint16_t DataSizeIn, uint16_t SlaveNoOut,
uint16_t OffsetByteOut, uint16_t DataSizeOut)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_INTTOOUT_CHANNEL_MAX 個通道
SlaveNoIn	uint16_t	IN	從站編號
OffsetByteIn	uint16_t	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSizeIn	uint16_t	IN	資料大小 ( 最大為 4Bytes )
SlaveNoOut	uint16_t	IN	從站編號
OffsetByteOut	uint16_t	IN	設定相對於從站模組 RxPdo Process Image 資料開始位置偏移量
DataSizeOut	uint16_t	IN	資料大小 ( 最大為 4Bytes )

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNoIn = 0;
uint16_t OffsetByteIn= 0;
uint16_t DataSizeIn= 0;
uint16_t SlaveNoOut = 0;
uint16_t OffsetByteOut= 0;
uint16_t DataSizeOut= 0;
ret = ECAT_SetPdInToOutParam(DeviceNo, ChannelNo, SlaveNoIn, OffsetByteIn, DataSizeIn,
SlaveNoOut, OffsetByteOut, DataSizeOut);
if(ret != 0)
    printf("Failed to set PdInToOut parameters:%d\n", ret);
```

---

## 6.86. ECAT\_GetPdoInToOutParam

### 說明:

讀取一個 Pdo 輸入輸出的設定。

### 格式:

```
int32_t ECAT_GetPdoInToOutParam(uint16_t DeviceNo, uint16_t ChannelNo,
uint16_t *SlaveNoIn, uint16_t *OffsetByteIn, uint16_t *DataSizeIn, uint16_t *SlaveNoOut,
uint16_t *OffsetByteOut, uint16_t *DataSizeOut)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_INTTOOUT_CHANNEL_MAX 個通道
SlaveNoIn	uint16_t*	OUT	從站編號
OffsetByteIn	uint16_t*	OUT	相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSizeIn	uint16_t*	OUT	資料大小 ( 最大為 4Bytes )
SlaveNoOut	uint16_t*	OUT	從站編號
OffsetByteOut	uint16_t*	OUT	相對於從站模組 RxPdo Process Image 資料開始位置偏移量
DataSizeOut	uint16_t*	OUT	資料大小 ( 最大為 4Bytes )

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNoIn = 0;
uint16_t OffsetByteIn= 0;
uint16_t DataSizeIn= 0;
uint16_t SlaveNoOut = 0;
uint16_t OffsetByteOut= 0;
uint16_t DataSizeOut= 0;
ret = ECAT_GetPdoInToOut(DeviceNo, ChannelNo, &SlaveNoIn, &OffsetByteIn, &DataSizeIn,
&SlaveNoOut, &OffsetByteOut, &DataSizeOut);
if(ret != 0)
    printf("Failed to get PdoInToOut parameters:%d\n", ret);
```

---

## 6.87. ECAT\_SetPdoInToOutCoeff

### 說明:

設定一個 Pdo 輸入輸出的係數。

輸出 = 輸入 \* gain + offset

### 格式:

int32\_t ECAT\_SetPdoInToOutCoeff(uint16\_t DeviceNo, uint16\_t ChannelNo, float gain, float offset)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_INTTOOUT_CHANNEL_MAX 個通道
gain	float	IN	增益係數
offset	float	IN	偏移係數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
```

```
uint16_t DeviceNo = 0;
```

```
uint16_t ChannelNo= 0;
```

```
float gain= 1;
```

```
float offset = 0;
```

```
ret = ECAT_SetPdoInToOutCoeff(DeviceNo, ChannelNo, gain, offset);
```

```
if(ret != 0)
```

```
    printf("Failed to set PdoInToOut Coeff:%d\n", ret);
```

---

## 6.88. ECAT\_GetPdoInToOutCoeff

### 說明:

讀取一個 Pdo 輸入輸出的係數。

### 格式:

```
int32_t ECAT_GetPdoInToOutCoeff(uint16_t DeviceNo, uint16_t ChannelNo, float
*gain, float *offset)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_INTTOOUT_CHANNEL_MAX 個通道
gain	float*	OUT	增益係數
offset	float*	OUT	偏移係數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
float gain;
float offset;
ret = ECAT_GetPdInToOutCoeff(DeviceNo, ChannelNo, &gain, &offset);
if(ret != 0)
    printf("Failed to get PdInToOut coeff:%d\n", ret);
```

---

## 6.89. ECAT\_SetPdoInToOutEnable

### 說明:

設定一個 Pdo 輸入輸出的啟用/禁用。

### 格式:

```
int32_t ECAT_SetPdoInToOutEnable(uint16_t DeviceNo, uint16_t ChannelNo,
uint16_t Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_INTTOOUT_CHANNEL_MAX 個通道
Enable	uint16_t	IN	0:禁用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint16_t ChannelNo= 0;  
uint16_t Enable= 0;  
ret = ECAT_SetPdInToOutEnable(DeviceNo, ChannelNo, Enable);  
if(ret != 0)  
    printf("Failed to set PdInToOutEanble:%d\n", ret);
```

---

## 6.90. ECAT\_GetPdoInToOutEnable

### 說明:

讀取一個 Pdo 輸入輸出的啟用/禁用設定。

### 格式:

```
int32_t ECAT_GetPdoInToOutEnable(uint16_t DeviceNo, uint16_t ChannelNo,
uint16_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ChannelNo	uint16_t	IN	通道編號 最多可設定: PDO_INTTOOUT_CHANNEL_MAX 個通道
Enable	uint16_t*	OUT	0:禁用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_GetPdInToOutEnable(DeviceNo, ChannelNo, &Enable);
if(ret != 0)
    printf("Failed to get PdInToOut enable:%d\n", ret);
```

---

## 6.91. ECAT\_SlaveNonBlockRegErrReadRequest

### 說明:

向從站模組要求取得錯誤計數器

可取得無效幀計數器(重要)、連接丟失計數器與轉發 CRC 錯誤計數器

當有出現通訊問題時(ECAT\_ERR\_WORKING\_COUNTER) , 可參考"EtherCAT 診斷"

注:此功能僅能在狀態 ECAT\_AS\_OP 時使用

### 格式:

int32\_t ECAT\_SlaveNonBlockRegErrReadRequest(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t Type)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Type	uint16_t	IN	計數器類別 REGISTER_TYPE_ERR(無效幀計數器) REGISTER_TYPE_LOST_LINK (連接丟失計數器) REGISTER_TYPE_FWD_CRC_ERR (轉發 CRC 錯誤計數器)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint8_t RxErr[4];
uint8_t CRCErr[4];
uint8_t LinkLoss[4];
uint8_t FwdCRCErr[4];
uint8_t Dummy[4];
/* REGISTER_TYPE_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErrReadRequest(DeviceNo, SlaveNo, REGISTER_TYPE_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrReadState(DeviceNo, SlaveNo, REGISTER_TYPE_ERR,
            RxErr, CRCErr);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("RxErr Port[0]:%u\n", RxErr[0]);
        printf("RxErr Port[1]:%u\n", RxErr[1]);
        printf("RxErr Port[2]:%u\n", RxErr[2]);
    }
}

```

```
printf("RxErr Port[3]:%u\n", RxErr[3]);

printf("CRCErr Port[0]:%u\n", CRCErr[0]);
printf("CRCErr Port[1]:%u\n", CRCErr[1]);
printf("CRCErr Port[2]:%u\n", CRCErr[2]);
printf("CRCErr Port[3]:%u\n", CRCErr[3]);
}
}
/* REGISTER_TYPE_LOST_LINK*/
do
{
    ret = ECAT_SlaveNonBlockRegErrReadRequest(DeviceNo, SlaveNo, REGISTER_TYPE_LOST_LINK);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrReadState(DeviceNo, SlaveNo,
REGISTER_TYPE_LOST_LINK, LinkLoss, Dummy);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("LinkLoss Port[0]:%u\n", LinkLoss[0]);
        printf("LinkLoss Port[1]:%u\n", LinkLoss[1]);
        printf("LinkLoss Port[2]:%u\n", LinkLoss[2]);
        printf("LinkLoss Port[3]:%u\n", LinkLoss[3]);
    }
}
}
```

---

```
/* REGISTER_TYPE_FWD_CRC_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErrReadRequest(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrReadState(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR, FwdCRCErr, Dummy);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("FwdCRCErr Port[0]:%u\n", FwdCRCErr[0]);
        printf("FwdCRCErr Port[1]:%u\n", FwdCRCErr[1]);
        printf("FwdCRCErr Port[2]:%u\n", FwdCRCErr[2]);
        printf("FwdCRCErr Port[3]:%u\n", FwdCRCErr[3]);
    }
}
}
```

---

## 6.92. ECAT\_SlaveNonBlockRegErrReadState

### 說明:

向從站模組取得錯誤計數器資料

可取得無效幀計數器(重要)、連接丟失計數器與轉發 CRC 錯誤計數器

當有出現通訊問題時(ECAT\_ERR\_WORKING\_COUNTER) , 可參考”EtherCAT 診斷”

注:此功能僅能在 ECAT\_AS\_OP 時使用

### 格式:

int32\_t ECAT\_SlaveNonBlockRegErrReadState(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t Type, uint8\_t \*Value1, uint8\_t \*Value2)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Type	uint16_t	IN	計數器類別 REGISTER_TYPE_ERR(無效幀計數器) REGISTER_TYPE_LOST_LINK (連接丟失計數器) REGISTER_TYPE_FWD_CRC_ERR (轉發 CRC 錯誤計數器)
Value1	uint8_t*	OUT	根據計數器類別，有不同定義 Type 為無效幀計數器時 取得無效幀(Rx 接收)計數器 Type 為連接丟失計數器時 取得連接丟失計數器 Type 為轉發 CRC 錯誤計數器時 取得轉發 CRC 錯誤計數器
Value2	uint8_t*	OUT	根據計數器類別，有不同定義 Type 為無效幀計數器時 取得無效幀(CRC 校驗)計數器



			Type 為連接丟失計數器時 無資料 Type 為轉發 CRC 錯誤計數器時 無資料
--	--	--	---

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint8_t RxErr[4];
uint8_t CRCErr[4];
uint8_t LinkLoss[4];
uint8_t FwdCRCErr[4];
uint8_t Dummy[4];
/* REGISTER_TYPE_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErrReadRequest(DeviceNo, SlaveNo, REGISTER_TYPE_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrReadState(DeviceNo, SlaveNo, REGISTER_TYPE_ERR,
            RxErr, CRCErr);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("RxErr Port[0]:%u\n", RxErr[0]);
        printf("RxErr Port[1]:%u\n", RxErr[1]);
        printf("RxErr Port[2]:%u\n", RxErr[2]);
    }
}

```

```
printf("RxErr Port[3]:%u\n", RxErr[3]);

printf("CRCErr Port[0]:%u\n", CRCErr[0]);
printf("CRCErr Port[1]:%u\n", CRCErr[1]);
printf("CRCErr Port[2]:%u\n", CRCErr[2]);
printf("CRCErr Port[3]:%u\n", CRCErr[3]);
}
}
/* REGISTER_TYPE_LOST_LINK*/
do
{
    ret = ECAT_SlaveNonBlockRegErrReadRequest(DeviceNo, SlaveNo, REGISTER_TYPE_LOST_LINK);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrReadState(DeviceNo, SlaveNo,
REGISTER_TYPE_LOST_LINK, LinkLoss, Dummy);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("LinkLoss Port[0]:%u\n", LinkLoss[0]);
        printf("LinkLoss Port[1]:%u\n", LinkLoss[1]);
        printf("LinkLoss Port[2]:%u\n", LinkLoss[2]);
        printf("LinkLoss Port[3]:%u\n", LinkLoss[3]);
    }
}
}
```

---

```
/* REGISTER_TYPE_FWD_CRC_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErrReadRequest(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrReadState(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR, FwdCRCErr, Dummy);
        }while (ret == -1040); //busy
        if(ret < 0)
        {
            printf("Failed to read:%d\n", ret);
        }
        else
        {
            printf("FwdCRCErr Port[0]:%u\n", FwdCRCErr[0]);
            printf("FwdCRCErr Port[1]:%u\n", FwdCRCErr[1]);
            printf("FwdCRCErr Port[2]:%u\n", FwdCRCErr[2]);
            printf("FwdCRCErr Port[3]:%u\n", FwdCRCErr[3]);
        }
    }
}
```

---

## 6.93. ECAT\_SlaveNonBlockRegErClrRequest

### 說明:

向從站模組要求清除錯誤計數器

無效幀計數器、連接丟失計數器與轉發 CRC 錯誤計數器最大值為 255，超過此值後就不會再計數，建議在初始化階段先清除錯誤計數器

注:此功能僅能在 ECAT\_AS\_OP 時使用

### 格式:

int32\_t ECAT\_SlaveNonBlockRegErClrRequest(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t Type)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Type	uint16_t	IN	計數器類別 REGISTER_TYPE_ERR(無效幀計數器) REGISTER_TYPE_LOST_LINK (連接丟失計數器) REGISTER_TYPE_FWD_CRC_ERR (轉發 CRC 錯誤計數器)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
/* REGISTER_TYPE_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErClrRequest(DeviceNo, SlaveNo, REGISTER_TYPE_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrClrState(DeviceNo, SlaveNo, REGISTER_TYPE_ERR);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("Clear successfully!\n");
    }
}
/* REGISTER_TYPE_LOST_LINK*/
do
{
    ret = ECAT_SlaveNonBlockRegErClrRequest(DeviceNo, SlaveNo, REGISTER_TYPE_LOST_LINK);
}while (ret == -1040); // request busy
if(ret < 0)

```

```
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrClrState(DeviceNo, SlaveNo, REGISTER_TYPE_LOST_LINK);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("Clear successfully!\n");
    }
}
/* REGISTER_TYPE_FWD_CRC_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErClrRequest(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrClrState(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR);
```

---

```
}while (ret == -1040); //busy
if(ret < 0)
{
    printf("Failed to read:%d\n", ret);
}
else
{
    printf("Clear successfully!\n");
}
}
```

---



## 6.94. ECAT\_SlaveNonBlockRegErrClrState

### 說明:

向從站模組確認錯誤計數器是否清除完成

注:此功能僅能在 ECAT\_AS\_OP 時使用

### 格式:

```
int32_t ECAT_SlaveNonBlockRegErrClrState(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t Type)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
Type	uint16_t	IN	計數器類別 REGISTER_TYPE_ERR(無效幀計數器) REGISTER_TYPE_LOST_LINK (連接丟失計數器) REGISTER_TYPE_FWD_CRC_ERR (轉發 CRC 錯誤計數器)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
/* REGISTER_TYPE_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErClrRequest(DeviceNo, SlaveNo, REGISTER_TYPE_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrClrState(DeviceNo, SlaveNo, REGISTER_TYPE_ERR);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("Clear successfully!\n");
    }
}
/* REGISTER_TYPE_LOST_LINK*/
do
{
    ret = ECAT_SlaveNonBlockRegErClrRequest(DeviceNo, SlaveNo, REGISTER_TYPE_LOST_LINK);
}while (ret == -1040); // request busy
if(ret < 0)
```

```
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrClrState(DeviceNo, SlaveNo, REGISTER_TYPE_LOST_LINK);
    }while (ret == -1040); //busy
    if(ret < 0)
    {
        printf("Failed to read:%d\n", ret);
    }
    else
    {
        printf("Clear successfully!\n");
    }
}
/* REGISTER_TYPE_FWD_CRC_ERR */
do
{
    ret = ECAT_SlaveNonBlockRegErClrRequest(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR);
}while (ret == -1040); // request busy
if(ret < 0)
{
    printf("Failed to request:%d\n", ret);
}
else
{
    printf("request successfully!\n");

    do
    {
        ret = ECAT_SlaveNonBlockRegErrClrState(DeviceNo, SlaveNo,
REGISTER_TYPE_FWD_CRC_ERR);
```

---

```
}while (ret == -1040); //busy
if(ret < 0)
{
    printf("Failed to read:%d\n", ret);
}
else
{
    printf("Clear successfully!\n");
}
}
```

---

## 6.95. ECAT\_Get\_2074A

### 說明:

取得從站模組編碼器數值。

注:支援 ECAT-2074A

### 格式:

int32\_t ECAT\_Get\_2074A(uint16\_t DeviceNo, uint16\_t SlaveNo, uint16\_t ChannelNo, int8\_t\* Status, int8\_t\* InputCycleCounter, uint32\_t\* MultiTurn, uint32\_t\* SingleTurn)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從站編號
ChannelNo	uint16_t	IN	編碼器編號
Status	int8_t *	OUT	編碼器狀態 (如表 6.12 所示)
InputCycleCounter	int8_t *	OUT	顯示是否有成功傳輸新的資料，成功讀取資料則次數加 1
MultiTurn	uint32_t *	OUT	顯示編碼器主站讀取到的 Multiturn 位置
SingleTurn	uint32_t *	OUT	顯示編碼器主站讀取到的 Singleturn 位置

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 6.12: 狀態

位元	對應訊號
Bit 0	BiSS Warning bit (在 BiSS-C 模式下編碼器傳輸給編碼器主站的 Warning bit , 此訊號為 active low)
Bit 1	BiSS Error bit (在 BiSS-C 模式下編碼器傳輸給編碼器主站的 Error bit , 此訊號為 active low)
Bit 2	Ready (顯示資料是否已經準備完成)
Bit 3	CRC Error (顯示 CRC 校驗成果是否有誤)
Bit 4~7	保留

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int8_t Status;
int8_t InputCycleCounter;
uint32_t MultiTurn;
uint32_t SingleTurn;

ret = ECAT_Get_2074A (DeviceNo, SlaveNo, EncNo, &Status, &InputCycleCounter, &MultiTurn,
&SingleTurn);
if(ret != 0)
{
    printf("Failed to get ECAT-2074A:%d\n", ret);
}
else
{
    printf("Encoder Status:%u\n", Status);
    printf("Encoder InputCycleCounter:%u\n", InputCycleCounter);
    printf("Encoder MultiTurn:%u, SingleTurn:\n", MultiTurn, SingleTurn);
}

```



## 7. 運動控制函式集說明

### 7.1. 運動控制初始化

#### 7.1.1. ECAT\_McInIt

**說明:**

初始化運動控制軸數，並配置軸號對應的從站編號。

**格式:**

```
int32_t ECAT_McInIt(uint16_t DeviceNo, uint16_t SlaveNo[], uint16_t SubAxisNo[],
uint16_t AxisCount)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t[]	IN	從站編號陣列，陣列的索引編號即為對應的軸號
SubAxisNo	uint16_t[]	IN	從站驅動器子軸號
AxisCount	uint16_t	IN	初始化軸數(最大可初始化軸數定義於標頭檔名稱 MC_AXIS_NO_MAX)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisCount=0;
uint16_t McSlaveNo[MC_AXIS_NO_MAX];
uint16_t McSubAxisNo[MC_AXIS_NO_MAX];
// axis0
McSlaveNo[0] = 0;
McSubAxisNo [0] = 0;
// axis1
McSlaveNo[1] = 1;
McSubAxisNo [1] = 0;

// the following module is 4-axis stepper driver
// axis2
McSlaveNo[2] = 2;
McSubAxisNo [2] = 0;
// axis3
McSlaveNo[3] = 2;
McSubAxisNo [3] = 1;
// axis4
McSlaveNo[4] = 2;
McSubAxisNo [4] = 2;
// axis5
McSlaveNo[5] = 2;
McSubAxisNo [5] = 3;

AxisCount = 6; // total = 1+1+4
ret = ECAT_McInIt(DeviceNo, McSlaveNo, McSubAxisNo, AxisCount);
if(ret < 0)
{
    printf("Failed to initialize motion control:%d\n",ret);
}
else
{
    printf("Initialize motion control successfully\n");
}

```

---

}

---

## 7.1.2. ECAT\_McInit\_Ex

### 說明:

使用軸參數檔初始化運動控制，參數檔的建立請參考"運動控制初始化參數檔編輯步驟"。

### 格式:

```
int32_t ECAT_McInit_Ex(uint16_t DeviceNo, char* FileName, uint16_t SlaveNo[],
uint16_t SubAxisNo[], uint16_t* AxisCnt)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
FileName	char*	IN	軸參數檔名稱
SlaveNo	uint16_t[]	OUT	從站編號陣列, 陣列的索引編號即為對應的軸號
SubAxisNo	uint16_t[]	OUT	從站驅動器子軸號
AxisCnt	uint16_t	OUT	軸數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
char* FileName=" File.motcfg"; // Local path
//char* FileName="C:\\ File.motcfg"; // Absolute path
uint16_t SlaveNo[MC_AXIS_NO_MAX];
uint16_t SubAxisNo[MC_AXIS_NO_MAX];
uint16_t AxisCnt;

ret = ECAT_McInit_Ex(DeviceNo, FileName, SlaveNo, SubAxisNo, &AxisCnt);
if(ret < 0)
{
    printf("Failed to initialize motion control:%d\n",ret);
}
else
{
    uint16_t i;
    printf("Initialize motion control successfully\n");
    printf("Axis Count: %u\n", AxisCnt);
    for(i = 0; i < AxisCnt; i++)
    {
        printf("Axis[%u].SlaveNo: %u\n", i, SlaveNo[i]);
    }
}
}
```

## 7.2. 單軸運動控制設定

### 7.2.1. ECAT\_McSetAxisDefaultMode

#### 說明:

設定指定軸號的預設運動模式。

初始的預設運動模式為 CSP 模式，在 ServoOn 以及 ErrorReset 時會將運動模式切換為預設運動模式，表 7.1 為各運動模式所需要的 PDO Entries。

表 7.1

	RxPdo Entries		TxPdo Entries	
MC_MODE_CSP	607A	Target Position	6064	Position actual value
MC_MODE_CSV	60FF	Target Velocity	6064	Position actual value
MC_MODE_CST	6071	Target Torque	6077	Torque actual value

#### 格式:

```
int32_t ECAT_McSetAxisDefaultMode(uint16_t DeviceNo, uint16_t AxisNo, int8_t Mode)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Mode	int8_t	IN	運動模式 MC_MODE_CSP(CSP 模式) MC_MODE_CSV(CSV 模式) MC_MODE_CST(CST 模式) 預設值: MC_MODE_CSP(CSP 模式)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int8_t Mode = MC_MODE_CSP;
ret = ECAT_McSetAxisDefaultMode(DeviceNo, AxisNo, Mode);
if(ret < 0)
{
    printf("Failed to set axis default mode:%d\n",ret);
}
else
{
    printf("Set axis default mode successfully!\n");
}
```

---

## 7.2.2. ECAT\_McGetAxisDefaultMode

### 說明:

取得指定軸號的預設運動模式。

### 格式:

```
int32_t ECAT_McGetAxisDefaultMode(uint16_t DeviceNo, uint16_t AxisNo, int8_t *Mode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Mode	int8_t *	OUT	運動模式

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int8_t Mode;
ret = ECAT_McGetAxisDefaultMode(DeviceNo, AxisNo, &Mode);
if(ret < 0)
{
    printf("Failed to get axis default mode:%d\n", ret);
}
else
{
    printf("Axis[%u] default mode:%f\n", AxisNo, Mode);
}
```

---



### 7.2.3. ECAT\_McSetAxisServoOn

#### 說明:

設定指定軸號伺服啟動訊號的狀態。

設定 Servo On 之後,需要幾個 ms 才會完成 Servo On,可以使用 ECAT\_McGetAxisState 取得軸狀態,當軸狀態為 MC\_AS\_STANDSTILL 代表 Servo On 完成

#### 格式:

```
int32_t ECAT_McSetAxisServoOn(uint16_t DeviceNo, uint16_t AxisNo, uint16_t State)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
State	uint16_t	IN	訊號狀態 0: OFF 1: ON

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t State = 1;
ret = ECAT_McSetAxisServoOn(DeviceNo, AxisNo, State);
if(ret < 0)
{
    printf("Failed to set axis ServoOn:%d\n",ret);
}
else
{
    printf("Set axis ServoOn successfully!\n");
}
```

---

## 7.2.4. ECAT\_McSetAxisPPU

### 說明:

設定指定軸號 Pulse per Unit 參數。

Pulses Per Unit，每一單位為幾個 pulses。若要將單位設為 mm，而每運動 1mm 需要 1000 個 pulse，那就將 PPU 設為 1000，如此一來，“user unit”就為 mm。需要注意的是，表 5.1 內的所有 PDO Entry 的單位都需要是 pulse 才可以自定義 PPU；而當驅動器的 PDO Entry(表 7.2)單位不為 pulse、pulse/s 以及 pulse/s<sup>2</sup> 時，需要將 PPU 設為 1，此時“user unit”就為該 PDO 的原始單位，如 rpm、rpm/s... ..等。

表 7.2

RxPDO		TxPDO		單位
607A	Target Position	6064	Position actual value	pulse
60FF	Target Velocity	606C	Velocity actual value	pulse/s
60B0	Position offset			pulse
60B1	Velocity offset			pulse/s
6099	Speed during search for switch			pulse/s
6099	Speed during search for zero			pulse/s
609A	Homing acceleration			pulse/s <sup>2</sup>

### 格式:

int32\_t ECAT\_McSetAxisPPU(uint16\_t DeviceNo, uint16\_t AxisNo, double PPU)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
PPU	double	IN	Pulse per Unit 參數 default:1

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double PPU = 100000;
ret = ECAT_McSetAxisPPU(DeviceNo, AxisNo, PPU);
if(ret < 0)
{
    printf("Failed to set axis PPU:%d\n",ret);
}
else
{
    printf("Set axis PPU successfully!\n");
}
```

---

## 7.2.5. ECAT\_McGetAxisPPU

### 說明:

取得指定軸號 Pulse per Unit 參數。

### 格式:

```
int32_t ECAT_McGetAxisPPU(uint16_t DeviceNo, uint16_t AxisNo, double *PPU)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
PPU	Double*	OUT	Pulse per Unit 參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double PPU;
ret = ECAT_McGetAxisPPU(DeviceNo, AxisNo, &PPU);
if(ret < 0)
{
    printf("Failed to get axis PPU:%d\n", ret);
}
else
{
    printf("Axis[%u] PPU:%f\n", AxisNo, PPU);
}
```

---

## 7.2.6. ECAT\_McSetAxisVelAccScale

### 說明:

設定指定軸號速度/加速度換算參數。

當驅動器的 PDO Entry(表 7.2)單位不為 pulse/s 或是 pulse/s<sup>2</sup> 時，可以使用此參數進行單位換算。

### 例:

PDO 原始速度單位為 rpm，要將 pulse/s 轉換為 rpm，公式為(1/1 圈所需的 pulse 數) \* 60，假設 1 圈所需要的 pulse 數為 10000，則速度換算參數就設定為 1/10000\*60 = 0.006

PDO 原始加速度單位為 rpm/s，要將 pulse/s<sup>2</sup> 轉換為 rpm/s，公式為(1/1 圈所需的 pulse 數) \* 60，假設 1 圈所需要的 pulse 數為 10000，則加速度換算參數就設定為 1/10000\*60 = 0.006

### 格式:

int32\_t ECAT\_McSetAxisVelAccScale(uint16\_t DeviceNo, uint16\_t AxisNo, double VelScal, double AccScal)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
VelScal	double	IN	速度換算參數 default: 1
AccScal	double	IN	加速度換算參數 default: 1

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double VelScal = 1;
double AccScal = 1;
ret = ECAT_McSetAxisVelAccScale(DeviceNo, AxisNo, VelScal, AccScal);
if(ret < 0)
{
    printf("Failed to set axis vel acc scale:%d\n",ret);
}
else
{
    printf("Set axis vel acc scale successfully!\n");
}
```

---

## 7.2.7. ECAT\_McGetAxisVelAccScale

### 說明:

取得指定軸號速度/加速度換算參數。

### 格式:

```
int32_t ECAT_McGetAxisVelAccScale(uint16_t DeviceNo, uint16_t AxisNo, double *VelScal, double *AccScal)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
VelScal	double*	OUT	速度換算參數
AccScal	double*	OUT	加速度換算參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double VelScal;
double AccScal;
ret = ECAT_McGetAxisVelAccScale(DeviceNo, AxisNo, &VelScal, &AccScal);
if(ret < 0)
{
    printf("Failed to get axis vel acc scale:%d\n", ret);
}
else
{
    printf("Axis[%u] VelScal:%f AccScal:%f, \n", AxisNo, VelScal, AccScal);
}
```

---

## 7.2.8. ECAT\_McSetProfileData

### 說明:

設定指定 Profile 編號位置資料,用於執行 Profile 運動控制(*ECAT\_McAxisMoveProfile*)。

### 格式:

```
int32_t ECAT_McSetProfileData(uint16_t DeviceNo, uint16_t ProfileNo, double *Data
, uint16_t DataSize)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ProfileNo	uint16_t	IN	Profile 編號, 可用編號範圍 0~15
Data	double*	IN	位置資料, 最大可儲存 3000 筆 double 資料
DataSize	uint16_t	IN	設定資料的數量大小

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
double Data[MC_PROFILE_DATA_MAX];
uint16_t DataSize = 10;
Data[0] = 0.00005
Data[1] = 0.00015
Data[2] = 0.00030
Data[3] = 0.00050
Data[4] = 0.00075
Data[5] = 0.00105
Data[6] = 0.00140
Data[7] = 0.00180
Data[8] = 0.00225
Data[9] = 0.00275
ret = ECAT_McSetProfileData(DeviceNo, ProfileNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to set profile data:%d\n",ret);
}
else
{
    printf("Set set profile data successfully!\n");
}
```

## 7.2.9. ECAT\_McGetProfileData

### 說明:

取得指定 Profile 編號位置資料。

### 格式:

```
int32_t ECAT_McGetProfileData(uint16_t DeviceNo, uint16_t ProfileNo, double *Data,
uint16_t DataSize)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15
Data	double*	OUT	位置資料，最大可儲存 3000 筆 double 資料
DataSize	uint16_t	IN	取得資料的數量大小

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t i;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
double Data[MC_PROFILE_DATA_MAX];
uint16_t DataSize = 10;
ret = ECAT_McGetProfileData(DeviceNo, ProfileNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to get profile data:%d\n", ret);
}
else
{
    printf("Set get profile data successfully!\n");
    for(i=0;i<DataSize;i++)
        printf("Data[%u]:%f\n", i, Data[i]);
}
```

---

## 7.2.10. ECAT\_McSetProfileInterval

### 說明:

設定指定 Profile 編號位置資料間隔數。

### 格式:

int32\_t ECAT\_McSetProfileInterval(uint16\_t DeviceNo, uint16\_t ProfileNo, uint16\_t Value)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15
Value	uint16_t	IN	位置資料間隔 default :1 每個 cycletime 讀一筆位置資料 2: 每兩個 cycletime 讀一筆位置資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



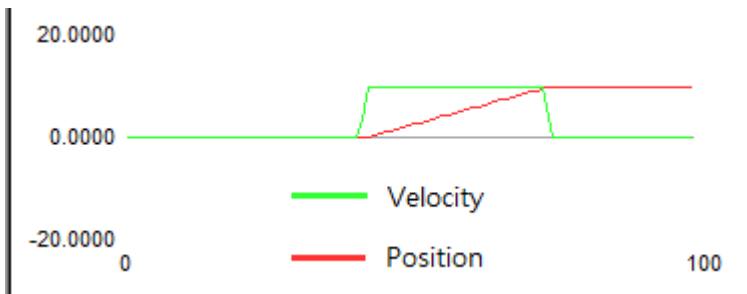
使用範例:

[C/C++]

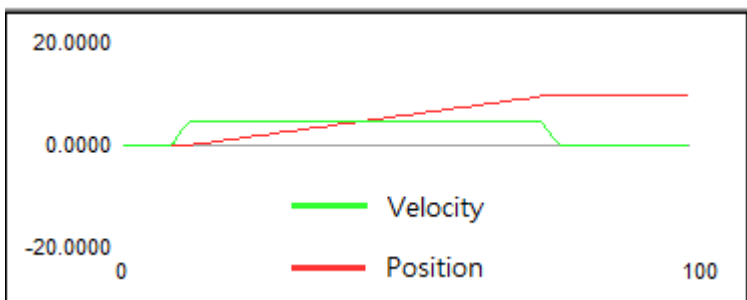
```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ProfileNo = 0;
uint16_t Value = 2;
ret = ECAT_McSetProfileInterval(DeviceNo, ProfileNo, Value);
if(ret < 0)
{
    printf("Failed to set profile Interval:%d\n", ret);
}
else
{
    printf("Set profile Interval successfully!\n");
}
    
```

Interval = 1 default 速度 100%



Interval = 2 速度 50%



## 7.2.11. ECAT\_McSetProfileCSV

### 說明:

設定指定 Profile 編號 CSV 檔位置資料，格式如圖 7.1 所示。資料可以分批傳入控制卡，如果一次傳入大量資料有可能會短暫影響 PC 的其他程式運行。用 LastFlag 控制傳輸，分多次傳輸時，只有是最後一次傳輸的 LastFlag = 1，其餘要設為 0。

### 格式:

```
int32_t ECAT_McSetProfileCSV(uint16_t DeviceNo, uint16_t ProfileNo, uint32_t Offset,
char *Data, uint32_t DataSize, uint8_t LastFlag)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15
Offset	uint32_t	IN	檔案寫入偏移量
Data	char *	IN	寫入資料
DataSize	uint32_t	IN	寫入資料的大小
LastFlag	uint8_t	IN	是否包含最後一筆資料的旗標 0: 後面還有資料，不是最後一筆 1: 此次傳輸是最後一次

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

	Axis 0 Position	Axis 1 Position	Axis 2 Position	
1	0.000013	0.000027	0.000040	First line, axis positions
2	0.000040	0.000080	0.000120	Second line, axis positions
3	0.000080	0.000160	0.000241	
4	0.000134	0.000267	0.000401	
5	0.000200	0.000401	0.000601	
6	0.000281	0.000561	0.000842	
7	0.000374	0.000748	0.001123	
8	0.000481	0.000962	0.001443	
9	0.000601	0.001203	0.001804	
10	0.000735	0.001470	0.002205	
11	0.000882	0.001764	0.002646	
12	0.001042	0.002085	0.003127	

圖 7.1

**使用範例:**

**[C/C++]**

```

FILE *pFile;
size_t file_Size;
char *buffer;
size_t result;
int32_t ret;
uint16_t ProfileNo = 0;
uint8_t LastFlag = 1;
char *file_name = "D:\xxx.csv"

pFile = fopen(file_name, "rb" );
if (pFile==NULL) {
    printf("Failed to open file:%s", file_name);
    return;
}

// obtain file size:
fseek (pFile, 0, SEEK_END);
file_Size = ftell(pFile);
fseek (pFile, 0, SEEK_SET);
    
```

---

```
// allocate memory to contain the whole file:
buffer = (char*)malloc(sizeof(char)*file_Size);
if (buffer == NULL) {
    printf("Failed to allocate memory");
    fclose(pFile);
    return;
}

// copy the file into the buffer:
result = fread(buffer, 1, file_Size, pFile);
if (result != file_Size) {
    printf("Failed to read from file");
    goto out_close;
}

/* the whole file is now loaded in the memory buffer. */
ret = ECAT_McSetProfileCSV(DeviceNo, ProfileNo, 0, buffer, file_Size, LastFlag);
if(ret != 0)
    printf("Failed to set profile csv data:%d", ret);

out_close:
    fclose(pFile);
    free(buffer);
```

---

## 7.2.12. ECAT\_McGetProfileCSV

### 說明:

取得指定 Profile 編號 CSV 檔位置資料，格式如圖 7.1 所示。資料可以分批由控制卡內部讀取，用 LastFlag 傳回資料傳輸是否已到結尾。LastFlag = 1 表示已經是到檔案最後位置，否則 LastFlag 會回傳 0。雖然可以設定一次讀回大量資料，但是有可能影響 PC 程式運行，建議 DataSize 設為 512 或以下。

### 格式:

```
int32_t ECAT_McGetProfileCSV(uint16_t DeviceNo, uint16_t ProfileNo, uint32_t *Offset,
char *Data, uint32_t *DataSize, uint8_t *LastFlag)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15
Offset	uint32_t *	OUT	檔案讀取偏移量
Data	char *	OUT	讀取資料
DataSize	uint32_t *	OUT	讀取資料的大小
LastFlag	uint8_t *	OUT	最後一筆讀取資料旗標 0: 後面還有資料可讀 1: 已達檔案結束位置

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

FILE * pFile;
char Data[2048];
int32_t ret;
char *file_name = "D:\xxx.csv"
uint16_t ProfileNo = 0;
uint8_t LastFlag;
uint32_t DataSize;
uint32_t Offset = 0;

pFile = fopen(file_name, "wb" );
if (pFile==NULL) {
    printf("Failed to create file:%s", file_name);
    return;
}

while(1)
{
    DataSize = 2048;
    LastFlag = 0;
    if((ret = ECAT_McGetProfileCSV(DeviceNo, ProfileNo, &Offset, Data,
    &DataSize, &LastFlag)) != 0) {
        printf("Failed to get profile csv data:%d", ret);
        fclose(pFile);
        return;
    }

    if (fwrite(Data , 1, DataSize, pFile) != DataSize) {
        printf("Failed to Write File");
        fclose(pFile);
        return;
    }

    if(LastFlag) {
        fclose(pFile);
        break;
    }
}

```

---

}

};

---

## 7.2.13. ECAT\_McSetAxisAccDecUnit

### 說明:

設定指定軸號執行單軸運動時使用的加速度單位。

設定單位為 MC\_ACC\_DEC\_MODE\_RATE 時，使用 ECAT\_McSetAxisAccDecRate 的設定值

設定單位為 MC\_ACC\_DEC\_MODE\_TIME 時，使用 ECAT\_McSetAxisAccTime 的設定值

### 格式:

```
int32_t ECAT_McSetAxisAccDecUnit(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Unit)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Unit	uint16_t	IN	執行單軸運動時使用的加速度單位 default: MC_ACC_DEC_MODE_TIME

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



表 7.3: 加速度單位

標頭檔定義名稱	數值	說明
MC_ACC_DEC_MODE_RATE	0	使用 ECAT_McSetAxisAccDecRate 的設定值
MC_ACC_DEC_MODE_TIME	1	使用 ECAT_McSetAxisAccTime 的設定值

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Unit = MC_ACC_DEC_MODE_RATE;
ret = ECAT_McSetAxisAccDecUnit(DeviceNo, AxisNo, Unit);
if(ret < 0)
{
    printf("Failed to set axis acceleration unit:%d\n", ret);
}
else
{
    printf("Set axis acceleration unit successfully!\n");
}

```

## 7.2.14. ECAT\_McGetAxisAccDecUnit

### 說明:

取得指定軸號執行單軸運動時使用的加速度單位設定。

### 格式:

```
int32_t ECAT_McGetAxisAccTime(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Unit)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Unit	uint16_t*	OUT	執行單軸運動時使用的加速度單位

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

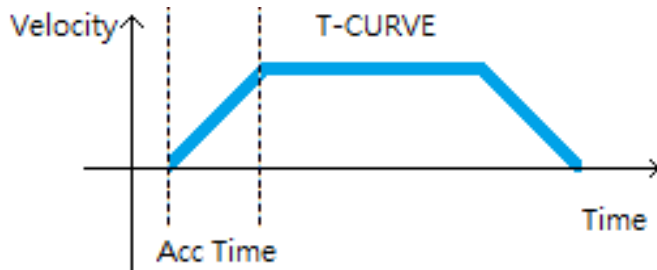
```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Unit;
ret = ECAT_McGetAxisAccTime(DeviceNo, AxisNo, &Unit);
if(ret < 0)
{
    printf("Failed to get axis acceleration unit:%d\n", ret);
}
else
{
    printf("Axis[%u] Acceleration unit:%f\n", AxisNo, Unit);
}
```

---

## 7.2.15. ECAT\_McSetAxisAccTime

### 說明:

設定指定軸號執行單軸運動時使用的加速度時間。時間愈短則加速度愈大。使用線性(梯形)加減速時會使用此函式所設定時間做加速或是減速；如果是 S 型加減速類型，則真正使用時間會變成此值的兩倍。



### 格式:

```
int32_t ECAT_McSetAxisAccTime(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Time_ms)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_ms	uint16_t	IN	執行單軸運動時使用的加速度時間 (單位: 毫秒) default:100

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Time_ms = 500;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
}
else
{
    printf("Set axis acceleration time successfully!\n");
}
```

---

## 7.2.16. ECAT\_McGetAxisAccTime

### 說明:

取得指定軸號執行單軸運動時使用的加速度時間設定。使用線性(梯形)加減速時會使用此函式所得的時間做加速或是減速；如果是 S 型加減速類型，則真正使用時間會變成此值的兩倍。

### 格式:

```
int32_t ECAT_McGetAxisAccTime(uint16_t DeviceNo, uint16_t AxisNo, uint16_t
*Time_ms)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_ms	uint16_t*	OUT	執行單軸運動時使用的加速度時間 (單位: 毫秒)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Time_ms;
ret = ECAT_McGetAxisAccTime(DeviceNo, AxisNo, &Time_ms);
if(ret < 0)
{
    printf("Failed to get axis acceleration time:%d\n", ret);
}
else
{
    printf("Axis[%u] Acceleration Time(ms):%f\n", AxisNo, Time_ms);
}
```

---

## 7.2.17. ECAT\_McSetAxisAccDecRate

### 說明:

設定指定軸號執行單軸運動時使用的加速度。

### 格式:

int32\_t ECAT\_McSetAxisAccDecRate(uint16\_t DeviceNo, uint16\_t AxisNo, double Rate)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Rate	uint16_t	IN	執行單軸運動時使用的加速度 (單位: user unit / S ^ 2) default:10000

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t rate = 50000;
ret = ECAT_McSetAxisAccDecRate(DeviceNo, AxisNo, rate);
if(ret < 0)
{
    printf("Failed to set axis acceleration:%d\n", ret);
}
else
{
    printf("Set axis acceleration successfully!\n");
}
```

---

## 7.2.18. ECAT\_McGetAxisAccDecRate

### 說明:

取得指定軸號執行單軸運動時使用的加速度設定。

### 格式:

```
int32_t ECAT_McGetAxisAccDecRate(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Rate)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Rate	uint16_t*	OUT	執行單軸運動時使用的加速度時間 (單位: user unit / S ^ 2)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Rate;
ret = ECAT_McGetAxisAccDecRate(DeviceNo, AxisNo, &Rate);
if(ret < 0)
{
    printf("Failed to get axis acceleration rate:%d\n", ret);
}
else
{
    printf("Axis[%u] Acceleration rate:%f\n", AxisNo, Rate);
}
```

---

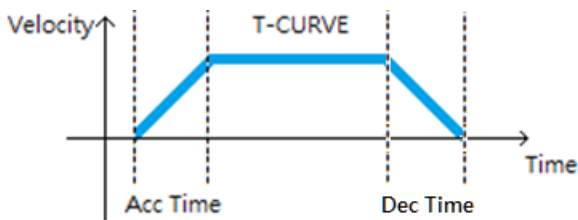
## 7.2.19. ECAT\_McSetAxisAccDecTime\_Stepper

### 說明:

設定指定軸號執行單軸運動時使用的加減速時間。時間愈短則加速度愈大。使用線性(梯形)加減速時會使用此函式所設定時間做加速或是減速。

注: (1)此函式支援 ECAT-2094S、ECAT-2091S

(2)若加減速為相同數值，可以使用 ECAT\_McSetAxisAccTime



### 格式:

```
int32_t ECAT_McSetAxisAccDecTime_Stepper(uint16_t DeviceNo, uint16_t AxisNo,
uint16_t AccTime_ms, uint16_t DecTime_ms)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
AccTime_ms	uint16_t	IN	執行單軸運動時使用的加速度時間 (單位: 毫秒) default:100
DecTime_ms	uint16_t	IN	執行單軸運動時使用的減速度時間 (單位: 毫秒) default:100

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t AccTime_ms = 500;
uint16_t DecTime_ms = 500;
ret = ECAT_McSetAxisAccDecTime_Stepper(DeviceNo, AxisNo, AccTime_ms , DecTime_ms);
if(ret < 0)
{
    printf("Failed to set axis acc dec time:%d\n", ret);
}
else
{
    printf("Set axis acc dec time successfully!\n");
}
```

---

## 7.2.20. ECAT\_McGetAxisAccDecTime\_Stepper

### 說明:

取得指定軸號執行單軸運動時使用的加減度時間設定。

注: 此函式支援 ECAT-2094S、ECAT-2091S

### 格式:

```
int32_t ECAT_McGetAxisAccDecTime_Stepper(uint16_t DeviceNo, uint16_t AxisNo,
uint16_t *AccTime_ms, uint16_t *DecTime_ms);
```

大

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
AccTime_ms	uint16_t*	OUT	執行單軸運動時使用的加速度時間 (單位: 毫秒)
DecTime_ms	uint16_t*	OUT	執行單軸運動時使用的減速度時間 (單位: 毫秒)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t AccTime_ms;
uint16_t DecTime_ms;
ret = McGetAxisAccDecTime_Stepper(DeviceNo, AxisNo, &AccTime_ms, &DecTime_ms);
if(ret < 0)
{
    printf("Failed to get axis acc dec time:%d\n", ret);
}
else
{
    printf("Axis[%u] acc dec Time(ms):%f, %f\n", AxisNo, AccTime_ms, DecTime_ms);
}
```

---

## 7.2.21. ECAT\_McSetAxisAccUnit\_Stepper

### 說明:

設定指定軸號執行單軸運動時使用的加減速單位，此設定只是為了將加減速單位正確的轉換為毫秒。

注: (1)此函式支援 ECAT-2094S、ECAT-2091S

(2)不論此值設定為多少，使用 ECAT\_McSetAxisAccDecTime\_Stepper 或 ECAT\_McSetAxisAccTime 設定加減速時，單位為毫秒

(3)除非必要請不要設定此值

(4)此設定會修改 8x20:08 的值

8020:0	POS Settings X		> 9 <
8020:01	Velocity min	RW	0x00000000 (0)
8020:02	Velocity max	RW	0x000F4240 (1000000)
8020:03	Acceleration pos	RW	0x03E8 (1000)
8020:04	Acceleration neg	RW	0x03E8 (1000)
8020:05	Deceleration pos	RW	0x03E8 (1000)
8020:06	Deceleration neg	RW	0x03E8 (1000)
8020:07	Emergency deceleration	RW	0x0000 (0)
8020:08	Acceleration unit	RW	Acceleration time from Vmin to Vmax [ms] (0)
8020:09	Acc-Dec parameter definition	RW	Acceleration->Start phase & Deceleration->Stop Phase (0)

### 格式:

int32\_t ECAT\_McSetAxisAccUnit\_Stepper(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t Unit, uint32\_t \*AbortCode)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Unit	uint16_t	IN	加減速單位
AbortCode	uint32_t *	OUT	SDO 操作錯誤終止代碼(請參考附錄 "SDO 操作錯誤終止代碼"說明)

### 回傳值:

0: 函式執行成功。



其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Unit = 0;
uint32_t AbortCode;
ret = ECAT_McSetAxisAccUnit_Stepper(DeviceNo, AxisNo, Unit, &AbortCode);
if(ret < 0)
{
    printf("Failed to set axis acc unit:%d\n", ret);
}
else
{
    printf("Set axis acc unit successfully!\n");
}
```

---

## 7.2.22. ECAT\_McGetAxisAccUnit\_Stepper

### 說明:

取得指定軸號執行單軸運動時使用的加減速單位。

注: (1)此函式支援 ECAT-2094S、ECAT-2091S

8020:0	POS Settings X		> 9 <
8020:01	Velocity min	RW	0x00000000 (0)
8020:02	Velocity max	RW	0x000F4240 (1000000)
8020:03	Acceleration pos	RW	0x03E8 (1000)
8020:04	Acceleration neg	RW	0x03E8 (1000)
8020:05	Deceleration pos	RW	0x03E8 (1000)
8020:06	Deceleration neg	RW	0x03E8 (1000)
8020:07	Emergency deceleration	RW	0x0000 (0)
8020:08	Acceleration unit	RW	Acceleration time from Vmin to Vmax [ms] (0)
8020:09	Acc-Dec parameter definition	RW	Acceleration-->Start phase & Deceleration-->Stop Phase (0)

### 格式:

```
int32_t ECAT_McGetAxisAccUnit_Stepper(uint16_t DeviceNo, uint16_t AxisNo, uint16_t
*Unit, uint32_t *AbortCode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Unit	uint16_t*	OUT	加減速單位
AbortCode	uint32_t*	OUT	SDO 操作錯誤終止代碼(請參考附錄 "SDO 操作錯誤終止代碼"說明)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

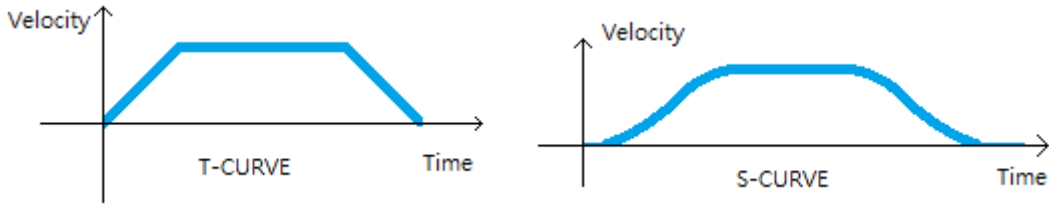
```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Unit;
uint32_t AbortCode;
ret = ECAT_McGetAxisAccUnit_Stepper(DeviceNo, AxisNo, &Unit, &AbortCode);
if(ret < 0)
{
    printf("Failed to get axis acc unit:%d\n", ret);
}
else
{
    printf("Axis[%u] acc unit:%u\n", AxisNo, Unit);
}
```

---

### 7.2.23. ECAT\_McSetAxisAccDecType

**說明:**

設定指定軸號執行單軸運動時使用的加速度類型。



**格式:**

int32\_t ECAT\_McSetAxisAccDecType(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t Type)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Type	uint16_t	IN	執行單軸運動時使用的加速度類型 1:T-Curve (線性) (Default) 2:S-Curve

**回傳值:**

- 0: 函式執行成功。
- 其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Type = 1; //T-Curve
ret = ECAT_McSetAxisAccDecType (DeviceNo, AxisNo, Type);
if(ret < 0)
{
    printf("Failed to set axis AccDecType:%d\n", ret);
}
else
{
    printf("Set axis AccDecType successfully!\n");
}
```

---

## 7.2.24. ECAT\_McGetAxisAccDecType

### 說明:

取得指定軸號執行單軸運動時使用的加速度類型。

### 格式:

```
int32_t ECAT_McGetAxisAxisAccDecType(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Type)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Type	uint16_t*	OUT	執行單軸運動時使用的加速度類型 1: T-Curve (線性) 2: S-Curve

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Type;
ret = ECAT_McGetAxisAccDecType(DeviceNo, AxisNo, &Type);
if(ret < 0)
{
    printf("Failed to get axis AccDecType:%d\n",ret);
}
else
{
    printf("Axis[%u] AccDecType:%f\n", AxisNo, Type);
}
```

---



## 7.2.25. ECAT\_McSetAxisEncoderPPR

### 說明:

設定指定軸號編碼器單圈脈波數。

注: (1)支援 Encorder 模組 / 步進馬達控制器

(2)不支援符合 CiA402 的產品

### 格式:

int32\_t ECAT\_McSetAxisEncoderPPR(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t PPR)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
PPR	Uint32_t	IN	編碼器單圈脈波 (單位: Pulses/Revolution) default:1

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR = 4000;
ret = ECAT_McSetAxisEncoderPPR(DeviceNo, AxisNo, PPR);
if(ret < 0)
{
    printf("Failed to set axis encoder PPR:%d\n", ret);
}
else
{
    printf("Set axis encoder PPR successfully!\n");
}
```

---

## 7.2.26. ECAT\_McGetAxisEncoderPPR

### 說明:

取得指定軸號編碼器單圈脈波設定值。

注: (1)支援 Encorder 模組 / 步進馬達控制器

(2)不支援符合 CiA402 的產品

### 格式:

int32\_t ECAT\_McGetAxisEncoderPPR(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t \*PPR)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
PPR	Uint32_t*	OUT	編碼器單圈脈波 (單位: Pulses/Revolution)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR ;
ret = ECAT_McGetAxisEncoderPPR(DeviceNo, AxisNo, &PPR);
if(ret < 0)
{
    printf("Failed to get axis encoder PPR:%d\n",ret);
}
else
{
    printf("Axis[%u] encoder PPR :%f\n", AxisNo, PPR);
}
```

---

## 7.2.27. ECAT\_McSetAxisMotorPPR

### 說明:

設定指定軸號馬達單圈脈波數。

注: (1)支援 Encorder 模組 / 步進馬達控制器

(2)不支援符合 CiA402 的產品

### 格式:

int32\_t ECAT\_McSetAxisMotorPPR(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t PPR)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
PPR	Uint32_t	IN	馬達單圈脈波數 (單位: Pulses/Revolution) default:1

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR = 4000;
ret = ECAT_McSetAxisMotorPPR(DeviceNo, AxisNo, PPR);
if(ret < 0)
{
    printf("Failed to set axis motor PPR:%d\n", ret);
}
else
{
    printf("Set axis motor PPR successfully!\n");
}
```

---

## 7.2.28. ECAT\_McGetAxisMotorPPR

### 說明:

取得指定軸號馬達單圈脈波數設定值。

注: (1)支援 Encorder 模組 / 步進馬達控制器

(2)不支援符合 CiA402 的產品

### 格式:

int32\_t ECAT\_McGetAxisMotorPPR(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t \*PPR)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
PPR	Uint32_t*	OUT	馬達單圈脈波數 (單位: Pulses/Revolution)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR ;
ret = ECAT_McGetAxisMotorPPR(DeviceNo, AxisNo, &PPR);
if(ret < 0)
{
    printf("Failed to get axis motor PPR:%d\n", ret);
}
else
{
    printf("Axis[%u] motor PPR :%f\n", AxisNo, PPR);
}
```

---



## 7.2.29. ECAT\_McSetEcamTable

### 說明:

設定 E-CAM 表格，電子凸輪跟隨運動控制使用(*ECAT\_McAxisCamIn*)。  
 指定一個 E-CAM 表格編號，並將資料填入。  
 可以使用 Cam Utility 來產生 E-CAM 表格

軟體及手冊的下載請參考”軟體安裝”章節

### 格式:

```
int32_t ECAT_McSetEcamTable(uint16_t DeviceNo, uint16_t TableNo, double *Data,
uint16_t DataSize)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
TableNo	uint16_t	IN	E-CAM 表格編號
Data	double*	IN	從站位置資料，共可設定 1000 筆資料 (單位:user unit)
DataSize	uint16_t	IN	寫入的資料筆數

### 回傳值:

0: 函式執行成功。  
 其他: 請參考附錄”函式錯誤回傳代碼”說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t TableNo = 0;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[3];
uint16_t DataSize = 3;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.5;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;

//Write E-CAM Table data to Data[3]
Data[0] = 0;
Data[1] = 1;
Data[2] = 0;

ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to set E-CAM table data:%d\n", ret);
    return;
}

ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);
if(ret < 0)
{
    printf("Failed to configure E-CAM table parameter:%d\n", ret);
    return;
}

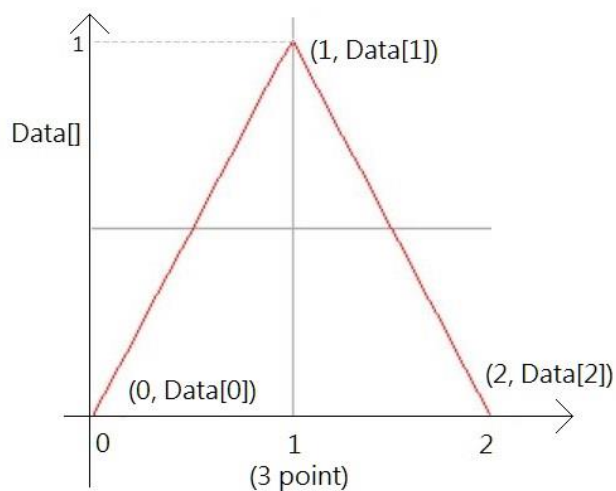
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill

```

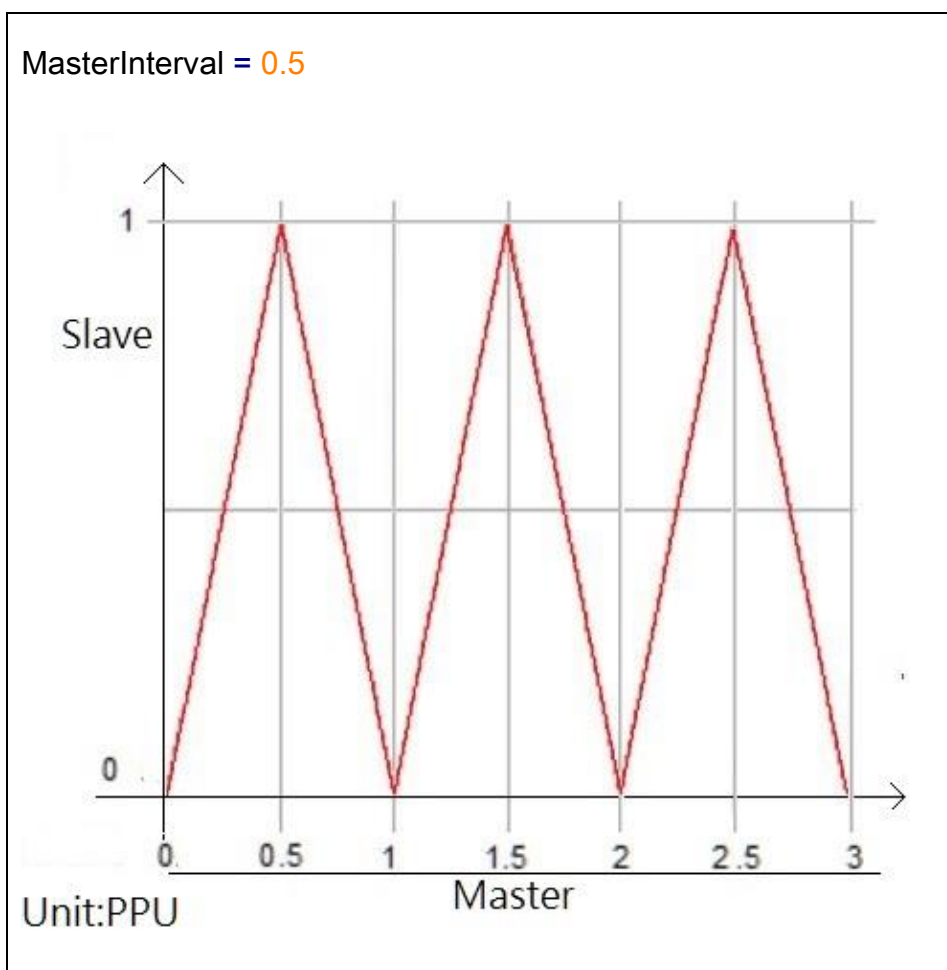
```
{
    ret = ECAT_McAxisCamin(DeviceNo, MasterNo, SlaveNo, TableNo
        , SyncSource, MasterInterval, SlaveScaling)
    if(ret < 0)
    {
        printf("Axis camin is failed:%d\n", ret);
        return;
    }
}
ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    } else {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
}
```

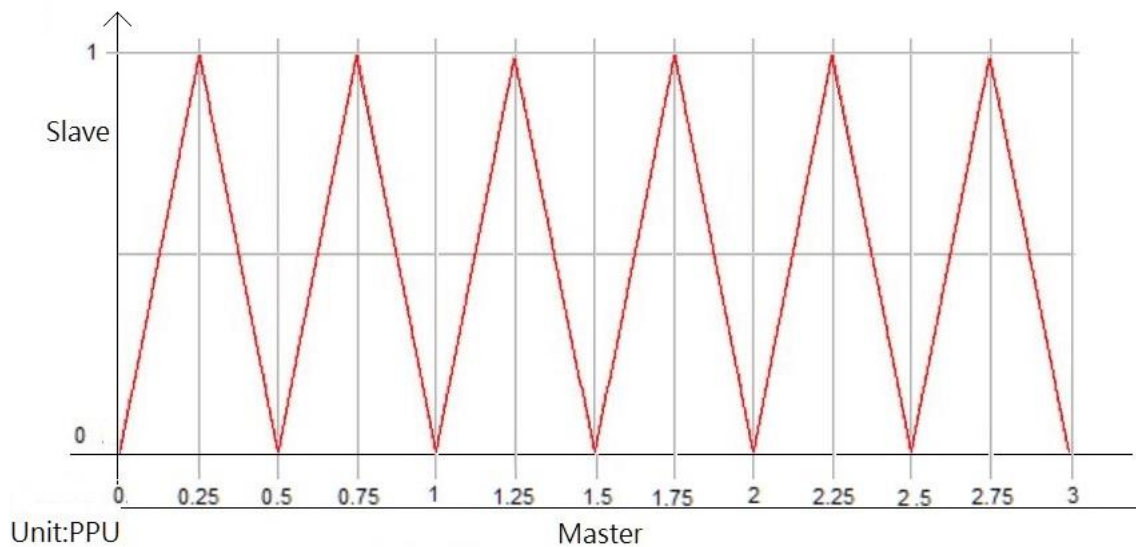
電子凸輪表:



電子凸輪跟隨運動控制使用示意圖:



MasterInterval = 0.25



## 7.2.30. ECAT\_McGetEcamTable

### 說明:

指定一個 E-CAM 表格編號，並將資料讀出。

### 格式:

```
int32_t ECAT_McGetEcamTable(uint16_t DeviceNo, uint16_t TableNo, double *Data,
uint16_t DataSize)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
TableNo	uint16_t	IN	E-CAM 表格編號
Data	double*	OUT	從站位置資料(單位:user unit)
DataSize	uint16_t	IN	讀取的資料筆數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t TableNo = 0;
double Data[1000];
uint16_t DataSize = 1000;
ret = ECAT_McGetEcamTable(DeviceNo, TableNo, Data, DataSize);

if(ret < 0)
{
    printf("Failed to get E-CAM table data:%d\n", ret);
}
else
{
    printf("Get E-CAM table data successfully!\n");
    for(i=0;i<DataSize;i++)
        printf("Data[%u]:%f\n", i, Data[i]);
}
```

## 7.2.31. ECAT\_McConfigEcamTable

### 說明:

配置 E-CAM 表格相關參數。

### 格式:

int32\_t ECAT\_McConfigEcamTable(uint16\_t DeviceNo, uint16\_t TableNo, uint8\_t SlaveAbs)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
TableNo	uint16_t	IN	E-CAM 表格編號
SlaveAbs	uint8_t	IN	從站位置資料類型 0: 相對位置 1: 絕對位置

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t TableNo = 0;
uint8_t SlaveAbs = 0;
ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);

if(ret < 0)
{
    printf("Failed to configure E-CAM table parameter:%d\n", ret);
}
else
{
    printf("Configure E-CAM table parameter successfully!\n");
}
```

---

## 7.2.32. ECAT\_McSetAxisTouchProbeProperty

### 說明:

設定指定軸號 Touch Probe 功能相關配置。注意：使用者必須確定指定軸的伺服馬達符合 CiA402 規範，並且提供 TouchProbe 功能。

### 格式:

int32\_t ECAT\_McSetAxisTouchProbeProperty(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t ProbeNo, uint8\_t Enable, uint8\_t Logic)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
ProbeNo	uint16_t	IN	Touch Probe 編號 1: Touch Probe 1 輸入 2: Touch Probe 2 輸入
Enable	uint8_t	IN	啟用/停用 Touch Probe 功能
Logic	uint8_t	IN	觸發邏輯準位 0: 負緣觸發 (falling edge) 1: 正緣觸發 (rising edge)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProbeNo = 1;
uint8_t Enable = 1;
uint8_t Logic = 1; //rising edge
ret = ECAT_McSetAxisTouchProbeProperty(DeviceNo, AxisNo, ProbeNo, Enable, Logic);
if(ret < 0)
{
    printf("Failed to set Touch Probe property:%d\n", ret);
}
else
{
    printf("Set Touch Probe property successfully!\n");
}
```

---

### 7.2.33. ECAT\_McGetAxisTouchProbeProperty

#### 說明:

取得指定軸號 Touch Probe 功能相關配置。注意：使用者必須確定指定軸的伺服馬達符合 CiA402 規範，並且提供 TouchProbe 功能。

#### 格式:

int32\_t ECAT\_McGetAxisTouchProbeProperty(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t ProbeNo, uint8\_t \*Enable, uint8\_t \*Logic)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
ProbeNo	uint16_t	IN	Touch Probe 編號 1: Touch Probe 1 輸入 2: Touch Probe 2 輸入
Enable	uint8_t *	OUT	啟用/停用 Touch Probe 功能
Logic	uint8_t *	OUT	觸發邏輯準位 0: 負緣觸發 (falling edge) 1: 正緣觸發 (rising edge)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProbeNo = 1;
uint8_t Enable;
uint8_t Logic;
ret = ECAT_McGetAxisTouchProbeProperty(DeviceNo, AxisNo, ProbeNo, &Enable, &Logic);
if(ret < 0)
{
    printf("Failed to get Touch Probe property:%d\n",ret);
}
else
{
    printf("Touch Probe[%u]->Enable:%u\n", ProbeNo, Enable);
    printf("Touch Probe[%u]->Logic:%u\n", ProbeNo, Logic);
}
```

---

## 7.2.34. ECAT\_McGetAxisTouchProbeValue

### 說明:

取得指定軸號 Touch Probe 探測位置。注意：使用者必須確定指定軸的伺服馬達符合 CiA402 規範，並且提供 TouchProbe 功能。

### 格式:

```
int32_t ECAT_McGetAxisTouchProbeValue(uint16_t DeviceNo, uint16_t AxisNo,
uint16_t ProbeNo, double *Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
ProbeNo	uint16_t	IN	Touch Probe 編號 1: Touch Probe 1 輸入 2: Touch Probe 2 輸入
Value	double *	OUT	探測位置

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

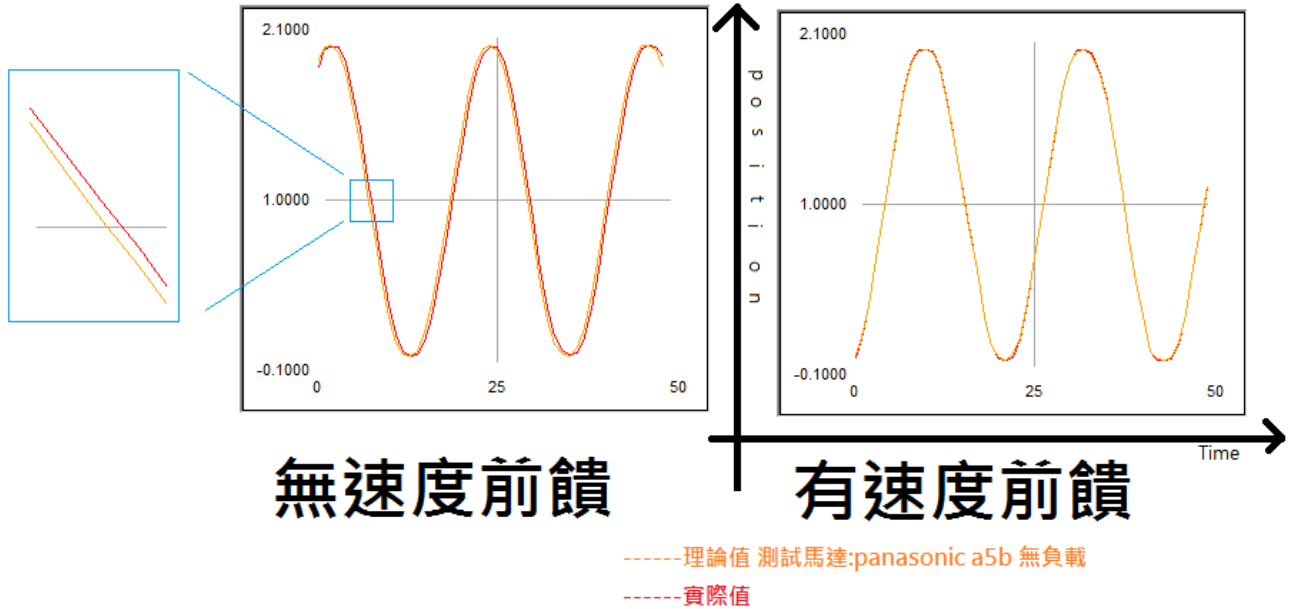
```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProbeNo = 1;
double Value;
ret = ECAT_McGetAxisTouchProbeValue((DeviceNo, AxisNo, ProbeNo, &Value);
if(ret < 0)
{
    printf("Failed to get Touch Probe value:%d\n",ret);
}
else
{
    printf("Touch Probe[%u]->Value:%f\n", ProbeNo, Value);
}
```

---

### 7.2.35. ECAT\_McSetAxisVelocityFeedForwardGain

說明:

設定指定軸號速度前饋增益。注意:只支援部分 CiA402 伺服馬達驅動器。



格式:

int32\_t ECAT\_McSetAxisVelocityFeedForwardGain(uint16\_t DeviceNo, uint16\_t AxisNo, double Gain)

參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Gain	double	IN	速度前饋增益 設置範圍:0(default)~1

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Gain = 0.95;
ret = ECAT_McSetAxisVelocityFeedForwardGain(DeviceNo, AxisNo, Gain);
if(ret < 0)
{
    printf("Failed to set axis Velocity Feed Forward Gain%d\n", ret);
}
else
{
    printf("Set axis Velocity Feed Forward Gain successfully!\n");
}
```

---

## 7.2.36. ECAT\_McGetAxisVelocityFeedForwardGain

### 說明:

取得指定軸號速度前饋增益。注意:只支援 CiA402 伺服馬達驅動器。

### 格式:

int32\_t ECAT\_McGetAxisVelocityFeedForwardGain(uint16\_t DeviceNo, uint16\_t AxisNo, double \* Gain)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Gain	Double*	OUT	速度前饋增益

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Gain;
ret = ECAT_McGetAxiStVelocityFeedForwardGain(DeviceNo, AxisNo, &Gain);
if(ret < 0)
{
    printf("Failed to get axis Velocity Feed Forward Gain:%d\n", ret);
}
else
{
    printf("Axis[%u] Velocity Feed Forward Gain:%f\n", AxisNo, Gain);
}
```

---

### 7.2.37. ECAT\_McSetAxisPosSoftwareLimitStatus

**說明:**

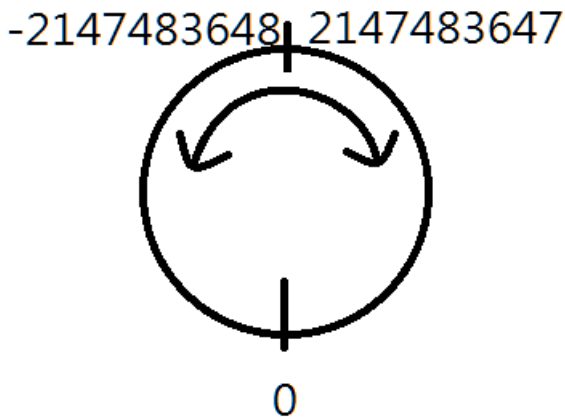
設定指定軸號之位置軟體極限之狀態。注:僅支援 CiA402 驅動器及虛擬軸。

注: (1)當停用此功能時, 因內部位置計數範圍為-2147483648 ~ 2147483647, 超出此範圍時, “軸號最後出錯代碼”為-1134

(2)當啟用無限旋轉功能時

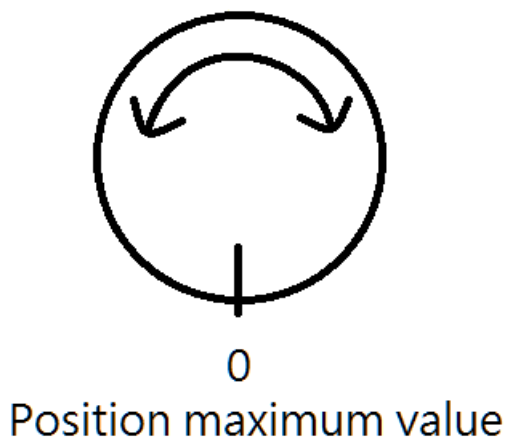
(2-1) 當位置最大、最小軟體極限為 0, 或位置最小軟體極限不為 0

若馬達向正方向運行, 且超過 2147483647, 則位置將從-2147483648 開始向正方向計數;  
若馬達向復方向運行, 且超過-2147483648, 則位置將從 2147483647 開始向負方向計數



(2-2) 當位置最小軟體極限為 0, 位置最大軟體極限為正值

若馬達向正方向運行, 且超過位置最大軟體極限, 則位置將從 0 開始向正方向計數;  
若馬達向負方向運行, 且超過 0, 則位置將從位置最大軟體極限開始向負方向計數



範例: Status: 0	單軸	<table border="1"> <thead> <tr> <th>Axis NO.</th> <th>CmdPosition</th> <th>Position</th> <th>Velocity</th> <th>Axis State</th> <th>Axis Error</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>10.000</td> <td>10.000</td> <td>0.0</td> <td>StandStill</td> <td>0</td> </tr> </tbody> </table>	Axis NO.	CmdPosition	Position	Velocity	Axis State	Axis Error	0	10.000	10.000	0.0	StandStill	0
	Axis NO.	CmdPosition	Position	Velocity	Axis State	Axis Error								
0	10.000	10.000	0.0	StandStill	0									
群組														
Status:1 ErrorStop: 0  X 軸軟體極限 最大值:8 最小值:-8	單軸	<table border="1"> <thead> <tr> <th>Axis NO.</th> <th>CmdPosition</th> <th>Position</th> <th>Velocity</th> <th>Axis State</th> <th>Axis Error</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>10.000</td> <td>8.000</td> <td>0.0</td> <td>StandStill</td> <td>-1134</td> </tr> </tbody> </table>	Axis NO.	CmdPosition	Position	Velocity	Axis State	Axis Error	0	10.000	8.000	0.0	StandStill	-1134
	Axis NO.	CmdPosition	Position	Velocity	Axis State	Axis Error								
0	10.000	8.000	0.0	StandStill	-1134									
群組														
Status:1 ErrorStop: 1  X 軸軟體極限 最大值:8 最小值:-8	單軸	<table border="1"> <thead> <tr> <th>Axis NO.</th> <th>CmdPosition</th> <th>Position</th> <th>Velocity</th> <th>Axis State</th> <th>Axis Error</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>8.000</td> <td>7.999</td> <td>0.0</td> <td>ErrorStop</td> <td>-1134</td> </tr> </tbody> </table>	Axis NO.	CmdPosition	Position	Velocity	Axis State	Axis Error	0	8.000	7.999	0.0	ErrorStop	-1134
	Axis NO.	CmdPosition	Position	Velocity	Axis State	Axis Error								
0	8.000	7.999	0.0	ErrorStop	-1134									
群組														

格式:

int32\_t ECAT\_McSetAxisPosSoftwareLimitStatus(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t Status, uint16\_t ErrorStop)

參數:

名稱	類型	IN or OUT	說明
----	----	-----------	----

DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Status	uint16_t	IN	<p>0:停用(default) 內部位置計數範圍為 -2147483648 ~ 2147483647 超出此範圍時，“軸號最後出錯代碼”為 -1134</p> <p>1:啟用軟體極限功能 2:啟用無限旋轉功能</p>
ErrorStop	uint16_t	IN	<p>0: 碰到軟體極限時， “軸號最後出錯代碼”為-1134， 命令位置正常運動， 實際位置會限制住。</p> <p>1: 碰到軟體極限時， 軸狀態變為 MC_AS_ERRORSTOP， 並停止運動，若為群組運動，則停止運動 且清除剩餘群組命令。</p>

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Status = 0;
uint16_t ErrorStop = 0;
ret = ECAT_McSetAxisPosSoftwareLimitStatus(DeviceNo,AxisNo, Status, ErrorStop);
if(ret < 0)
{
    printf("Failed to set axis position software limit status :%d\n",ret);
}
else
{
    printf("Set axis position software limit status successfully!\n");
}
```

---

## 7.2.38. ECAT\_McGetAxisPosSoftwareLimitStatus

### 說明:

取得指定軸號之位置軟體極限之狀態。注:僅支援 CiA402 及虛擬軸。

### 格式:

```
int32_t ECAT_McGetAxisPosSoftwareLimitStatus(uint16_t DeviceNo, uint16_t AxisNo,
uint16_t *Status, uint16_t *ErrorStop)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Status	uint16_t	OUT	0:停用 1:啟用
ErrorStop	uint16_t	OUT	0: 碰到軟體極限時， 軸號最後出錯代碼為-1134， 命令位置正常運動， 實際位置會限制住。 1: 碰到軟體極限時， 軸狀態變為 MC_AS_ERRORSTOP， 並停止運動，若為群組運動，則停止運動 且清除剩餘群組命令。

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Status = 0;
uint16_t ErrorStop = 0;
ret = ECAT_McGetAxisPosSoftwareLimitStatus(DeviceNo, AxisNo, &Status, &ErrorStop);
if(ret < 0)
{
    printf("Failed to get axis position software limit status:%d\n",ret);
}
else
{
    printf("Axis[%u] position software limit [Status:%f] , [ErrorStop:%f] \n", AxisNo, Status, ErrorStop);
}
```

---

## 7.2.39. ECAT\_McSetAxisPosSoftwareLimit

### 說明:

設定指定軸號之位置軟體極限。注:僅支援 CiA402 及虛擬軸。

### 格式:

```
int32_t ECAT_McSetAxisPosSoftwareLimit(uint16_t DeviceNo, uint16_t AxisNo, double
Maximum, double Minimum);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Maximum	double	IN	位置最大軟體極限(單位: user unit)
Minimum	double	IN	位置最小軟體極限(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Maximum = 100.0;
double Minimum = -100.0;
ret = ECAT_McSetAxisPosSoftwareLimit(DeviceNo,AxisNo, Maximum, Minimum);
if(ret < 0)
{
    printf("Failed to set axis position software limit :%d\n",ret);
}
else
{
    printf("Set axis position software limit successfully!\n");
}
```

---

## 7.2.40. ECAT\_McGetAxisPosSoftwareLimit

### 說明:

取得指定軸號之位置軟體極限。注:僅支援 CiA402 及虛擬軸。

### 格式:

```
int32_t ECAT_McGetAxisPosSoftwareLimit(uint16_t DeviceNo, uint16_t AxisNo, double *Maximum, double *Minimum)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Maximum	Double*	OUT	位置最大軟體極限(單位: user unit)
Minimum	Double*	OUT	位置最小軟體極限(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Maximum;
double Minimum;
ret = ECAT_McGetAxisPosSoftwareLimit(DeviceNo,AxisNo,&Maximum,&Minimum);
if(ret < 0)
{
    printf("Failed to get axis position software limit:%d\n",ret);
}
else
{
    printf("Axis[%u] position software limit [Maximin:%f] , [Minimum:%f] \n", AxisNo, Maximum, Minimum);
}
```

---

## 7.2.41. ECAT\_OpenMotionConfig

### 說明:

解析 Utility 產生出來的參數檔，請參考"運動控制初始化參數檔編輯步驟"。

### 格式:

```
int32_t ECAT_OpenMotionConfig(char* bstrFileName, uint16_t *AxisCnt
, uint16_t SlaveNo[], uint16_t SubAxisNo[], double PPU[], int32_t HomeMethod[]
, double HomeSpeedSeachSw[], double HomeSpeedSeachZr[], double HomeAcc[]
, uint32_t EncoderPPR[], uint32_t MotorPPR[])
```

### 參數:

名稱	類型	IN or OUT	說明
bstrFileName	char*	IN	軸設定檔案名稱
AxisCnt	uint16_t	OUT	軸數
SlaveNo	uint16_t *	OUT	從站編號陣列，陣列的索引編號即為對應的軸號
SubAxisNo	uint16_t *	OUT	從站驅動器子軸號
PPU	Double*	OUT	Pulse per Unit 參數
HomeMethod	int32_t *	OUT	原點復歸模式(範圍: 1~37) 請查閱伺服馬達文件資料確認模式是否支援
HomeSpeedSeachSw	Double*	OUT	尋找原點(ORG)訊號時使用的速度 (單位: user unit/s)
HomeSpeedSeachZr	Double*	OUT	尋找馬達索引(EZ)訊號時使用的速度 (單位: user unit/s)
HomeAcc	Double*	OUT	執行自動原點復歸時使用的加速度 (單位: user unit/s^2)
EncoderPPR	uint32_t *	OUT	編碼器單圈脈波 (單位: Pulses/Revolution)
MotorPPR	uint32_t *	OUT	馬達單圈脈波數 (單位: Pulses/Revolution)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
char* Filename = "MotionConfig.motcfg";
uint16_t AxisCnt;
uint16_t SlaveNo[MC_AXIS_NO_MAX];
uint16_t SubAxisNo[MC_AXIS_NO_MAX];
double PPU[MC_AXIS_NO_MAX];
int32_t HomeMethod[MC_AXIS_NO_MAX];
double HomeSpeedSeachSw [MC_AXIS_NO_MAX];
double HomeSpeedSeachZr[MC_AXIS_NO_MAX];
double HomeAcc[MC_AXIS_NO_MAX];
uint32_t EncoderPPR [MC_AXIS_NO_MAX];
uint32_t MotorPPR [MC_AXIS_NO_MAX];

CoInitialize(NULL);
ret = ECAT_OpenMotionConfig(FileName, &AxisCnt
, SlaveNo, SubAxisNo, PPU, HomeMethod
, HomeSpeedSeachSw, HomeSpeedSeachZr, HomeAcc
, EncoderPPR, MotorPPR);
CoUninitialize();
if(ret < 0)
{
    printf("Failed to Open Motion Config file:%d\n",ret);
}
```



## 7.2.42. ECAT\_McSetAxisMaxVelocity

### 說明:

設定指定軸號最大速度，提供以下函式使用。

*ECAT\_McAxisMoveAbs\_P2P*

*ECAT\_McAxisMoveRel\_P2P*

*ECAT\_McGroupMoveLineAbs\_P2P*

*ECAT\_McGroupMoveLineRel\_P2P*

### 格式:

int32\_t ECAT\_McSetAxisMaxVelocity(uint16\_t DeviceNo, uint16\_t AxisNo, double MaxVelocity)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
MaxVelocity	double	IN	最大速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double MaxVelocity = 100;
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}
else
{
    printf("Set axis MaxVelocity successfully!\n");
}
```

---

## 7.2.43. ECAT\_McGetAxisMaxVelocity

### 說明:

取得指定軸號指定軸號最大速度。

### 格式:

```
int32_t ECAT_McGetAxisMaxVelocity(uint16_t DeviceNo, uint16_t AxisNo, double  
*MaxVelocity)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
MaxVelocity	Double*	OUT	最大速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double MaxVelocity;
ret = ECAT_McGetAxisMaxVelocity(DeviceNo, AxisNo, &MaxVelocity);
if(ret < 0)
{
    printf("Failed to get axis MaxVelocity:%d\n", ret);
}
else
{
    printf("Axis[%u] MaxVelocity:%fn", AxisNo, MaxVelocity);
}
```

---

## 7.2.44. ECAT\_McSetAxisDIActiveLevel

### 說明:

設定指定軸號極限與原點的檢出準位。

對於一般支援 CiA402 的驅動器來說，通常為”高準位檢出”，也就是說，當 object 0x60FD DI bit 訊號為 1 時，為檢出狀態，此時馬達無法運轉；而當驅動器為”低準位檢出”時，也就是說，當 object 0x60FD DI bit 訊號為 0 時，為檢出狀態，此時馬達無法運轉，就需要使用此函式告知 EtherCAT MainDevice 該驅動器為”低準位檢出”。

注: (1)不論設定為”低準位檢出”或”高準位檢出”，使用 ECAT\_McGetAxisDI 讀取訊號時 1 代表該訊號被檢出。

(2)需要確認控制卡版本為 1.0.18 或以上才可使用

### 格式:

```
int32_t ECAT_McSetAxisDIActiveLevel(uint16_t DeviceNo, uint16_t AxisNo, uint8_t POT, uint8_t NOT, uint8_t ORG)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
POT	uint8_t	IN	正極限訊號 0: 低準位檢出 1: 高準位檢出(預設)
NOT	uint8_t	IN	負極限訊號 0: 低準位檢出 1: 高準位檢出(預設)
ORG	uint8_t	IN	原點訊號 0: 低準位檢出 1: 高準位檢出(預設)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint8_t POT = 0;
uint8_t NOT = 0;
uint8_t ORG = 0;
ret = ECAT_McSetAxisDIActiveLevel(DeviceNo, AxisNo, POT, NOT, ORG);
if(ret < 0)
{
    printf("Failed to set axis di active level:%d\n",ret);
}
else
{
    printf("Set axis di active level successfully!\n");
}
```

---

## 7.2.45. ECAT\_McGetAxisDIActiveLevel

### 說明:

讀取指定軸號極限與原點的檢出準位。

### 格式:

```
int32_t ECAT_McGetAxisDIActiveLevel(uint16_t DeviceNo, uint16_t AxisNo, uint8_t
*POT, uint8_t *NOT, uint8_t *ORG)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
POT	uint8_t*	OUT	正極限訊號 0: 低準位檢出 1: 高準位檢出(預設)
NOT	uint8_t*	OUT	負極限訊號 0: 低準位檢出 1: 高準位檢出(預設)
ORG	uint8_t*	OUT	原點訊號 0: 低準位檢出 1: 高準位檢出(預設)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint8_t POT;
uint8_t NOT;
uint8_t ORG;
ret = ECAT_McGetAxisDIActiveLevel(DeviceNo, AxisNo, &POT, &NOT, &ORG);
if(ret < 0)
{
    printf("Failed to get axis di active level:%d\n",ret);
}
else
{
    printf("POT: %u, NOT: %u, ORG: %u\n ", POT, NOT, ORG);
}
```

---

## 7.2.46. ECAT\_McSetAxisActualPosition

### 說明:

設定指定軸號當前位置。

使用 Home Method 37 進行設定

注: (1)此函式執行會將狀態切換至 MC\_AS\_HOMING，此函式返回後需檢查狀態為 MC\_AS\_STANDSTILL 才算完成。

(2) ECAT\_McAxisHome 已包含此指令，不需要再另外使用此函式

### 格式:

int32\_t ECAT\_McSetAxisActualPosition(uint16\_t DeviceNo, uint16\_t AxisNo, double Position)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Position	double	IN	指定軸號當前位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos = 0;
uint32_t State;

ret = ECAT_McSetAxisActualPos(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to set axis actual position:%d\n", ret);
}
else
{
    while(1)
    {
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if(ret == 0 && State == MC_AS_STANDSTILL)
            break;
    }
}
```

---

## 7.2.47. ECAT\_McSetAxisActualPositionBy35

### 說明:

設定指定軸號當前位置。

使用 Home Method 35 進行設定

注: (1)此函式執行會將狀態切換至 MC\_AS\_HOMING，此函式返回後需檢查狀態為 MC\_AS\_STANDSTILL 才算完成。

(2) ECAT\_McAxisHome 已包含此指令，不需要再另外使用此函式

### 格式:

```
int32_t ECAT_McSetAxisActualPositionBy35(uint16_t DeviceNo, uint16_t AxisNo,
double Position)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Position	double	IN	指定軸號當前位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos = 0;
uint32_t State;

ret = ECAT_McSetAxisActualPosBy35(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to set axis actual position:%d\n", ret);
}
else
{
    while(1)
    {
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if(ret == 0 && State == MC_AS_STANDSTILL)
            break;
    }
}
```

---

## 7.2.48. ECAT\_McSetAxisCommandPosition

**說明:**

不再支援

## 7.2.49. ECAT\_McSetAxisInpSignal

### 說明:

設定指定軸號的定位訊號來源。

當啟用此功能時，MC\_AS\_STANDSTILL 與 MC\_GS\_STANDBY 就代表定位完成。

當使用群組命令且命令緩衝區有命令時(BUFFERED 模式)，會待到定位完成後才開始下一個命令。

注: (1)可以與 ECAT\_McSetAxisInpCompare 同時使用。

(2)運動命令速度過大且時間過短時，可能造成運動命令已經下完但是馬達還沒開始運轉的情形，在此情形可能造成定位訊號的誤判，建議搭配 ECAT\_McSetAxisInpCompare 一起使用。

例:表 7.4 為預設之 PDO Mapping，以 Panasonic 驅動器為例，INP 訊號為 Digital inputs 的第 24bit，所以 Offset 設 13，Bit 設 24。

表 7.4

	TxPdo Entries	Size(byte)	Offset(byte)
Mode0	Statusword	2	0
	Error code	2	2
	Modes of operation display	1	4
	Position actual value	4	5
	Velocity actual value	4	9
	Digital inputs	4	13

### 格式:

int32\_t ECAT\_McSetAxisInpSignal(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t Offset, uint16\_t Bit, int32\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號

AxisNo	uint16_t	IN	軸號
Offset	uint16_t	IN	TxPDO 偏移值
Bit	uint16_t	IN	指定位元
Enable	uint32_t	IN	預設值: 禁用 0: 禁用 1: 啟用

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Offset= 13;
uint16_t Bit= 24;
uint32_t Enable = 1;
ret = ECAT_McSetAxisInpSignal(DeviceNo, AxisNo, Offset, Bit, Enable);
if(ret < 0)
{
    printf("Failed to set axis inp signal:%d\n", ret);
}
else
{
    printf("Set axis inp signal successfully!\n");
}
```

---

## 7.2.50. ECAT\_McGetAxisInpSignal

### 說明:

取得指定軸號的定位訊號來源

### 格式:

```
int32_t ECAT_McGetAxisInpSignal(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Offset,
uint16_t *Bit, int32_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Offset	uint16_t*	OUT	TxPDO 偏移值
Bit	uint16_t*	OUT	指定位元
Enable	uint32_t*	OUT	0: 禁用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Offset;
uint16_t Bit;
uint32_t Enable;
ret = ECAT_McGetAxisInpSignal(DeviceNo, AxisNo, &Offset, &Bit, &Enable);
if(ret < 0)
{
    printf("Failed to get axis inp signal:%d\n", ret);
}
else
{
    printf("Axis[%u] Offset:%u, Bit: %u, Enable: %u\n", AxisNo, Offset, Bit, Enable);
}
```

---

## 7.2.51. ECAT\_McSetAxisInpCompare

### 說明:

設定指定軸號的定位條件。

當啟用此功能時，MC\_AS\_STANDSTILL 與 MC\_GS\_STANDBY 就代表定位完成。

當使用群組命令且命令緩衝區有命令時(BUFFERED 模式)，會待到定位完成後才開始下一個命令。

當一個命令執行完，命令位置與實際位置的差值在定位區間之內，且經過穩定時間，才會將狀態轉變為 MC\_AS\_STANDSTILL/ MC\_GS\_STANDBY。

注：可以與 ECAT\_McSetAxisInpSignal 同時使用。

### 格式:

int32\_t ECAT\_McSetAxisInpCompare(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t Time\_ms, double PosWindow, int32\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_ms	uint32_t	IN	穩定時間 單位: ms
PosWindow	double	IN	定位區間 單位:user unit
Enable	uint32_t	IN	預設值: 禁用 0: 禁用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms = 3;
double PosWindow = 0.1;
uint32_t Enable = 1;
ret = ECAT_McSetAxisInpCompare(DeviceNo, AxisNo, Time_ms, PosWindow, Enable);
if(ret < 0)
{
    printf("Failed to set axis inp compare:%d\n", ret);
}
else
{
    printf("Set axis inp compare successfully!\n");
}
```

---

## 7.2.52. ECAT\_McGetAxisInpCompare

### 說明:

取得指定軸號的定位條件

### 格式:

```
int32_t ECAT_McGetAxisInpCompare(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *Time_ms, double *PosWindow, int32_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_ms	uint32_t*	OUT	穩定時間 單位: ms
PosWindow	double*	OUT	定位區間 單位:user unit
Enable	uint32_t*	OUT	0: 禁用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms;
double PosWindow;
uint32_t Enable;
ret = ECAT_McGetAxisInpCompare(DeviceNo, AxisNo, &Time_ms, &PosWindow, &Enable);
if(ret < 0)
{
    printf("Failed to get axis inp compare:%d\n", ret);
}
else
{
    printf("Axis[%u] Time_ms:%u, PosWindow: %f, Enable: %u\n", AxisNo, Time_ms, PosWindow, Enable);
}
```

---



## 7.2.53. ECAT\_McSetAxisInpTimeOut

### 說明:

設定指定軸號的定位超時。

當啟用定位功能時，若命令已經送完，但是一段時間之後沒有檢測到定位訊號，則軸狀態變為 MC\_AS\_ERRORSTOP。

### 格式:

int32\_t ECAT\_McSetAxisInpTimeOut(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t Time\_ms)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_ms	uint32_t	IN	超時 單位: ms 預設值: 2000ms

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms = 30;
ret = ECAT_McSetAxisInpTimeOut(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis inp timeout:%d\n", ret);
}
else
{
    printf("Set axis inp timeout successfully!\n");
}
```

---

## 7.2.54. ECAT\_McGetAxisInpTimeOut

### 說明:

取得指定軸號的定位超時

### 格式:

int32\_t ECAT\_McGetAxisInpTimeOut(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t \*Time\_ms)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_ms	uint32_t*	OUT	超時 單位: ms

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms;
ret = ECAT_McGetAxisInpTimeOut(DeviceNo, AxisNo, &Time_ms);
if(ret < 0)
{
    printf("Failed to get axis inp timeout:%d\n", ret);
}
else
{
    printf("Axis[%u] Time_ms:%u\n", AxisNo, Time_ms);
}
```

---

## 7.2.55. ECAT\_McSetAxisWanErrEnable

### 說明:

設定指定軸號當遇到 WAN (警告)訊號時是否將狀態改變為 MC\_AS\_ERRORSTOP。

注: WAN (警告)訊號可使用 ECAT\_McGetAxisDI 讀取

### 格式:

int32\_t ECAT\_McSetAxisWanErrEnable(uint16\_t DeviceNo, uint16\_t AxisNo, uint32\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Enable	uint32_t	IN	預設值: 啟用 0: 禁用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Enable = 0;
ret = ECAT_McSetAxisWanErrEnable(DeviceNo, AxisNo, Enable);
if(ret < 0)
{
    printf("Failed to set axis WanErr enable :%d\n", ret);
}
else
{
    printf("Set axis WanErr enable successfully!\n");
}
```

---

## 7.2.56. ECAT\_McGetAxisWanErrEnable

### 說明:

取得指定軸號的 WAN (警告)訊號功能設定。

### 格式:

```
int32_t ECAT_McGetAxisWanErrEnable(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Enable	uint32_t*	OUT	0: 禁用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Enable;
ret = ECAT_McGetAxisWanErrEnable(DeviceNo, AxisNo, &Enable);
if(ret < 0)
{
    printf("Failed to get WanErr enable:%d\n", ret);
}
else
{
    printf("Axis[%u] WanErr enable:%u\n", AxisNo, Enable);
}
```

---



## 7.2.57. ECAT\_McEnable\_Directly\_Ex

### 說明:

取代舊的函式" ECAT\_McEnable\_Directly"

啟用\_Directly 相關函式，請在 ECAT\_StartDeviceOpTask 前使用此函式

使用 ECAT\_StopDeviceOpTask 與 ECAT\_OpenDevice 時會自動禁用此功能

注: 在 EnumCycleTime 為 1ms 時，軸數必須少於 64 軸

在 EnumCycleTime 為 0.5ms 時，軸數必須少於 32 軸

### 格式:

```
int32_t ECAT_McEnable_Directly_Ex(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;

ret = ECAT_McEnable_Directly_Ex(DeviceNo, AxisNo, Enable);
if(ret < 0)
{
    printf("Failed to enable :%d\n", ret);
}
else
{
    printf("Enable successfully!\n");
}
```

---

## 7.3. 單軸運動狀態

### 7.3.1. ECAT\_McGetAxisActualPos

#### 說明:

取得指定軸號當前位置。

注意:將 **AxisNo** 設為 65535 時，可以一次取得所有軸的當前位置。

#### 格式:

```
int32_t ECAT_McGetAxisActualPos(uint16_t DeviceNo, uint16_t AxisNo, double *Pos)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double*	OUT	指定軸號當前位置(單位:user unit)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos;
ret = ECAT_McGetAxisActualPos(DeviceNo, AxisNo, &AxisPos);
if(ret < 0)
{
    printf("Failed to get axis actual position:%d\n", ret);
}
else
{
    printf("Axis Actual Position:%f\n", AxisPos);
}

```

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double AxisPos[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisActualPos(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to get axis actual position:%d\n", ret);
}
else
{
    int i;
    for(i=0;i< MC_AXIS_NO_MAX;i++)
    {
        printf("Axis[%d] Actual Position:%f\n", i, AxisPos[ i ] );
    }
}

```



### 7.3.2. ECAT\_McGetAxisActualPos\_Ex

**說明:**

降低使用 ECAT\_McGetAxisActualPos 所花費的時間

### 7.3.3. ECAT\_McGetAxisActualPos\_Directly

**說明:**

降低使用 ECAT\_McGetAxisActualPos\_Ex 所花費的時間

注:需先啟用 ECAT\_McEnable\_Directly

### 7.3.4. ECAT\_McGetAxisCommandPos

**說明:**

取得指定軸號命令位置。

注意:將 **AxisNo** 設為 65535 時 , 可以一次取得所有軸的命令位置。

**格式:**

```
int32_t ECAT_McGetAxisCommandPos(uint16_t DeviceNo, uint16_t AxisNo, double *Pos)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double*	OUT	指定軸號命令位置(單位:user unit)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos;
ret = ECAT_McGetAxisCommandPos(DeviceNo, AxisNo, &AxisPos);
if(ret < 0)
{
    printf("Failed to get axis command position:%d\n", ret);
}
else
{
    printf("Axis command Position:%f\n", AxisPos);
}
```

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double AxisPos[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisCommandPos(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to get axis command position:%d\n", ret);
}
else
{
    int i;
    for(i=0;i< MC_AXIS_NO_MAX;i++)
    {
        printf("Axis[%d] CommandPosition:%f\n", i, AxisPos[ i ] );
    }
}
```



### 7.3.5. ECAT\_McGetAxisCommandPos\_Ex

**說明:**

降低使用 ECAT\_McGetAxisCommandPos 所花費的時間

### 7.3.6. ECAT\_McGetAxisCommandPos\_Directly

**說明:**

降低使用 ECAT\_McGetAxisCommandPos\_Ex 所花費的時間

注:需先啟用 ECAT\_McEnable\_Directly

### 7.3.7. ECAT\_McGetAxisActualVel

#### 說明:

取得指定軸號當前速度。

注意:將 **AxisNo** 設為 65535 時，可以一次取得所有軸的當前速度。

#### 格式:

int32\_t ECAT\_McGetAxisActualVel(uint16\_t DeviceNo, uint16\_t AxisNo, double \*Vel)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Vel	double*	OUT	取得指定軸號當前速度 (單位: user unit/s)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisVel;
ret = ECAT_McGetAxisActualVel(DeviceNo, AxisNo, &AxisVel);
if(ret < 0)
{
    printf("Failed to get axis actual velocity:%d\n", ret);
}
else
{
    printf("Axis Actual Velocity:%f\n", AxisVel);
}

```

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double AxisVel[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisActualVel(DeviceNo, AxisNo, AxisVel);
if(ret < 0)
{
    printf("Failed to get axis actual velocity:%d\n", ret);
}
else
{
    int i;
    for(i=0;i< MC_AXIS_NO_MAX;i++)
    {
        printf("Axis[%d] Actual velocity:%f\n", i, AxisVel[ i ]);
    }
}

```



### 7.3.8. ECAT\_McGetAxisActualVel\_Ex

**說明:**

降低使用 ECAT\_McGetAxisActualVel 所花費的時間

注:需先啟用 ECAT\_McEnable\_Directly



### 7.3.9. ECAT\_McGetAxisActualPosVel\_Ex

#### 說明:

取代舊的函式"ECAT\_McGetAxisActualPosVel"

取得指定軸號當前位置及速度。

注意:將 AxisNo 設為 65535 時，可以一次取得所有軸的當前位置及速度。

#### 格式:

int32\_t ECAT\_McGetAxisActualPosVel\_Ex(uint16\_t DeviceNo, uint16\_t AxisNo, float \*Pos, float \*Vel)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	float*	OUT	指定軸號當前位置(單位:user unit)
Vel	float*	OUT	指定軸號當前速度(單位:user unit/s)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
float AxisPos;
float AxisVel;

ret = ECAT_McGetAxisActualPosVel_Ex(DeviceNo, AxisNo, &AxisPos, &AxisVel);
if(ret < 0)
{
    printf("Failed to get axis actual position and velocity:%d\n", ret);
}
else
{
    printf("Axis Actual Position:%f , Velocity:%f \n", AxisPos, AxisVel);
}

```

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 65535;
float AxisPos[MC_AXIS_NO_MAX];
float AxisVel[MC_AXIS_NO_MAX];

ret = ECAT_McGetAxisActualPosVel_Ex(DeviceNo, AxisNo, AxisPos, AxisVel);
if(ret < 0)
{
    printf("Failed to get axis actual position and velocity:%d\n", ret);
}
else
{
    int i;
    for(i=0;i< MC_AXIS_NO_MAX;i++)
    {
        printf("Axis[%d] Actual Position:%f , Velocity:%f \n", i, AxisPos[ i ], AxisVel [ i ] );
    }
}

```

---

}

}

---

### 7.3.10. ECAT\_McGetAxisActualTorque

**說明:**

取得指定軸號當前轉矩。

注意:將 **AxisNo** 設為 65535 時，可以一次取得所有軸的當前轉矩。

**格式:**

int32\_t ECAT\_McGetAxisActualTorque(uint16\_t DeviceNo, uint16\_t AxisNo, double \*Torque)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Torque	double*	OUT	轉矩 單位請參考驅動器使用手冊的物件 (Torque Actual Value: 6077h)，一般為額定轉矩的 0.1% 額定轉矩請參考 (Motor rated torque: 6076h)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Torque;
ret = ECAT_McGetAxisActualTorque(DeviceNo, AxisNo, &Torque);
if(ret < 0)
{
    printf("Failed to get axis actual torque:%d\n", ret);
}
else
{
    printf("Axis Actual torque:%f\n", Torque);
}

```

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double Torque[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisActualTorque(DeviceNo, AxisNo, Torque);
if(ret < 0)
{
    printf("Failed to get axis actual torque:%d\n", ret);
}
else
{
    int i;
    for(i=0;i< MC_AXIS_NO_MAX;i++)
    {
        printf("Axis[%d] Actual torque:%f\n", i, Torque[ i ] );
    }
}

```

### 7.3.11. ECAT\_McGetAxisState

**說明:**

取得指定軸號當前狀態。狀態定義可參考表 7.5。

**格式:**

```
int32_t ECAT_McGetAxisState(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *State)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
State	uint32_t*	OUT	取得指定軸號當前狀態編號(如表 7.5 所示)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.5: 軸狀態編號

標頭檔定義名稱	數值	說明
MC_AS_DISABLED	0	軸尚未啟用
MC_AS_STANDSTILL	1	軸啟用且為停止狀態，準備接收新的運動命令
MC_AS_ERRORSTOP	2	軸目前出現錯誤且為停止狀態
MC_AS_STOPPING	3	軸目前在停止運動中
MC_AS_HOMING	4	軸目前在原點復歸中
MC_AS_DISCRETEMOTION	5	軸目前單軸運動中
MC_AS_CONTINUOUSMOTION	6	軸目前連續運動中
MC_AS_SYNCHRONIZEDMOTION	7	軸目前同步運動中 軸在群組中，群組正在進行補間運動

## 使用範例:

[C/C++]

```

int32_t ret;
char buf[512];
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n", ret);
}
else
{
    switch(State)
    {
        case MC_AS_DISABLED:
            printf(buf,"Disabled");
            break;
        case MC_AS_STANDSTILL:
            sprintf(buf,"StandStill");
            break;
    }
}

```

---

```
case MC_AS_ERRORSTOP:
    sprintf(buf,"ErrorStop");
    break;
case MC_AS_STOPPING:
    sprintf(buf,"Stopping");
    break;
case MC_AS_HOMING:
    sprintf(buf,"Homing");
    break;
case MC_AS_DISCRETEMOTION:
    sprintf(buf,"DiscMotion");
    break;
case MC_AS_CONTINUOUSMOTION:
    sprintf(buf,"ContMotion");
    break;
case MC_AS_SYNCHRONIZEDMOTION:
    sprintf(buf,"SyncMotion");
    break;
default:
    sprintf(buf,"Invalid");
}
printf("Axis State:%s\n", buf);
}
```

---



## 7.3.12. ECAT\_McGetAxisState\_Ex

### 說明:

降低使用 ECAT\_McGetAxisState 所花費的時間

注:需先啟用 ECAT\_McEnable\_Directly

### 7.3.13. ECAT\_McGetAxisState\_Directly

**說明:**

降低使用 ECAT\_McGetAxisState\_Ex 所花費的時間

注:需先啟用 ECAT\_McEnable\_Directly

### 7.3.14. ECAT\_McGetAxisLastError

**說明:**

取得指定軸號最後出錯代碼。

**格式:**

```
int32_t ECAT_McGetAxisLastError(uint16_t DeviceNo, uint16_t AxisNo, int32_t *Error)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Error	int32_t *	OUT	指定軸號最後出錯代碼(請參考附錄"函式錯誤回傳代碼"說明)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Error;
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n",ret);
}
else
{
    if(State == MC_AS_ERRORSTOP)
    {
        ret = ECAT_McGetAxisLastError(DeviceNo, AxisNo, &Error);
        if(ret < 0)
        {
            printf("Failed to get axis last error:%d\n", ret);
        }
        else
        {
            printf("Axis Last Error:%d\n", Error);
        }
    }
}
}
```

### 7.3.15. ECAT\_McGetAxisLastError\_Ex

**說明:**

降低使用 ECAT\_McGetAxisLastError 所花費的時間

### 7.3.16. ECAT\_McGetAxisDriveError

**說明:**

取得指定軸號驅動器出錯代碼。

**格式:**

```
int32_t ECAT_McGetAxisDriveError(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Error)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Error	int16_t *	OUT	指定軸號驅動器出錯代碼(請根據驅動器使用手冊尋找相關錯誤代碼說明)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Error;
int16_t DriveError;
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n", ret);
}
else
{
    if(State == MC_AS_ERRORSTOP)
    {
        ret = ECAT_McGetAxisLastError(DeviceNo, AxisNo, &Error);
        if(ret < 0)
        {
            printf("Failed to get axis last error:%d\n", ret);
        }
        else
        {
            printf("Axis Last Error:%d\n", Error);
            if(Error == ECAT_ERR_MC_DRIVE_FAULT) //Drive fault
            {
                ret = ECAT_McGetAxisDriveError(DeviceNo, AxisNo, &DriveError);
                if(ret < 0)
                {
                    printf("Failed to get axis drive error:%d\n", ret);
                }
                else
                {
                    printf("Axis Drive Error:%d\n", DriveError);
                }
            }
        }
    }
}
```

---

```
}  
}  
}
```

---



### 7.3.17. ECAT\_McGetAxisDriveError\_Ex

**說明:**

降低使用 ECAT\_McGetAxisDriveError 所花費的時間

## 7.3.18. ECAT\_McGetAxisDI

### 說明:

取得指定軸號運動控制相關 I/O 訊號狀態。

### 格式:

```
int32_t ECAT_McGetAxisDI(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *DI)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
DI	uint32_t *	OUT	軸 I/O 訊號狀態(如表 7.6 所示) 位元若為 0:表示對應的訊號狀態為 OFF 位元若為 1:表示對應的訊號狀態為 ON

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.6: 軸 I/O 訊號狀態

位元	對應訊號
Bit 0	NOT (負方向位置極限)
Bit 1	POT (正方向位置極限)
Bit 2	ORG (原點)
Bit 3	ALM (錯誤)
Bit 4	WAN (警告)
Bit 5	SVN (伺服啟動)
Bit 6	VIR (虛擬軸)
Bit 7	INP(定位完成) 注: 需先設定定位訊號來源或定位條件後此 Bit 有效
Bit 8~31	保留

## 使用範例:

[C/C++]

```
typedef struct axis_di{
    union
    {
        struct
        {
            uint8_t NOT      : 1;      //Negative limit switch
            uint8_t POT      : 1;      //Positive limit switch
            uint8_t ORG      : 1;      //home switch
            uint8_t ALM      : 1;      //alarm
            uint8_t WAN      : 1;      //warning
            uint8_t SVN      : 1;      //serve on status
            uint8_t VIR      : 1;      //virtual axis
            uint8_t INP      : 1;      //in position
            uint32_t reserved : 24;    //Reserved(bit8~bit31)
        };
        uint32_t DIs;
    };
}axis_di_t;
```

---

```
/******  
int32_t ret;  
axis_di_t AxisDI;  
uint16_t DeviceNo = 0;  
uint16_t AxisNo = 0;  
ret = ECAT_McGetAxisDI(DeviceNo, AxisNo, &AxisDI.DIs);  
if(ret < 0)  
{  
    printf("Failed to get axis DI:%d\n", ret);  
}  
else  
{  
    printf("AxisNo[%u]-+-AxisDI\n"  
        "    |-NOT:%d\n"  
        "    |-POT:%d\n"  
        "    |-ORG:%d\n"  
        "    |-ALM:%d\n"  
        "    |-WAN:%d\n"  
        "    |-SVN:%d\n"  
        "    |-VIR:%d\n"  
        "    |-INP:%d\n"  
        "\n", AxisNo, AxisDI.NOT, AxisDI.POT, AxisDI.ORG  
        , AxisDI.ALM, AxisDI.WAN, AxisDI.SVN, AxisDI.VIR, AxisDI.INP);  
}  
*/
```

---

### 7.3.19. ECAT\_McGetAxisDI\_Ex

**說明:**

降低使用 ECAT\_McGetAxisDI 所花費的時間

### 7.3.20. ECAT\_McGetAxisDI\_Directly

**說明:**

降低使用 ECAT\_McGetAxisDI\_Ex 所花費的時間

注:需先啟用 ECAT\_McEnable\_Directly

### 7.3.21. ECAT\_McGetAxisDI\_60FD

#### 說明:

取得指定軸號 object 0x60fd Digital Input，關於每個 bit 的訊號含意需要翻閱該驅動器的說明文件。

#### 格式:

```
int32_t ECAT_McGetAxisDI_60FD(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *DI)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
DI	uint32_t *	OUT	軸 DI 訊號狀態

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t DI = 0;
ret = ECAT_McGetAxisDI_60FD(DeviceNo, AxisNo, &DI);
if(ret < 0)
{
    printf("Failed to get axis DI:%d\n", ret);
}
else
{
    printf("AxisNo[%u], AxisDI: %u \n", AxisNo, DI);
}
```

---



### 7.3.22. ECAT\_McGetAxisHomeState

**說明:**

取得指定軸號在 ECAT\_McInit 之後是否歸 HOME 過。

**格式:**

int32\_t ECAT\_McGetAxisHomeState(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t \*State)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
State	uint16_t *	OUT	0:尚未歸 home 過 1:已歸 home 過

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t State;
ret = ECAT_McGetAxisHomeState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis home state:%d\n",ret);
}
```

---

## 7.4. 單軸自動原點復歸

### 7.4.1. ECAT\_McSetAxisHomeMethod

**說明:**

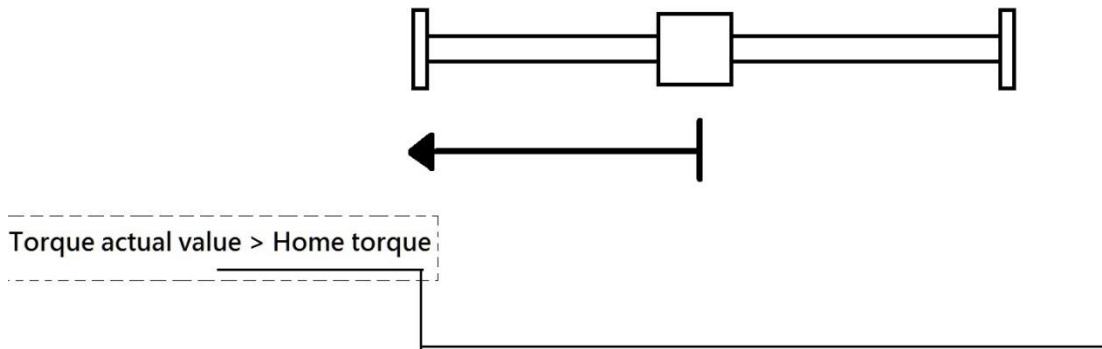
設定自動原點復歸的模式。

請查閱伺服馬達文件資料確認支援的模式。

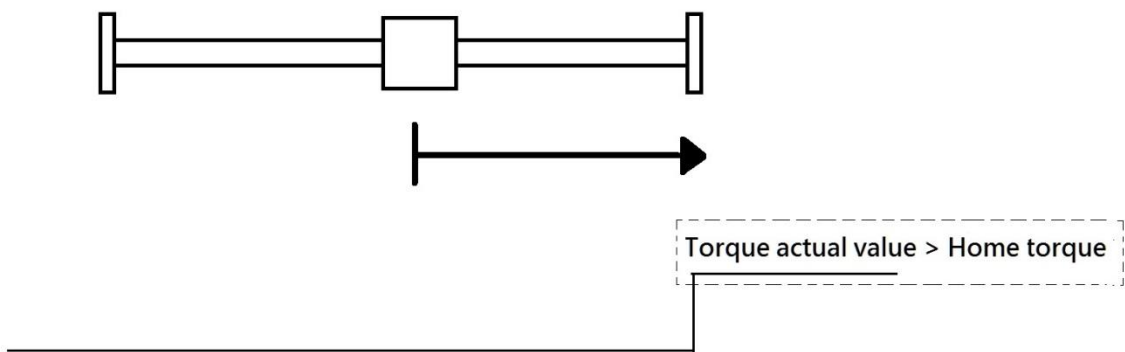
請參考”CiA402 原點復歸模式(Homing Mode/ hm mode)”

注:(1)原點復歸模式 38 39 為扭力原點復歸模式,不在 CiA402 規範當中,支援 CiA402 模組(需含有 6072h(Max torque)、6077h(Torque actual value), 需支援原點復歸模式 37)

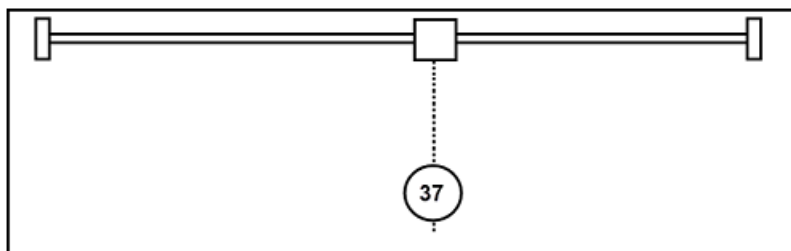
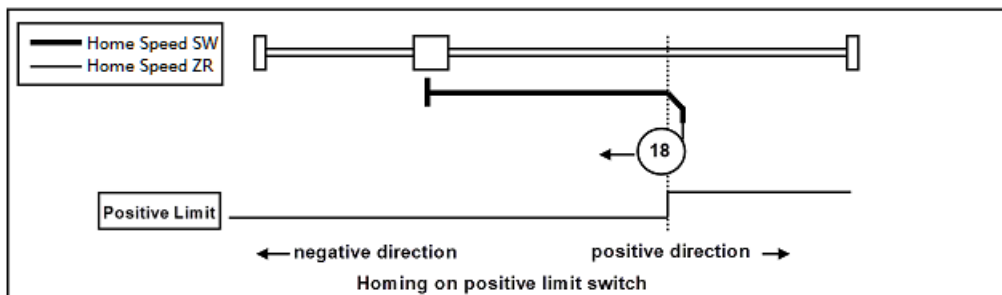
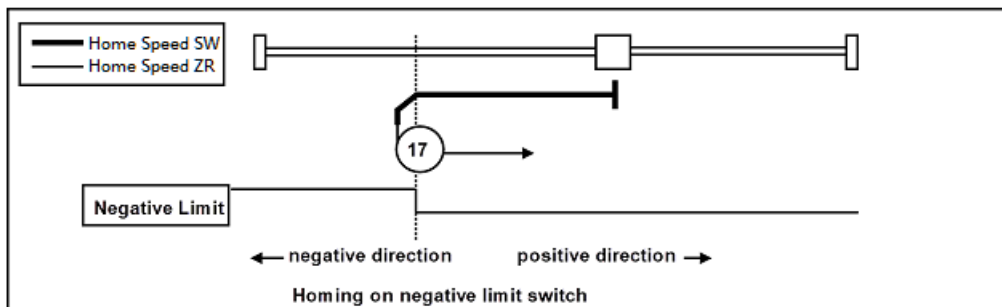
Home 38



Home 39



注:(2) ECAT-2091S/ ECAT-2094S 支援原點復歸模式 17、18、37



**格式:**

int32\_t ECAT\_McSetAxisHomeMethod(uint16\_t DeviceNo, uint16\_t AxisNo, int32\_t Method)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Method	int32_t	IN	原點復歸模式(範圍: 1~37、38~39) 請查閱伺服馬達文件資料確認模式是否支援

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int32_t Method = 1;
ret = ECAT_McSetAxisHomeMethod(DeviceNo, AxisNo, Method);
if(ret < 0)
{
    printf("Failed to set axis home method:%d\n", ret);
}
else
{
    printf("Set axis home method successfully!\n");
}
```

---

## 7.4.2. ECAT\_McGetAxisHomeMethod

### 說明:

取得自動原點復歸的模式。

### 格式:

int32\_t ECAT\_McGetAxisHomeMethod(uint16\_t DeviceNo, uint16\_t AxisNo, int32\_t \*Method)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Method	int32_t*	OUT	原點復歸模式(範圍: 1~37、38~39)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int32_t Method;
ret = ECAT_McGetAxisHomeMethod(DeviceNo, AxisNo, &Method);
if(ret < 0)
{
    printf("Failed to get axis home method:%d\n", ret);
}
else
{
    printf("Axis[%u] Home Method:%d\n", AxisNo, Method);
}
```

---



### 7.4.3. ECAT\_McSetAxisHomeSpeed

**說明:**

設定執行自動原點復歸時使用的速度。

**格式:**

int32\_t ECAT\_McSetAxisHomeSpeed(uint16\_t DeviceNo, uint16\_t AxisNo, double SeachSw, double SeachZr)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
SeachSw	double	IN	尋找原點(ORG)訊號時使用的速度 (單位: user unit/s)
SeachZr	double	IN	尋找馬達索引(EZ)訊號時使用的速度 (單位: user unit/s)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double SeachSw = 100.0;
double SeachZr = 10.0;
ret = ECAT_McSetAxisHomeSpeed(DeviceNo, AxisNo, SeachSw, SeachZr);
if(ret < 0)
{
    printf("Failed to set axis home speed:%d\n", ret);
}
else
{
    printf("Set axis home speed successfully!\n");
}
```

---

## 7.4.4. ECAT\_McGetAxisHomeSpeed

### 說明:

取得自動原點復歸時使用的速度。

### 格式:

```
int32_t ECAT_McGetAxisHomeSpeed(uint16_t DeviceNo, uint16_t AxisNo, double
*SeachSw, double *SeachZr)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
SeachSw	Double*	OUT	尋找原點(ORG)訊號時使用的速度 (單位: user unit/s)
SeachZr	Double*	OUT	尋找馬達索引(EZ)訊號時使用的速度 (單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double SeachSw;
double SeachZr;
ret = ECAT_McGetAxisHomeSpeed(DeviceNo, AxisNo, &SeachSw, &SeachZr);
if(ret < 0)
{
    printf("Failed to get axis home speed:%d\n", ret);
}
else
{
    printf("Axis[%u] Home Speed [Search Switch:%f] / [Search Zero:%f] \n", AxisNo, SeachSw, SeachZr);
}
```

---

## 7.4.5. ECAT\_McSetAxisHomeAcc

### 說明:

設定執行自動原點復歸時使用的加速度。

### 格式:

```
int32_t ECAT_McSetAxisHomeAcc(uint16_t DeviceNo, uint16_t AxisNo, double Acc)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Acc	double	IN	執行自動原點復歸時使用的加速度 (單位: user unit/s <sup>2</sup> )

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Acc = 1000.0;
ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, Acc);
if(ret < 0)
{
    printf("Failed to set axis home acceleration:%d\n", ret);
}
else
{
    printf("Set axis home acceleration successfully!\n");
}
```

---

## 7.4.6. ECAT\_McGetAxisHomeAcc

### 說明:

取得自動原點復歸時使用的加速度。

### 格式:

```
int32_t ECAT_McGetAxisHomeAcc(uint16_t DeviceNo, uint16_t AxisNo, double *Acc)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Acc	Double*	OUT	執行自動原點復歸時使用的加速度 (單位: user unit/s <sup>2</sup> )

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Acc;
ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, &Acc);
if(ret < 0)
{
    printf("Failed to get axis home acceleration:%d\n", ret);
}
else
{
    printf("Axis[%u] Home Acceleration:%f\n", AxisNo, Acc);
}
```

---

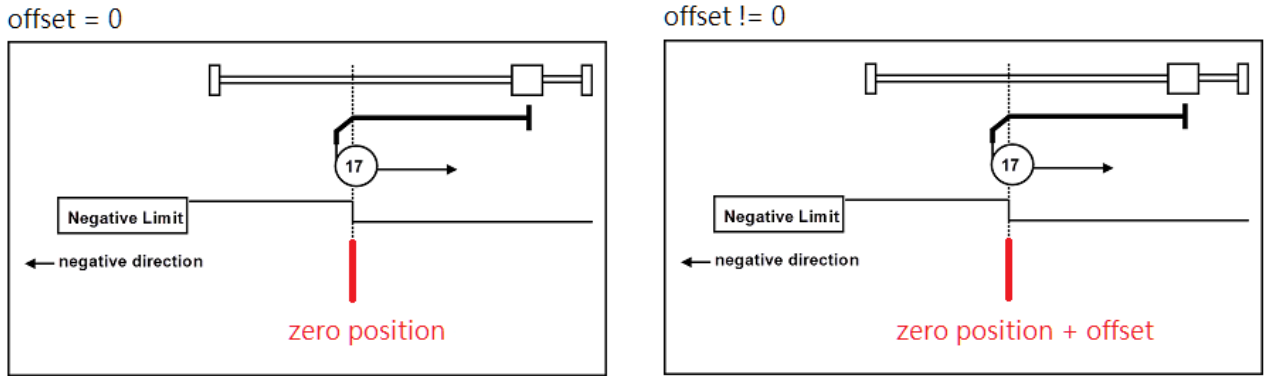


### 7.4.7. ECAT\_McSetAxisHomeOffset

**說明:**

設定原點復歸完成後位置偏移。

以歸原點方法 17 為例:



**格式:**

int32\_t ECAT\_McSetAxisHomeOffset(uint16\_t DeviceNo, uint16\_t AxisNo, double Offset)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Offset	double	IN	原點位置相對於零點位置的偏移量

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double HomeOffset = 5.0;
ret = ECAT_McSetAxisHomeOffset(DeviceNo, AxisNo, HomeOffset);
if(ret < 0)
{
    printf("Failed to set axis home offset:%d\n", ret);
}
else
{
    printf("Set axis home offset successfully!\n");
}
```

---

## 7.4.8. ECAT\_McGetAxisHomeOffset

### 說明:

取得原點復歸完成後位置偏移。

### 格式:

int32\_t ECAT\_McGetAxisHomeOffset(uint16\_t DeviceNo, uint16\_t AxisNo, double \*Offset)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Offset	Double*	OUT	原點位置相對於零點位置的偏移量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double HomeOffset;
ret = ECAT_McGetAxisHomeOffset(DeviceNo, AxisNo, &HomeOffset);
if(ret < 0)
{
    printf("Failed to get axis home offset:%d\n", ret);
}
else
{
    printf("Axis[%u] Home Offset:%f\n", AxisNo, HomeOffset);
}
```

---

## 7.4.9. ECAT\_McSetAxisHomeTorque

### 說明:

設定執行自動原點復歸時使用的扭力。

注: (1)只在原點復歸模式 38、39 時生效

(2)支援 CiA402 模組(需含有 6072h(Max torque)、6077h(Torque actual value) , 需支援原點復歸模式 37)

### 格式:

```
int32_t ECAT_McSetAxisHomeTorque(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Torque);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Torque	uint16_t	IN	執行自動原點復歸時使用的扭力 (單位: 額定扭力的 0.1%) 額定扭力可用 SDO 命令從 6076h(Motor rated torque) 讀出 當 6077h(Torque actual value)的值大於此值時, 將目前做為原點。

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int16_t Torque= 500;
ret = ECAT_McSetAxisHomeTorque(DeviceNo, AxisNo, Torque);
if(ret < 0)
{
    printf("Failed to set axis home torque:%d\n", ret);
}
```

---

## 7.4.10. ECAT\_McGetAxisHomeTorque

### 說明:

取得執行自動原點復歸時使用的扭力。

### 格式:

```
int32_t ECAT_McGetAxisHomeTorque(uint16_t DeviceNo, uint16_t AxisNo, uint16_t
*Torque);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Torque	int16_t*	OUT	執行自動原點復歸時使用的扭力 (單位: 額定扭力的 0.1%) 額定扭力可用 SDO 命令從 6076h(Motor rated torque) 讀出

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int16_t Torque;
ret = ECAT_McGetAxisHomeTorque (DeviceNo, AxisNo, &Torque);
if(ret < 0)
{
    printf("Failed to get axis home torque:%d\n", ret);
}
else
{
    printf("Axis[%u] Home torque:%d\n", AxisNo, Torque);
}
```

---



## 7.4.11. ECAT\_McSetAxisHomeStable

### 說明:

設定執行自動原點復歸時使用的位置穩定條件。

原點復歸會等待實際位置在 PosRange 範圍內，並且維持 Time\_s 後，才會將軸狀態 MC\_AS\_HOMING 轉為 MC\_AS\_STANDSTILL

### 格式:

```
int32_t ECAT_McSetAxisHomeStable(uint16_t DeviceNo, uint16_t AxisNo, double Time_s, double TimeOut_s, double PosRange)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_s	double	IN	穩定時間(單位: 秒)
TimeOut_s	double	IN	超時(單位: 秒)
PosRange	double	IN	位置穩定範圍(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Time_s = 0.5;
double TimeOut_s = 5.0;
double PosRange = 10.0;
ret = ECAT_McSetAxisHomeStable(DeviceNo, AxisNo, Time_s, TimeOut_s, PosRange);
if(ret < 0)
{
    printf("Failed to set axis home stable window:%d\n", ret);
}
else
{
    printf("Set axis home stable window successfully!\n");
}
```

---

## 7.4.12. ECAT\_McGetAxisHomeStable

### 說明:

取得自動原點復歸時使用的位置穩定條件。

### 格式:

```
int32_t ECAT_McGetAxisHomeStable(uint16_t DeviceNo, uint16_t AxisNo, double*
Time_s, double* TimeOut_s, double* PosRange)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Time_s	double	OUT	穩定時間(單位: 秒)
TimeOut_s	double	OUT	超時(單位: 秒)
PosRange	double	OUT	位置穩定範圍(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Time_s = 0.5;
double TimeOut_s = 5.0;
double PosRange = 10.0;
ret = ECAT_McGetAxisHomeStable(DeviceNo, AxisNo, &Time_s, &TimeOut_s, &PosRange);
if(ret < 0)
{
    printf("Failed to get axis home stable window:%d\n", ret);
}
else
{
    printf("Axis[%u] Home stable window [stable window:%f] / [TimeOut_s:%f] / [PosRange:%f] \n", AxisNo,
Time_s, TimeOut_s, PosRange);
}
```

---

## 7.4.13. ECAT\_McAxisHome

### 說明:

開始執行自動原點復歸

復歸方法可參考 8.5 CiA402 原點復歸模式(Homing Mode/ hm mode)。

注:(1)若相關 SDO 不存在，會導致錯誤，請看 *ECAT\_McAxisHomeEx*。

(2)此函式返回後需檢查狀態為 MC\_AS\_STANDSTILL 才算完成。

(3)原點復歸完成後，會自動設定編碼器位置與命令位置，  
若此時位置不為零，使用 *ECAT\_McAxisMoveAbs* 走到 0 即可。

### 格式:

```
int32_t ECAT_McAxisHome(uint16_t DeviceNo, uint16_t AxisNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Method = 1;
double SeachSw = 100.0;
double SeachZr = 10.0;
double Acc = 1000.0;
double HomeOffset = 5.0;

ret = ECAT_McSetAxisHomeMethod(DeviceNo, AxisNo, Method);
if(ret < 0)
{
    printf("Failed to set axis home method:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeSpeed(DeviceNo, AxisNo, SeachSw, SeachZr);
if(ret < 0)
{
    printf("Failed to set axis home speed:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, Acc);
if(ret < 0)
{
    printf("Failed to set axis home acceleration:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeOffset(DeviceNo, AxisNo, HomeOffset);
if(ret < 0)
{
    printf("Failed to set axis home offset:%d\n", ret);
```

---

```
    return;
}

ret = ECAT_McAxisHome(DeviceNo, AxisNo);
if(ret < 0)
{
    printf("Failed to start axis home:%d\n", ret);
}
else
{
    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_HOMING) //Homing

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis homing successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
}
```

---

## 7.4.14. ECAT\_McAxisHomeEx

### 說明:

開始執行自動原點復歸。

注:此命令執行時會將相關參數以 SDO 通訊的方式設定,因部分伺服馬達不支援原點覆歸相關 SDO,請查看表 7.7 並查閱伺服馬達文件資料確認相關 SDO 是否支援。

### 格式:

```
int32_t ECAT_McAxisHomeEx(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Settings)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Settings	uint16_t	IN	欲設定參數(如表 7.7 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



表 7.7 歸原點參數定義

標頭檔定義名稱	數值	說明	SDO Index
MC_AS_HOME_SPEED_SW	1	尋找原點(ORG)訊號時使用的速度	6099h:01h
MC_AS_HOME_SPEED_ZR	2	尋找馬達索引(EZ)訊號時使用的速度	6099h:02h
MC_AS_HOME_ACC	4	執行自動原點復歸時使用的加速度	609Ah:00h
MC_AS_HOME_OFFSET	8	原點位置相對於零點位置的偏移量	607Ch:00h

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Method = 1;
double SeachSw = 100.0;
double SeachZr = 10.0;
double Acc = 1000.0;
double HomeOffset = 5.0;
uint16_t Settings = MC_AS_HOME_SPEED_SW | MC_AS_HOME_SPEED_ZR | MC_AS_HOME_ACC |
MC_AS_HOME_OFFSET;

```

```
ret = ECAT_McSetAxisHomeMethod(DeviceNo, AxisNo, Method);
```

```

if(ret < 0)
{
    printf("Failed to set axis home method:%d\n", ret);
    return;
}

```

```
ret = ECAT_McSetAxisHomeSpeed(DeviceNo, AxisNo, SeachSw, SeachZr);
```

```

if(ret < 0)
{
    printf("Failed to set axis home speed:%d\n", ret);
    return;
}

```

```
ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, Acc);
if(ret < 0)
{
    printf("Failed to set axis home acceleration:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeOffset(DeviceNo, AxisNo, HomeOffset);
if(ret < 0)
{
    printf("Failed to set axis home offset:%d\n", ret);
    return;
}

ret = ECAT_McAxisHomeEx(DeviceNo, AxisNo, Settings);
if(ret < 0)
{
    printf("Failed to start axis home:%d\n", ret);
}
else
{
    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_HOMING) //Homing

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis homing successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
}
```

## 7.5. 單軸運動控制

### 7.5.1. ECAT\_McAxisErrorReset

#### 說明:

重置指定軸號的錯誤狀態，異常發生時軸將切換至錯誤狀態，請在異常排除後執行此函式重置軸的狀態。

使用此函式後之後，需要幾個 ms 才會完成重置軸狀態，可以使用 ECAT\_McGetAxisState 取得軸狀態，軸狀態為 MC\_AS\_STANDSTILL(處於 Servo On 狀態)或 MC\_AS\_DISABLED(處於 Servo off 狀態)重置軸狀態完成

#### 格式:

```
int32_t ECAT_McAxisErrorReset(uint16_t DeviceNo, uint16_t AxisNo)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n", ret);
}
else
{
    if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        ret = ECAT_McAxisErrorReset(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to reset axis error:%d\n", ret);
        }
        else
        {
            printf("Reset axis error successfully!\n");
        }
    }
}
}
```

## 7.5.2. ECAT\_McAxisMoveAbs

### 說明:

開始執行指定軸號單軸絕對位置運動控制。

### 格式:

int32\_t ECAT\_McAxisMoveAbs(uint16\_t DeviceNo, uint16\_t AxisNo, double Pos, double Vel)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double	IN	目標位置(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {

```

---

```
        printf("Axis error stop\n");  
    }  
}  
}
```

---

### 7.5.3. ECAT\_McAxisMoveRel

**說明:**

開始執行指定軸號單軸相對距離運動控制。

**格式:**

int32\_t ECAT\_McAxisMoveRel(uint16\_t DeviceNo, uint16\_t AxisNo, double Pos, double Vel)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double	IN	目標位置(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveRel(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {

```

---

```
        printf("Axis error stop\n");  
    }  
}  
}
```

---

## 7.5.4. ECAT\_McAxisMoveAbs\_P2P

### 說明:

開始執行指定軸號單軸絕對位置運動控制。

注:使用單軸最大速度執行(*ECAT\_McSetAxisMaxVelocity*)。

### 格式:

int32\_t ECAT\_McAxisMoveAbs\_P2P(uint16\_t DeviceNo, uint16\_t AxisNo, double Pos)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double	IN	目標位置(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double MaxVelocity = 100;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n", ret);
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs_P2P(DeviceNo, AxisNo, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
```

---

```
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
}
```

---

### 7.5.5. ECAT\_McAxisMoveRel\_P2P

**說明:**

開始執行指定軸號單軸相對距離運動控制。

注:使用單軸最大速度執行(*ECAT\_McSetAxisMaxVelocity*)。

**格式:**

int32\_t ECAT\_McAxisMoveRel\_P2P(uint16\_t DeviceNo, uint16\_t AxisNo, double Pos)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double	IN	目標位置(單位: user unit)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double MaxVelocity = 100;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n", ret);
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveRel_P2P(DeviceNo, AxisNo, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion
    }
}

```

```
if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
```

---



## 7.5.6. ECAT\_McAxisChangePos

### 說明:

當指定軸號處於移動中，該運動命令可用於改變終點位置。

### 格式:

```
int32_t ECAT_McAxisChangePos(uint16_t DeviceNo, uint16_t AxisNo, double Pos)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double	IN	終點位置 (單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveRel(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if (State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisPos = 20.0;
            ret = ECAT_McAxisChangePos(DeviceNo, AxisNo, AxisPos);
            if(ret < 0)
            {
                printf("Failed to call axis cahnge position function:%d\n", ret);
            }
        }
    }
}
```

```
do
{
    sleep(1);
    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
}while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
```

---

## 7.5.7. ECAT\_McAxisChangeVel

### 說明:

當指定軸號處於移動中，該運動命令可用於改變移動速度。

### 格式:

```
int32_t ECAT_McAxisChangeVel(uint16_t DeviceNo, uint16_t AxisNo, double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveRel(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if (State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisVel = 5.0;
            ret = ECAT_McAxisChangeVel(DeviceNo, AxisNo, AxisVel);
            if(ret < 0)
            {
                printf("Failed to call axis cahnge velocity function:%d\n", ret);
            }
        }
    }
}

```

```
do
{
    sleep(1);
    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
}while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
```

---

## 7.5.8. ECAT\_McAxisMoveSuperimposed

### 說明:

開始執行指定軸號單軸相對距離疊加運動控制。

### 注:

在此命令執行中 ECAT\_McAxisChangePos 及 ECAT\_McAxisChangeVel 無法使用。

### 格式:

int32\_t ECAT\_McAxisMoveSuperimposed(uint16\_t DeviceNo, uint16\_t AxisNo, double Pos, double Vel)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Pos	double	IN	目標位置 (單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 100;
double AxisPos = 6.0;
double AxisVel = 3.0;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if (State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisPos = 2.0;
            AxisVel = 2.0;
            ret = ECAT_McAxisMoveSuperimposed(DeviceNo, AxisNo, AxisPos, AxisVel);
            if(ret < 0)
            {
                printf("Failed to call AxisMoveSuperimposed function:%d\n", ret);
            }
        }
    }
}
```

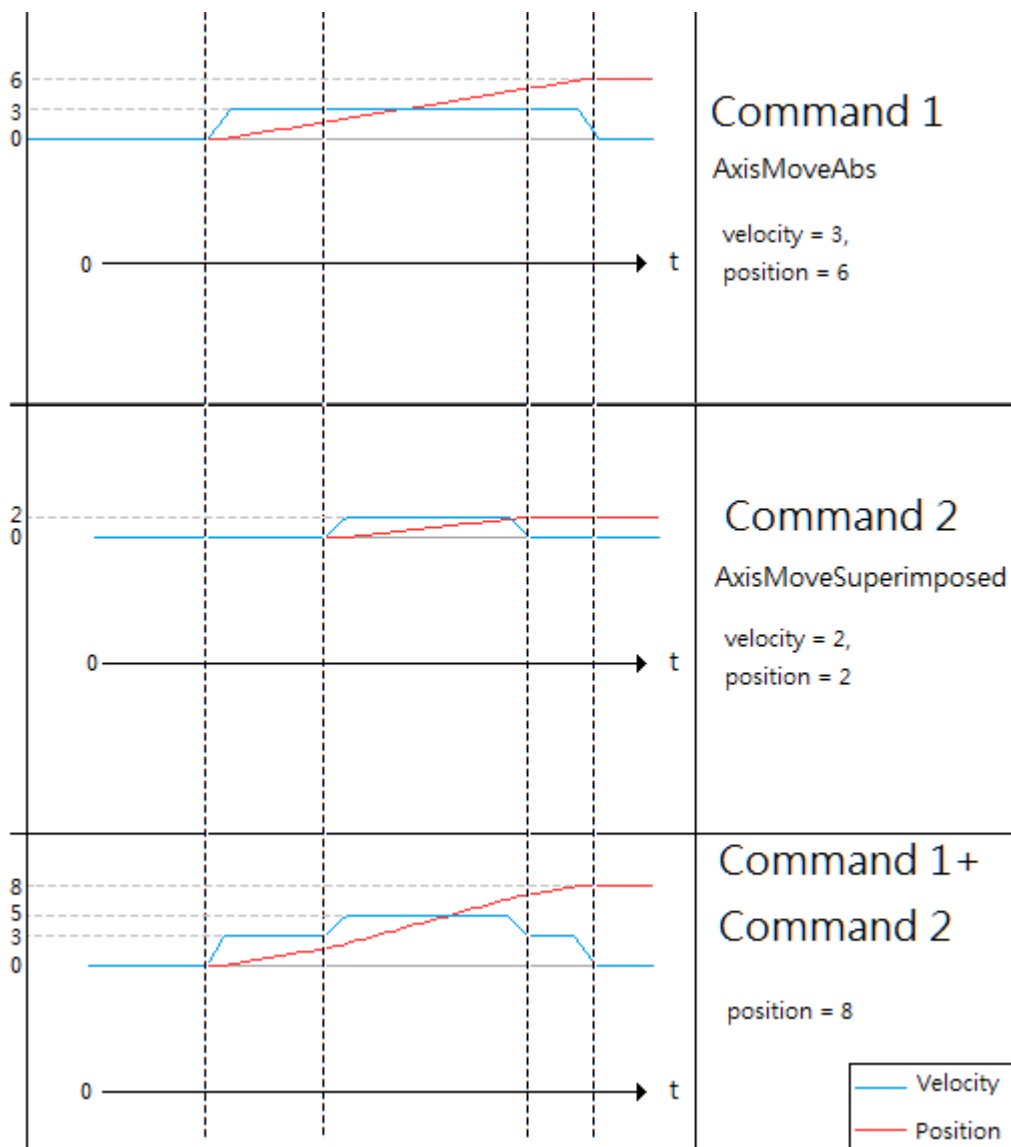


```
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
    }
}
```

---



## 7.5.9. ECAT\_McAxisHaltSuperimposed

### 說明:

停止指定軸號單軸疊加運動控制，執行此命令將減速後停止運動。

### 格式:

```
int32_t ECAT_McAxisHaltSuperimposed(uint16_t DeviceNo, uint16_t AxisNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveSuperimposed(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move superimposed:%d\n", ret);
    }
    else
    {
        sleep(1);
        ret = ECAT_McAxisHaltSuperimposed(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to stop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
            {
```

---

```
    sleep(1);
    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
}while(State == MC_AS_STOPPING) //Stopping

if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move stop successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
}
}
```

---

## 7.5.10. ECAT\_McAxisMoveVel

### 說明:

開始執行指定軸號單軸等速連續運動控制，此運動持續移動，直到停止指令(執行函式或硬體訊號)觸發終止運動。

### 格式:

```
int32_t ECAT_McAxisMoveVel(uint16_t DeviceNo, uint16_t AxisNo, double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveVel(DeviceNo, AxisNo, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move vel:%d\n", ret);
    }
}
```

## 7.5.11. ECAT\_McAxisMoveVelEx

### 說明:

開始執行指定軸號單軸等速連續運動控制，此運動持續移動，直到停止指令(執行函式或硬體訊號)觸發終止運動。

ECAT\_McAxisMoveVel 會重置加速度，若尚未到達目標速度又有新命令，此時加速度會從 0 開始計算;當加速度時間比下命令的周期還長時(例如加速度時間為 100，每 50ms 下一次命令時)，會造成無法達到目標速度的現象

ECAT\_McAxisMoveVelEx 若在加速中又有新命令，會和先前的加速度一起計算，且可以接受速度為 0 的命令

### 格式:

```
int32_t ECAT_McAxisMoveVelEx(uint16_t DeviceNo, uint16_t AxisNo, double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveVelEx(DeviceNo, AxisNo, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move vel:%d\n", ret);
    }
}
```

## 7.5.12. ECAT\_McAxisMoveVelByPos

### 說明:

利用位置控制方式，開始執行指定軸號單軸等速連續運動控制，此運動持續移動，直到停止指令(執行函式或硬體訊號)觸發終止運動。

### 格式:

```
int32_t ECAT_McAxisMoveVelByPos(uint16_t DeviceNo, uint16_t AxisNo, double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveVelByPos(DeviceNo, AxisNo, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move vel:%d\n", ret);
    }
}
```

### 7.5.13. ECAT\_McAxisMoveTor

#### 說明:

開始執行指定軸號單軸轉矩運動控制，此運動持續移動，直到停止指令(執行函式或硬體訊號)觸發終止運動。

#### 格式:

```
int32_t ECAT_McAxisMoveTor(uint16_t DeviceNo, uint16_t AxisNo, double Torque)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Torque	double	IN	轉矩 單位請參考驅動器使用手冊的物件 (Target torque: 6071h), 一般為額定轉矩的 0.1% 額定轉矩請參考 (Motor rated torque: 6076h)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisTor = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

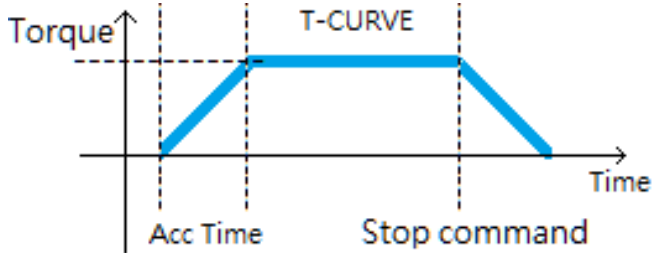
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveTor(DeviceNo, AxisNo, AxisTor);
    if(ret < 0)
    {
        printf("Failed to start axis move torque:%d\n", ret);
    }
}
```

## 7.5.14. ECAT\_McAxisMoveTorEx

### 說明:

開始執行指定軸號單軸轉矩運動控制，此運動持續移動，直到停止指令(執行函式或硬體訊號)觸發終止運動。

使用此功能後，經過 AccTime 會到達目標轉矩



ECAT\_McAxisMoveTor 會重置 AccTime, 若尚未到達目標轉矩又有新命令, 此時 AccTime 會從 0 開始計算; 當 AccTime 比下命令的周期還長時(例如 AccTime 為 100, 每 50ms 下一次命令時), 會造成無法達到目標 AccTime 的現象

ECAT\_McAxisMoveTorEx 若未到達目標轉矩又有新命令, 會和先前的 AccTime 一起計算

### 格式:

```
int32_t ECAT_McAxisMoveTorEx(uint16_t DeviceNo, uint16_t AxisNo, double Torque)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Torque	double	IN	轉矩 單位請參考驅動器使用手冊的物件 (Target torque: 6071h), 一般為額定轉矩的 0.1% 額定轉矩請參考 (Motor rated torque: 6076h)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisTor = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveTorEx(DeviceNo, AxisNo, AxisTor);
    if(ret < 0)
    {
        printf("Failed to start axis move torque:%d\n", ret);
    }
}
```



## 7.5.15. ECAT\_McAxisGearIn

### 說明:

開始執行指定軸號電子齒輪跟隨運動控制。

主軸與從軸以設定的比例及參考來源開始電子齒輪跟隨運動控制。

注:僅支援 CiA402 驅動器及虛擬軸。

### 格式:

int32\_t ECAT\_McAxisGearIn(uint16\_t DeviceNo, uint16\_t MasterNo, uint16\_t SlaveNo, int32\_t RatioNum, uint32\_t RationDen, uint16\_t SyncSource)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
MasterNo	uint16_t	IN	跟隨主軸之軸號
SlaveNo	uint16_t	IN	從軸之軸號
RatioNum	int32_t	IN	電子齒輪比分子數值
RationDen	uint32_t	IN	電子齒輪比分母數值
SyncSource	uint16_t	IN	跟隨主軸的位置參考來源(如表 7.8 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.8: 跟隨主軸的位置參考來源

標頭檔定義名稱	數值	說明
MC_AXIS_SYNC_SOURCE_SET_VALUE	0	參考主軸的理論位置
MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE	1	參考主軸的實際位置

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t RatioNum = 1;
uint32_t RationDen = 2;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;

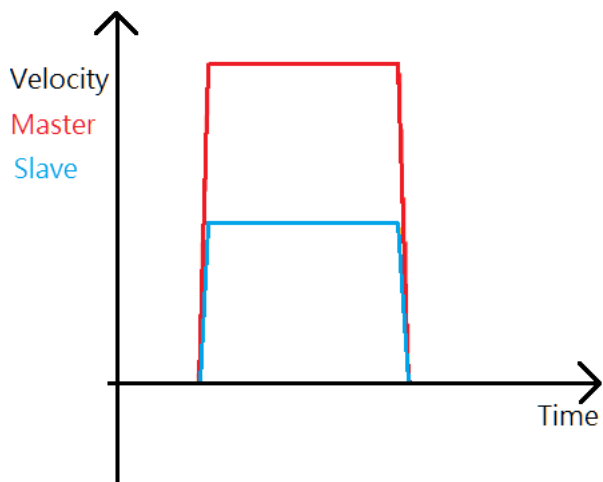
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGearIn(DeviceNo, MasterNo, SlaveNo, RatioNum, RationDen, SyncSource)
    if(ret < 0)
    {
        printf("Axis gearin is failed:%d\n", ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)

```

```
{
    printf("Failed to start axis move abs:%d\n", ret);
}
else
{
    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
}
```



## 7.5.16. ECAT\_McAxisGearOut

### 說明:

解除指定軸號電子齒輪跟隨運動控制，若 Stop 參數設定為 0，解除後指定軸號將變為等速連續運動控制，待使用者下停止指令(執行函式或硬體訊號)觸發終止運動。

### 格式:

```
int32_t ECAT_McAxisGearOut(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Stop)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從軸之軸號
Stop	uint16_t	IN	是否停止運動 0:不停止，等速連續運動 1:減速停止 2:立即停止

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 1;
uint16_t Stop = 0;
uint32_t State;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_SYNCHRONIZEDMOTION)
{
    ret = ECAT_McAxisGearOut(DeviceNo, SlaveNo, Stop)
    if(ret < 0)
    {
        printf("Axis gearout is failed:%d\n", ret);
        return;
    }
}
```

---

## 7.5.17. ECAT\_McAxisGearInByPos

### 說明:

依指定軸號開始執行飛剪運動控制。

主軸和從軸在設定的位置做跟隨運動控制，且從軸在設定的位置停止跟隨並返回開始點。

主軸可搭配 ECAT\_McSetAxisPosSoftwareLimitStatus API 使用，將 Status 參數設為 2: 啟用無限旋轉功能，並使用 ECAT\_McSetAxisPosSoftwareLimit API 設定軟體極限值為材料長度，如此就能反覆進行飛剪運動控制。

主軸需要先執行 ECAT\_McAxisMoveVel 等速連續運動控制。

加速度類型僅支援 T-Curve (線性)。

### 格式:

```
int32_t ECAT_McAxisGearInByPos(uint16_t DeviceNo, uint16_t MasterNo, uint16_t SlaveNo, int32_t RatioNum, uint32_t RationDen, uint16_t SyncSource, double MasterSynPos, double SlaveSynPos, double GearOutPos, double MaterialLength, double ReturnVel, uint16_t ReturnAcctime)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
MasterNo	uint16_t	IN	跟隨主軸之軸號
SlaveNo	uint16_t	IN	從軸之軸號
RatioNum	int32_t	IN	電子齒輪比分子數值
RationDen	uint32_t	IN	電子齒輪比分母數值
SyncSource	uint16_t	IN	跟隨主軸的位置參考來源(如表 7.8 所示)
MasterSynPos	double	IN	開始同步時的主軸位置
SlaveSynPos	double	IN	開始同步時的從軸位置
GearOutPos	double	IN	結束同步時的從軸位置

MaterialLength	double	IN	材料長度(僅用於計算從軸是否可以即時返回原點)
ReturnVel	double	IN	從軸返回時的速度
ReturnAcctime	uint16_t	IN	從軸返回時的加速度時間

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
double AxisActualVel;
double MasterSyncPos= 710;
double SlaveSyncPos= 10;
double SlaveGearOutPos= 300;
double MaterialLength= 1000;
double ReturnVel= 10;
double ReturnAcctime= 200;

int32_t RatioNum = 1;
uint32_t RationDen = 2;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveVel(DeviceNo, MasterNo, AxisVel)
    if(ret < 0)
    {
        printf("Axis move velocity failed:%d\n", ret);
        return;
    }
}

do
{
    sleep(1);
    ret = ECAT_McGetAxisActualVel(DeviceNo, AxisNo, &AxisActualVel);
}while( fabs(AxisActualVel - AxisVel) < 0.01) //constant velocity

```



```
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGearInByPos (DeviceNo, MasterNo, SlaveNo, RatioNum, RationDen, SyncSource
, MasterSyncPos, SlaveSyncPos, SlaveGearOutPos, MaterialLength, ReturnVel, ReturnAcctime)
    if(ret < 0)
    {
        printf("Axis gearin is failed:%d\n", ret);
        return;
    }
}
```

---

## 7.5.18. ECAT\_McAxisMoveProfile

### 說明:

開始執行指定軸號單軸 Profile 運動控制。

注:(1)需先使用 *ECAT\_McSetProfileData* 設定 Profile 資料

(2)根據 *ECAT\_McSetProfileInterval* 設定的間隔進行 Profile 運動控制

### 格式:

int32\_t ECAT\_McAxisMoveProfile(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t ProfileNo, uint16\_t TotalStep)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15 (資料放在內部的記憶體上，電腦斷電後會消失不見)
TotalStep	uint16_t	IN	總移動筆數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
uint32_t State;
uint16_t TotalStep = 1000;
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveProfile(DeviceNo, AxisNo, ProfileNo, TotalStep);
    if(ret < 0)
    {
        printf("Failed to start axis move profile:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

## 7.5.19. ECAT\_McAxisMoveProfileCSV

### 說明:

開始執行指定軸號單軸 Profile 運動控制，格式如圖 7.1 所示。

注:(1)需先使用 *ECAT\_McSetProfileCSV* 設定 Profile 資料。

(2)根據 *ECAT\_McSetProfileInterval* 設定的間隔進行 Profile 運動控制

### 格式:

```
int32_t ECAT_McAxisMoveProfileCSV(uint16_t DeviceNo, uint16_t AxisNo, uint16_t ProfileNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15 (資料放在內部的檔案裡，控制卡斷電後重開，資料依然存在)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
uint32_t State;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveProfileCSV(DeviceNo, AxisNo, ProfileNo);
    if(ret < 0)
    {
        printf("Failed to start axis move profile CSV:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

## 7.5.20. ECAT\_McAxisCamIn

### 說明:

開始執行指定軸號電子凸輪跟隨運動控制。

### 格式:

int32\_t ECAT\_McAxisCamIn(uint16\_t DeviceNo, uint16\_t MasterNo, uint16\_t SlaveNo, uint16\_t TableNo, uint16\_t SyncSource, double MasterInterval, double SlaveScaling)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
MasterNo	uint16_t	IN	跟隨主軸之軸號
SlaveNo	uint16_t	IN	從軸之軸號
TableNo	int16_t	IN	E-CAM 表格編號
SyncSource	uint16_t	IN	跟隨主軸的位置參考來源(如表 7.8 所示)
MasterInterval	double	IN	E-CAM 表格中主站各點位置間隔
SlaveScaling	double	IN	從站位置輸出比例

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t TableNo = 0;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[1000];
uint16_t DataSize = 1000;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.001;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;
/**
***Write E-CAM Table data to Data[1000]
**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to set E-CAM table data:%d\n", ret);
    return;
}

ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);
if(ret < 0)
{
    printf("Failed to configure E-CAM table parameter:%d\n", ret);
    return;
}
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisCamIn(DeviceNo, MasterNo, SlaveNo, TableNo
        , SyncSource, MasterInterval, SlaveScaling)

```

---

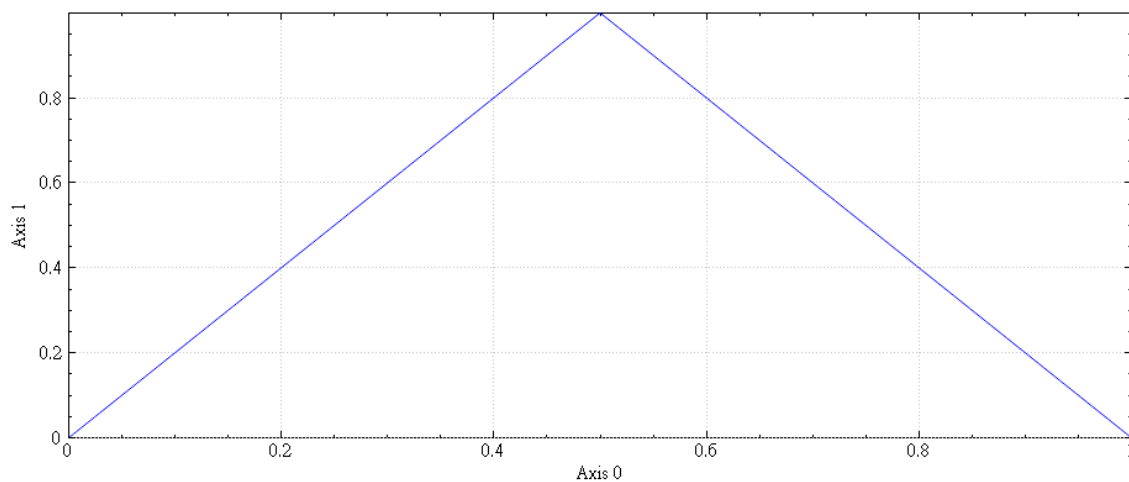
```
    if(ret < 0)
    {
        printf("Axis camin is falied:%d\n", ret);
        return;
    }
}
ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    } else {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
}
```

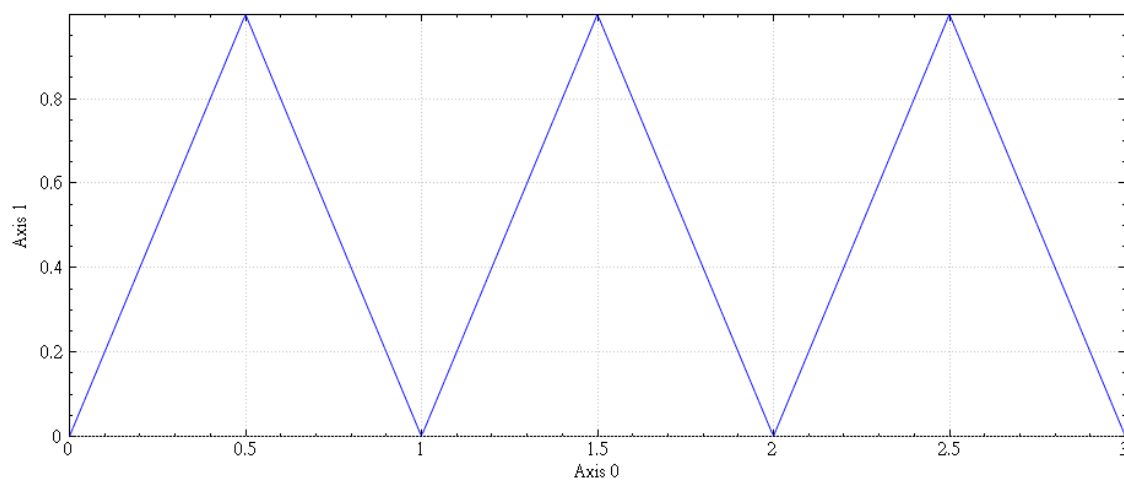
---



電子凸輪表:



電子凸輪跟隨運動控制使用示意圖:



## 7.5.21. ECAT\_McAxisCamPhaseShift

### 說明:

設定電子凸輪跟隨運動控制主站相位偏移值。

### 格式:

int32\_t ECAT\_McAxisCamPhaseShift(uint16\_t DeviceNo, uint16\_t SlaveNo, double PhaseShift)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從軸之軸號
PhaseShift	double	IN	主站相位偏移值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t TableNo = 0;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[1000];
uint16_t DataSize = 1000;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.001;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
double PhaseShift = -0.5;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;
/**
***Write E-CAM Table data to Data[1000]
**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to set E-CAM table data:%d\n", ret);
    return;
}

ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);
if(ret < 0)
{
    printf("Failed to configure E-CAM table parameter:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisCamIn(DeviceNo, MasterNo, SlaveNo, TableNo

```

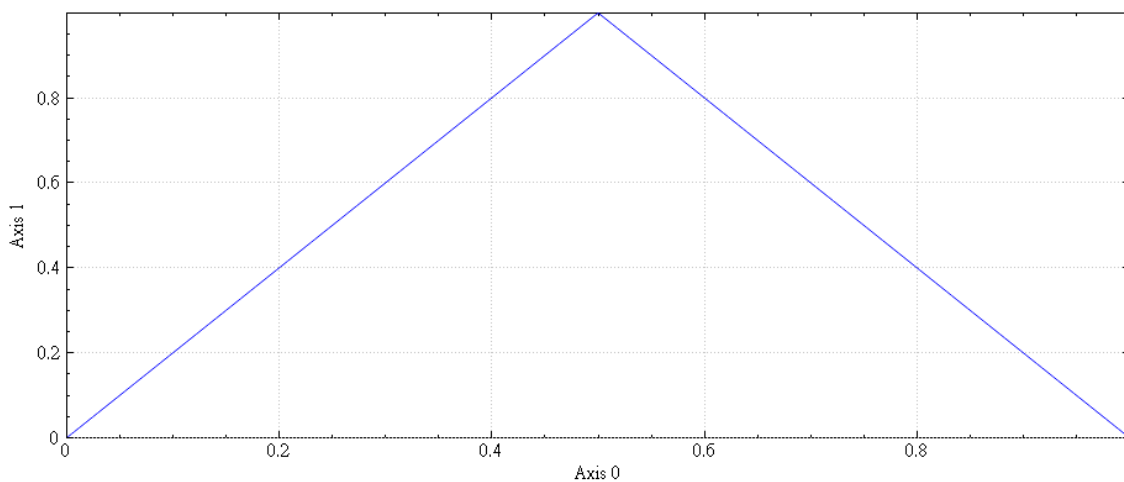
```
        , SyncSource, MasterInterval, SlaveScaling)
    if(ret < 0)
    {
        printf("Axis camin is falied:%d\n", ret);
        return;
    }
}

ret = ECAT_McAxisCamPhaseShift(DeviceNo, SlaveNo, PhaseShift)
if(ret < 0)
{
    printf("Failed to set cam phase shift:%d\n", ret);
    return;
}

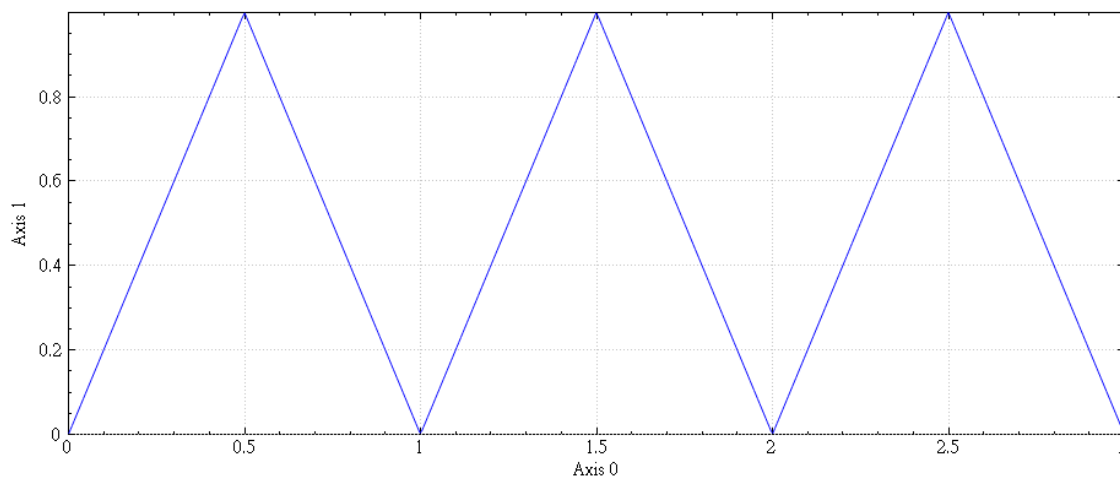
ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    } else {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
}
```

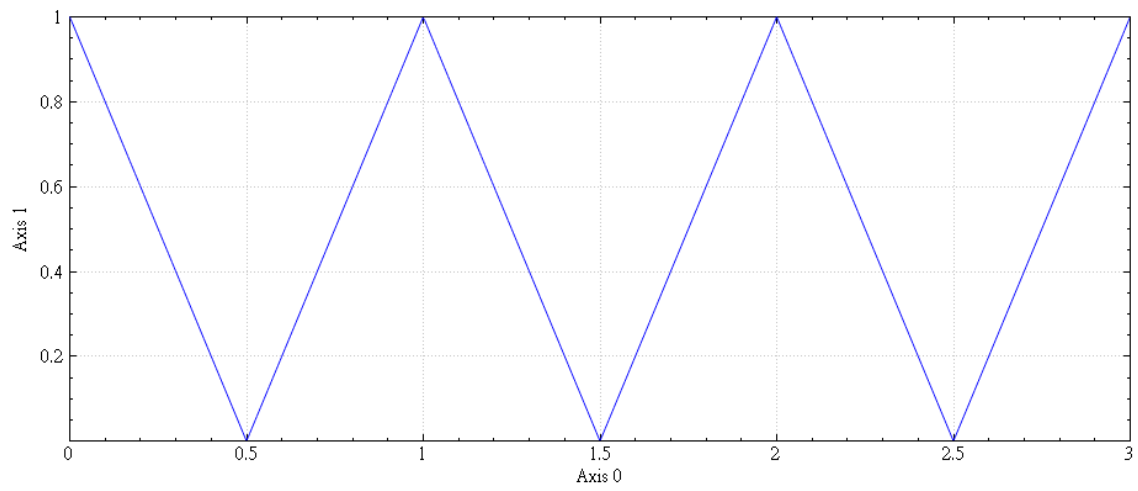
電子凸輪表:



電子凸輪跟隨運動控制使用示意圖:



電子凸輪跟隨運動控制設定主站相位偏移示意圖:



## 7.5.22. ECAT\_McAxisCamOut

### 說明:

停止指定軸號電子凸輪跟隨運動控制。

### 格式:

int32\_t ECAT\_McAxisCamOut(uint16\_t DeviceNo, uint16\_t SlaveNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從軸之軸號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;

ret = ECAT_McAxisCamOut(DeviceNo, SlaveNo);
if (ret != 0) {
    printf("Failed to cam out:%d\n", ret);
}
```

---



### 7.5.23. ECAT\_McAxisGantryIn

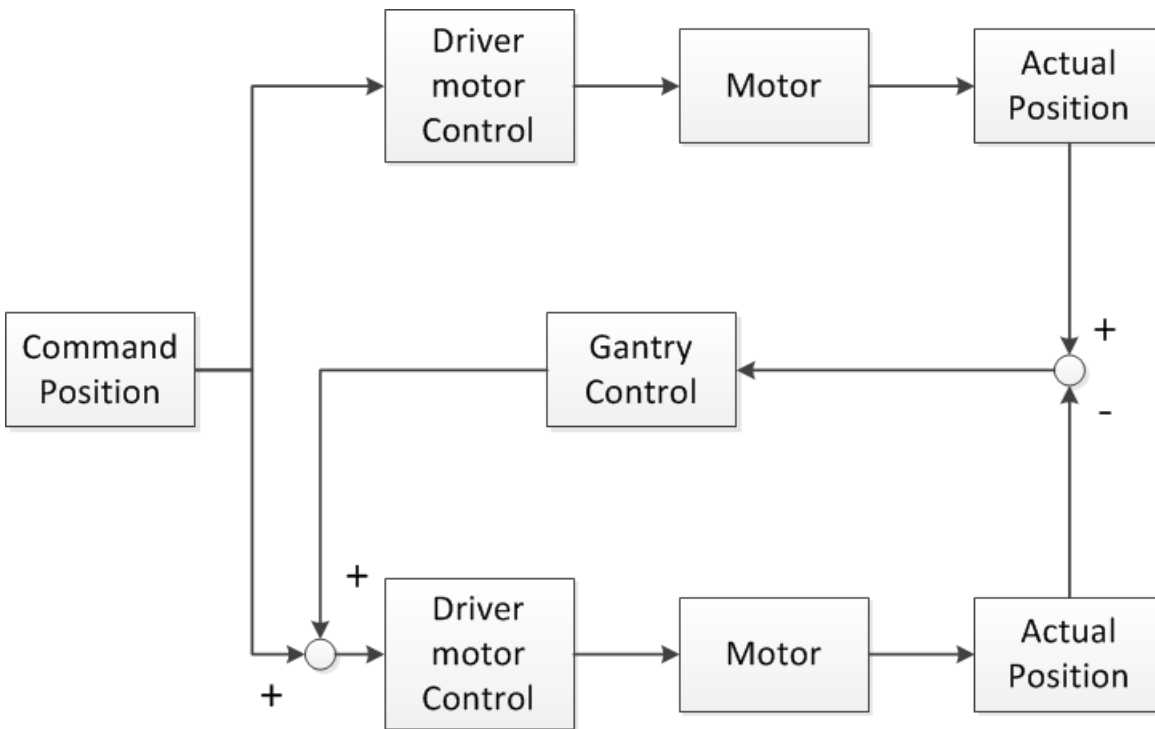
**說明:**

開始執行指定軸號龍門控制，從軸跟隨主軸**位置命令**移動。

注: (1)使用 ECAT\_McAxisGantryOut 停止指定軸號龍門控制。

(2)主軸軸號必須比從軸軸號小，否則**位置命令**會落後一個 CycleTime。

(3) home 217~230、235、237 支援龍門歸原點，主軸 servo on 且從軸 servo on，gantry in 後，對主軸下歸原點，歸原點完成後位置不會清除，只需對主軸使用 ECAT\_McSetAxisActualPosition 設定為 0，成功後主軸與從軸位置都會為 0(韌體版本須為 1.0.27 或以上)



主軸及從軸的龍門控制迴路經過 PI 控制器後加至從軸命令位置。

**格式:**

int32\_t ECAT\_McAxisGantryIn(uint16\_t DeviceNo, uint16\_t MasterNo, uint16\_t SlaveNo, int32\_t Direction)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
MasterNo	uint16_t	IN	跟隨主軸之軸號
SlaveNo	uint16_t	IN	從軸之軸號
Direction	int32_t	IN	主軸與從軸運動方向 1:相同 -1:相反

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
    if(ret < 0)
    {
        printf("Axis gantryin is failed:%d\n", ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

```

```
if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
```

---

## 7.5.24. ECAT\_McAxisGantryMaxPosDiff

### 說明:

設置龍門控制主軸及從軸的最大位置誤差量，位置相差超過設定值時運動停止。

### 格式:

int32\_t ECAT\_McAxisGantryMaxPosDiff(uint16\_t DeviceNo, uint16\_t SlaveNo, double Value)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從軸之軸號
Value	double	IN	最大位置誤差量(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;
double Value = 1;
uint16_t Status = 1;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
    if(ret < 0)
    {
        printf("Axis gantryin is failed:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiff(DeviceNo, SlaveNo, Value)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation is failed:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiffStatus(DeviceNo, SlaveNo, Status)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation Status is failed:%d\n", ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);

```

---

```
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

---

## 7.5.25. ECAT\_McAxisGantryMaxPosDiffStatus

### 說明:

設置龍門控制主軸及從軸的最大位置誤差量狀態。

### 格式:

```
int32_t ECAT_McAxisGantryMaxPosDiffStatus(uint16_t DeviceNo, uint16_t SlaveNo,
uint16_t Status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從軸之軸號
Status	uint16_t	IN	0:不啟用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;
double Value = 1;
uint16_t Status = 1;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
    if(ret < 0)
    {
        printf("Axis gantryin is failed:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiff(DeviceNo, SlaveNo, Value)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation is failed:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiffStatus(DeviceNo, SlaveNo, Status)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation Status is failed:%d\n", ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);

```

---

```
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

---

### 7.5.26. ECAT\_McAxisGantryGain

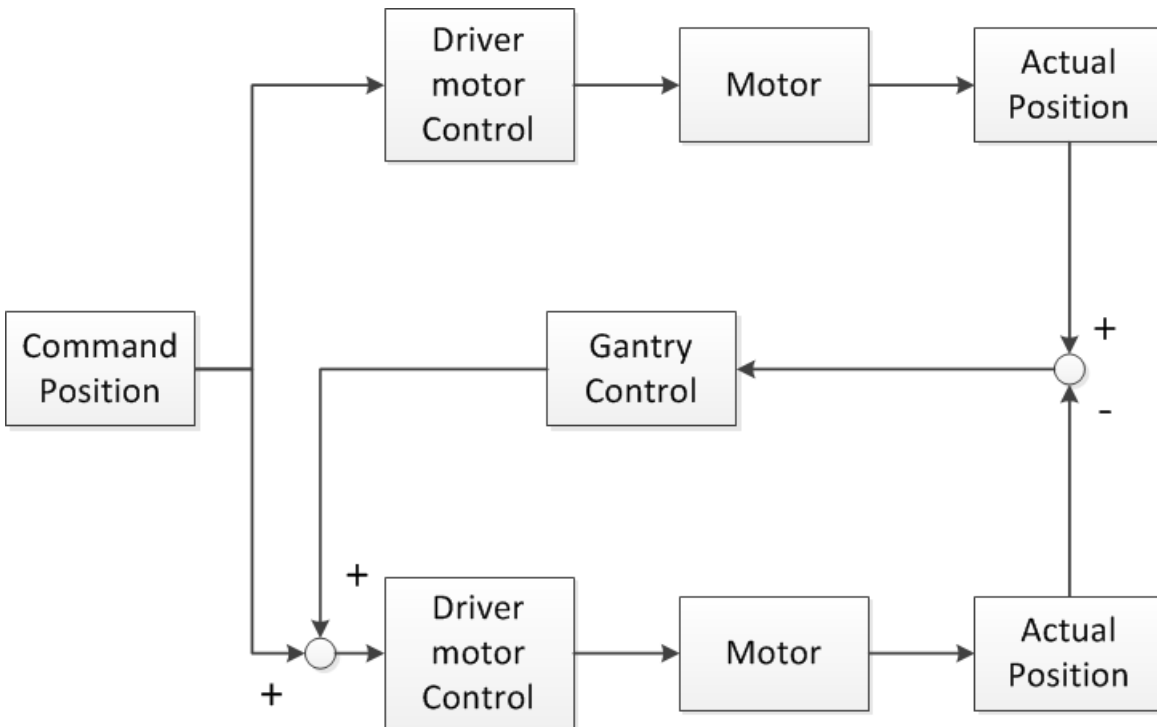
**說明:**

參數不理想可能導致震盪，請小心設定，可以使用 ECAT\_McAxisGantryMaxPosDiff 定最大位置誤差量或使用緊急停止，當位置震盪/發散時可以及時停止。

可使用 Gantry Utility 進行調整(Firmware Ver.需要在 1.0.24 或以上)。

Gantry Utility 為泓格所開發的工具軟體，

軟體及手冊的下載請參考”軟體安裝”章節



主軸及從軸的龍門控制迴路經過 PI 控制器後加至從軸命令位置。

注:(1)參數從零開始，每次增加 0.1，參數太大可能會引發震盪，請小心設定。

**格式:**

int32\_t ECAT\_McAxisGantryGain(uint16\_t DeviceNo, uint16\_t SlaveNo, double Kp, double Ki)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
SlaveNo	uint16_t	IN	從軸之軸號
Kp	double	IN	速度控制迴路的比例項係數
Ki	double	IN	速度控制迴路的積分項係數

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;
double Value = 1;
uint16_t Status = 1;
double Kp = 0.1;
double Ki = 0.1;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
    if(ret < 0)
    {
        printf("Axis gantryin is falied:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiff(DeviceNo, SlaveNo, Value)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation is falied:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiffStatus(DeviceNo, SlaveNo, Status)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation Status is falied:%d\n", ret);
        return;
    }
}

```

---

```
ret = ECAT_McAxisGantryGain(DeviceNo, SlaveNo, Kp, Ki)
if(ret < 0)
{
    printf("Set Axis gantry Gain is failed:%d\n", ret);
    return;
}

}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
}
```

---

## 7.5.27. ECAT\_McAxisGantryOut

### 說明:

停止指定軸號龍門控制。

### 格式:

int32\_t ECAT\_McAxisGantryOut(uint16\_t DeviceNo, uint16\_t AxisNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	龍門從軸之軸號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_SYNCHRONIZEDMOTION)
{
    ret = ECAT_McAxisGantryOut(DeviceNo, SlaveNo)
    if(ret < 0)
    {
        printf("Axis gantryout is failed:%d\n", ret);
        return;
    }
}
```

---



## 7.5.28. ECAT\_McAxisMoveAbsAdv\_Ex

### 說明:

取代舊的函式"ECAT\_McAxisMoveAbsAdv"

開始執行指定軸號單軸絕對位置運動控制。

### 格式:

```
int32_t ECAT_McAxisMoveAbsAdv_Ex(uint16_t DeviceNo, uint16_t AxisNo, double
EndPos, double StartVel, double ReqVel, double FinalVel, double Accel, double Decel,
uint8_t AccDecMode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
EndPos	double	IN	目標位置(單位: user unit)
StartVel	double	IN	初始速度(單位: user unit/s)
ReqVel	double	IN	移動速度(單位: user unit/s)
FinalVel	double	IN	結束速度(單位: user unit/s)
Accel	double	IN	加速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
Decel	double	IN	減速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
AccDecMode	uint8_t	IN	加減速模式: 0: 加減速度率(user unit/s <sup>2</sup> ) 1: 加減速度時間(second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
    final_vel = 2;
    accel = 4;
    decel = 4;
    end_pos = 2;
    ret = ECAT_McAxisMoveAbsAdv_Ex(DeviceNo, AxisNo, end_pos,
        start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 2
    start_vel = 2;
    req_vels = 2.5;
    final_vel = 1.5;
    accel = 6;
    decel = 6;
    end_pos = 5;
    ret = ECAT_McAxisMoveAbsAdv_Ex(DeviceNo, AxisNo, end_pos,

```

---

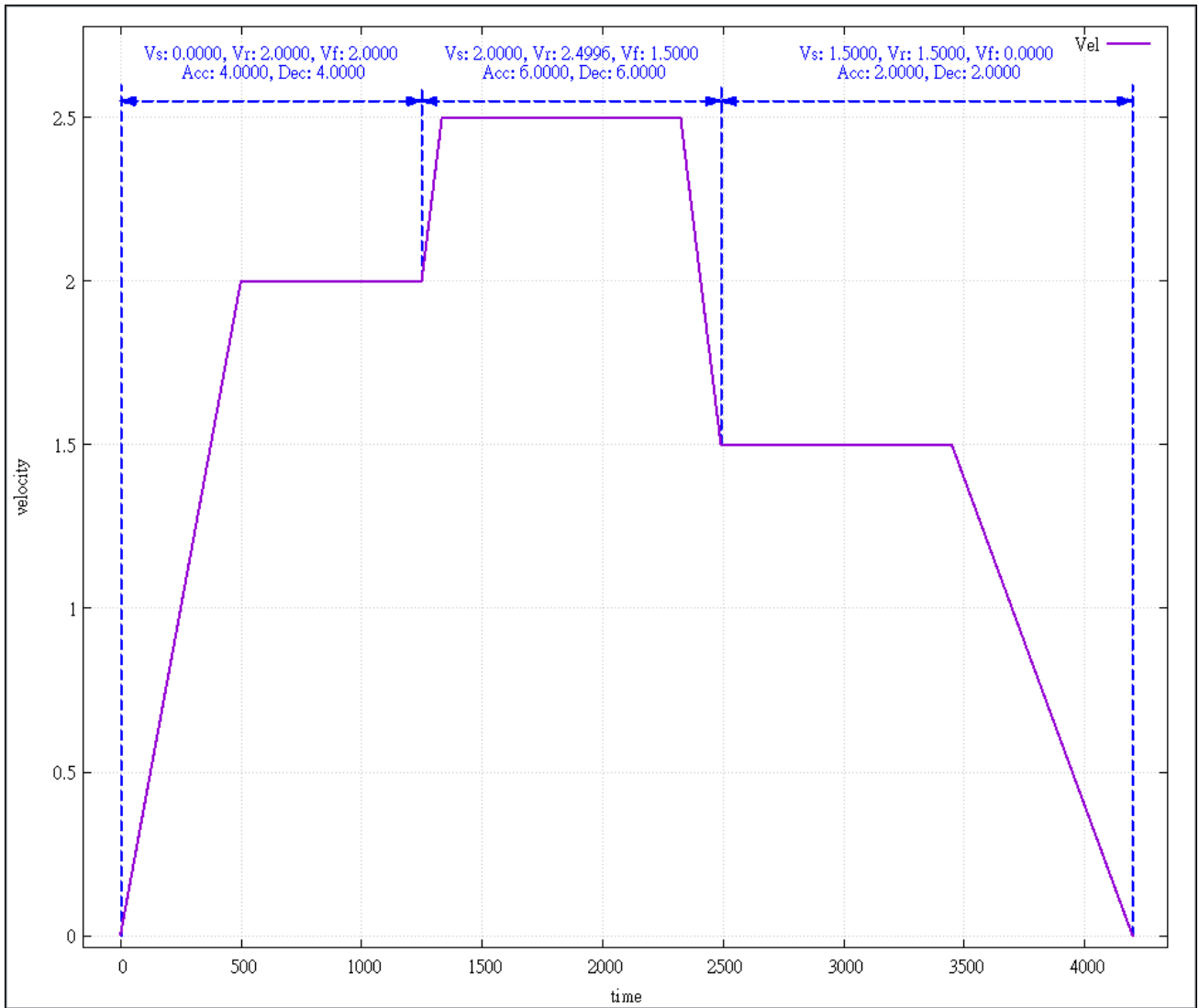
```
        start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 3
    start_vel = 1.5;
    req_vels = 1.5;
    final_vel = 0;
    accel = 2;
    decel = 2;
    end_pos = 7;
    ret = ECAT_McAxisMoveAbsAdv_Ex(DeviceNo, AxisNo, end_pos,
        start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
```

---



## 7.5.29. ECAT\_McAxisMoveRelAdv\_Ex

### 說明:

取代舊的函式"ECAT\_McAxisMoveRelAdv"

開始執行指定軸號單軸相對位置運動控制。

### 格式:

```
int32_t ECAT_McAxisMoveRelAdv_Ex(uint16_t DeviceNo, uint16_t AxisNo, double
EndPos, double StartVel, double ReqVel, double FinalVel, double Accel, double Decel,
uint8_t AccDecMode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
EndPos	double	IN	目標位置(單位: user unit)
StartVel	double	IN	初始速度(單位: user unit/s)
ReqVel	double	IN	移動速度(單位: user unit/s)
FinalVel	double	IN	結束速度(單位: user unit/s)
Accel	double	IN	加速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
Decel	double	IN	減速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
AccDecMode	uint8_t	IN	加減速模式: 0: 加減速度率(user unit/s <sup>2</sup> ) 1: 加減速度時間(second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
    final_vel = 2;
    accel = 4;
    decel = 4;
    end_pos = 2;
    ret = ECAT_McAxisMoveAbsAdv_Ex(DeviceNo, AxisNo, end_pos,
        start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 2
    start_vel = 2;
    req_vels = 2.5;
    final_vel = 1.5;
    accel = 6;
    decel = 6;
    end_pos = 3;
    ret = ECAT_McAxisMoveRelAdv_Ex(DeviceNo, AxisNo, end_pos,

```

---

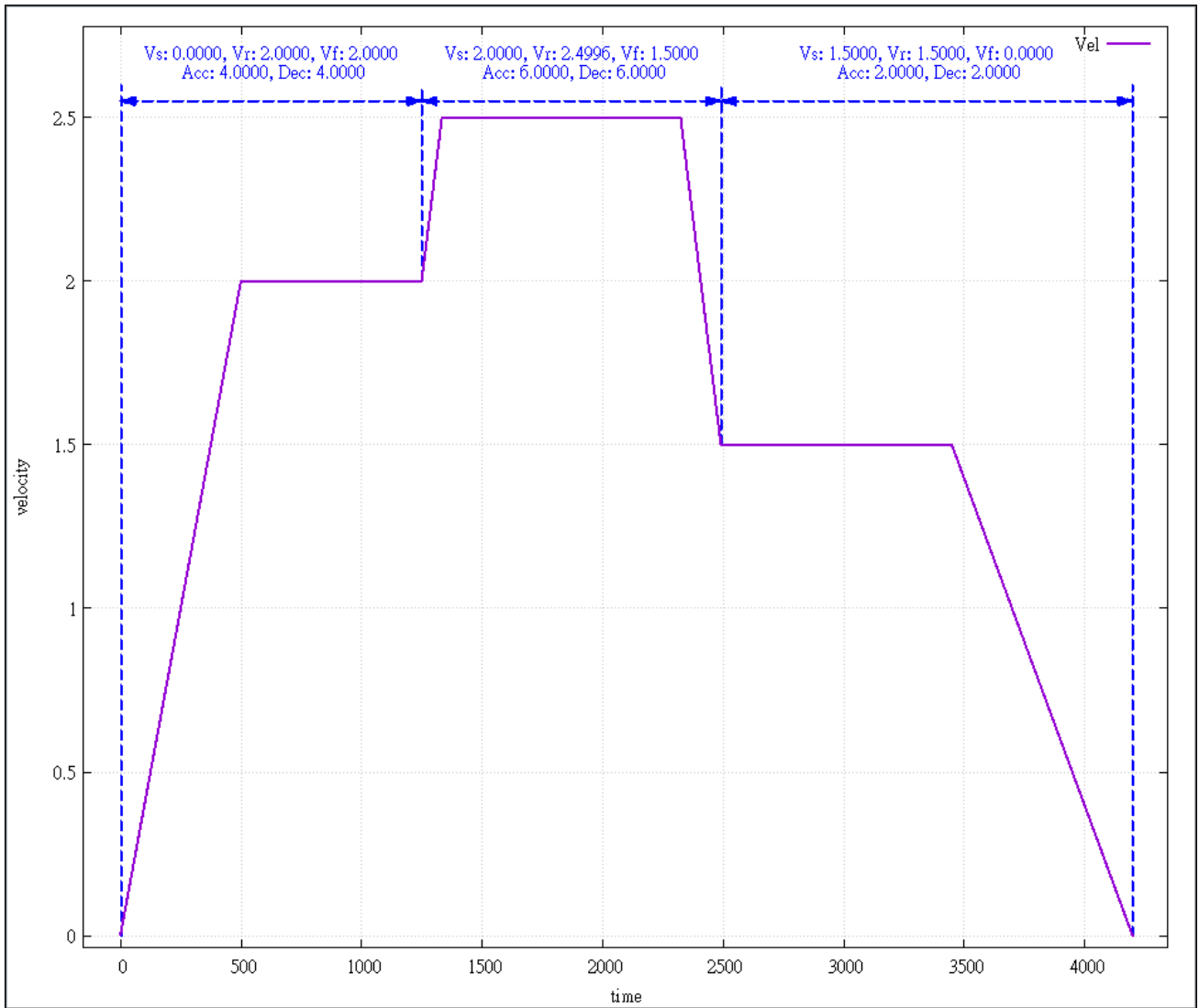
```
        start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 3
    start_vel = 1.5;
    req_vels = 1.5;
    final_vel = 0;
    accel = 2;
    decel = 2;
    end_pos = 2;
    ret = ECAT_McAxisMoveRelAdv_Ex(DeviceNo, AxisNo, end_pos,
        start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
```

---





### 7.5.30. ECAT\_McAxisMove\_CiA402\_PP

**說明:**

開始執行單軸 CiA402 位置模式運動控制。

注: 此函式包含 SDO 命令

**格式:**

int32\_t ECAT\_McAxisMove\_CiA402\_PP(uint16\_t DeviceNo, uint16\_t AxisNo, uint8\_t Abort, uint8\_t AbsMove, double EndPos, double Vel, double Accel, double Decel)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Abort	uint8_t	IN	中止當前命令
AbsMove	uint8_t	IN	絕對移動模式: 0: 相對移動 1: 絕對移動
EndPos	double	IN	目標位置(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)
Accel	double	IN	加速度率(user unit/s <sup>2</sup> )
Decel	double	IN	減速度率(user unit/s <sup>2</sup> )

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double req_vel;
double accel;
double decel;
double end_pos;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    req_vels = 2;
    accel = 4;
    decel = 4;
    end_pos = 2;
    ret = ECAT_McAxisMove_CiA402_PP(DeviceNo, AxisNo, 0, 1,
        end_pos, req_vel, accel, decel);

    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 2
    req_vels = 2.5;
    accel = 6;
    decel = 6;
    end_pos = 3;
    ret = ECAT_McAxisMove_CiA402_PP(DeviceNo, AxisNo, 0, 1,
        end_pos, req_vel, accel, decel);

    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    do
    {
        sleep(1);
```

---

```
    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
}while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}

}
```

---

### 7.5.31. ECAT\_McAxisMove\_CiA402\_PV

**說明:**

開始執行單軸 CiA402 速度模式運動控制。

注: 此函式包含 SDO 命令

**格式:**

int32\_t ECAT\_McAxisMove\_CiA402\_PV(uint16\_t DeviceNo, uint16\_t AxisNo, double Vel, double Accel, double Decel)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Vel	double	IN	移動速度(單位: user unit/s)
Accel	double	IN	加速度率(user unit/s <sup>2</sup> )
Decel	double	IN	減速度率(user unit/s <sup>2</sup> )

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double req_vel;
double accel;
double decel;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    req_vels = 2;
    accel = 4;
    decel = 4;
    ret = ECAT_McAxisMove_CiA402_PV(DeviceNo, AxisNo, req_vel, accel, decel);
    if (ret != 0)
        printf("Failed to start move:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}

```



## 7.5.32. ECAT\_McAxisMove\_CiA402\_PT

### 說明:

開始執行單軸 CiA402 扭力模式運動控制。

注: 此函式包含 SDO 命令

### 格式:

int32\_t ECAT\_McAxisMove\_CiA402\_PT(uint16\_t DeviceNo, uint16\_t AxisNo, double Torque, double Slope)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
Torque	double	IN	目標扭力值，最大扭力值的千分比，數值 1~1000
Slope	double	IN	扭力上升率(0.1%/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double torque, double slope;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    torque = 50;
    slope = 10;

    ret = ECAT_McAxisMove_CiA402_PT(DeviceNo, AxisNo, torque, slope);
    if (ret != 0)
        printf("Failed to start move:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
    }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

    if(State == MC_AS_STANDSTILL) //StandStill
        printf("Axis move successfully!\n");
    else if(State == MC_AS_ERRORSTOP) //ErrorStop
    {
        printf("Axis error stop\n");
    }
}
}
```

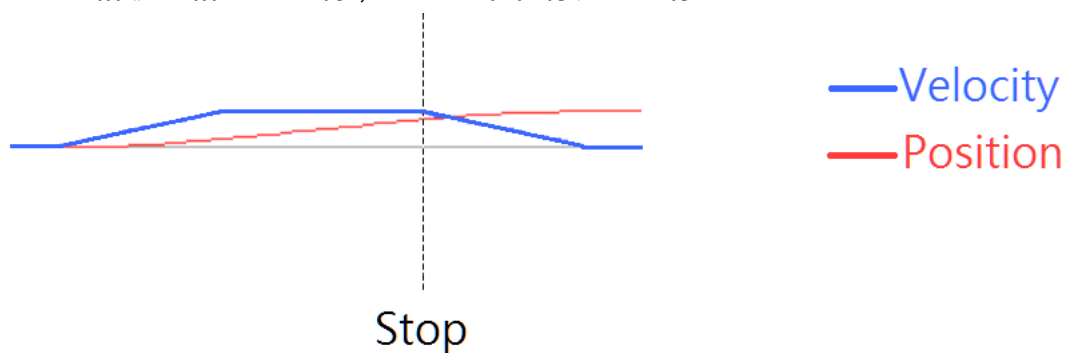




### 7.5.33. ECAT\_McAxisStop

**說明:**

停止指定軸號單軸運動控制，執行此命令將減速後停止運動。



**格式:**

```
int32_t ECAT_McAxisStop(uint16_t DeviceNo, uint16_t AxisNo)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(1);
        ret = ECAT_McAxisStop(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to stop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
            {
```

---

```
    sleep(1);
    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
}while(State == MC_AS_STOPPING) //Stopping

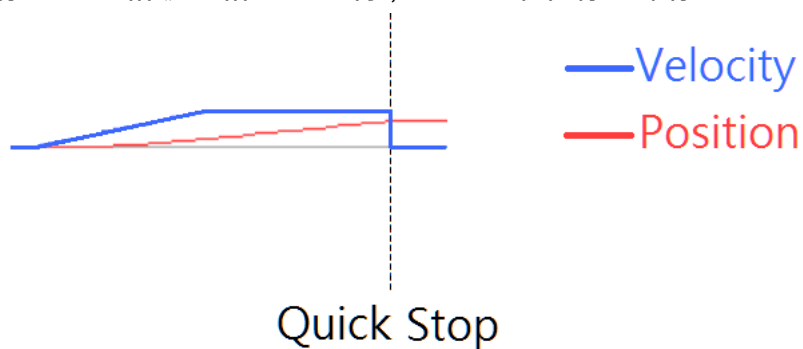
if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move stop successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
}
}
```

---

### 7.5.34. ECAT\_McAxisQuickStop

**說明:**

停止指定軸號單軸運動控制，執行此命令將立即停止運動。



**格式:**

int32\_t ECAT\_McAxisQuickStop(uint16\_t DeviceNo, uint16\_t AxisNo)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號

**回傳值:**

- 0: 函式執行成功。
- 其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(1);
        ret = ECAT_McAxisQuickStop(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to quickstop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
            {
```

---

```
    sleep(1);
    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
}while(State == MC_AS_STOPPING) //Stopping

if(State == MC_AS_STANDSTILL) //StandStill
    printf("Axis move stop successfully!\n");
else if(State == MC_AS_ERRORSTOP) //ErrorStop
{
    printf("Axis error stop\n");
}
}
}
}
```

---

## 7.6. 群組運動控制設定

### 7.6.1. ECAT\_McAddAxisToGroup\_Ex

**說明:**

取代舊的函式"ECAT\_McAddAxisToGroup"  
將指定軸加入群組。

**格式:**

int32\_t ECAT\_McAddAxisToGroup\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t AxisNo)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號(最大可使用群組編號範圍定義於標頭檔 MC_GROUP_NO_MAX)
AxisNo	uint16_t	IN	指定軸號

**回傳值:**

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
AxisNo = 0;
ret = ECAT_McAddAxisToGroup_Ex(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup_Ex(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
```

---

## 7.6.2. ECAT\_McRemoveAxisFromGroup\_Ex

### 說明:

取代舊的函式"ECAT\_McRemoveAxisFromGroup"  
將指定軸由群組中移除。

### 格式:

int32\_t ECAT\_McRemoveAxisFromGroup\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t AxisNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
AxisNo	uint16_t	IN	指定軸號

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
AxisNo = 1;
ret = ECAT_McRemoveAxisFromGroup_Ex(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to remove axis from group:%d\n", ret);
}
else
{
    printf("Remove axis from group successfully!\n");
}
```

---

### 7.6.3. ECAT\_McUngroupAllAxes\_Ex

**說明:**

取代舊的函式"ECAT\_McRemoveAxisFromGroup"  
移除群組所有軸。

**格式:**

int32\_t ECAT\_McRemoveAxisFromGroup\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo,  
uint16\_t AxisNo)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號

**回傳值:**

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
ret = ECAT_McUngroupAllAxes_Ex(DeviceNo, GroupNo);
if(ret < 0)
{
    printf("Failed to ungroup all axes:%d\n", ret);
}
else
{
    printf("Ungroup all axes successfully!\n");
}
```

---

## 7.6.4. ECAT\_McSetGroupCmdMode\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupCmdMode"

設定群組命令的執行模式，立即切換。

注:命令執行到一半時可直接切換模式。

### 格式:

```
int32_t ECAT_McSetGroupCmdMode_Ex(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t CmdMode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
CmdMode	uint16_t	IN	群組命令執行模式編號(如表 7.9 所示) default: BUFFERED Mode

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.9: 群組命令執行模式編號

標頭檔定義名稱	數值	說明
MS_GRP_CM_ABORTING	0	命令緩衝區會被全部清除，同時，新的命令會中止目前執行的運動命令並且馬上執行新命令的模式
MS_GRP_CM_BUFFERED	1	新的命令會先在命令緩衝區排隊，所有命令依序執行。等到目前運動命令完成後(減速停止)才執行下一個命令的模式
MS_GRP_CM_BLENDED	2	新的命令會先在命令緩衝區排隊等待，所有命令依序執行。但是執行中的命令在進入減速段的瞬間會同時執行下一個命令，使前後兩運動命令部分混合執行的命令模式

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t CmdMode = MS_GRP_CM_BUFFERED;
ret = ECAT_McSetGroupCmdMode_Ex(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
}
else
{
    printf("Set group command mode successfully!\n");
}

```

## 7.6.5. ECAT\_McSetGroupCmdModeEx\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupCmdModeEx"

設定群組命令的執行模式，排隊切換。

注:將此命令放進命令緩衝區排隊，在 MS\_GRP\_CM\_ABORTING 模式下立即切換模式。

### 格式:

```
int32_t ECAT_McSetGroupCmdModeEx_Ex(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t CmdMode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
CmdMode	uint16_t	IN	群組命令執行模式編號(如表 7.9 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



表 7.10: 群組命令執行模式編號

標頭檔定義名稱	數值	說明
MS_GRP_CM_ABORTING	0	命令緩衝區會被全部清除，同時，新的命令會中止目前執行的運動命令並且馬上執行新命令的模式
MS_GRP_CM_BUFFERED	1	新的命令會先在命令緩衝區排隊，所有命令依序執行。等到目前運動命令完成後(減速停止)才執行下一個命令的模式
MS_GRP_CM_BLENDED	2	新的命令會先在命令緩衝區排隊等待，所有命令依序執行。但是執行中的命令在進入減速段的瞬間會同時執行下一個命令，使前後兩運動命令部分混合執行的命令模式

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t CmdMode = MS_GRP_CM_BUFFERED;
ret = ECAT_McSetGroupCmdModeEx_Ex(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
}
else
{
    printf("Set group command mode successfully!\n");
}

```

## 7.6.6. ECAT\_McGetGroupCmdMode

### 說明:

取得群組命令的執行模式。

### 格式:

```
int32_t ECAT_McGetGroupCmdMode(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t *CmdMode)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
CmdMode	uint16_t*	OUT	群組命令執行模式編號(如表 7.9 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t CmdMode;
ret = ECAT_McGetGroupCmdMode(DeviceNo, GroupNo, &CmdMode);
if(ret < 0)
{
    printf("Failed to get group command mode:%d\n", ret);
}
else
{
    printf("Group[%u] Command Mode:%u\n", GroupNo, CmdMode);
}
```

---

## 7.6.7. ECAT\_McSetGroupAccTime\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupAccTime"

設定群組執行群組運動時使用的加速度時間。時間愈短則加速度愈大。使用線性(梯形)加減速時會使用此函式所設定時間做加速或是減速；如果是 S 型加減速類型，則真正使用時間會變成此值的兩倍。

### 格式:

```
int32_t ECAT_McSetGroupAccTime_Ex(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Time_ms)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Time_ms	uint16_t	IN	執行群組運動時使用的加速度時間 (單位: 毫秒) default:100

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Time_ms = 500;
ret = ECAT_McSetGroupAccTime_Ex(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set group acceleration time:%d\n", ret);
}
else
{
    printf("Set group acceleration time successfully!\n");
}
```

---

## 7.6.8. ECAT\_McSetGroupAccTimeEx

### 說明:

設定群組執行群組運動時使用的加速度時間。時間愈短則加速度愈大。使用線性(梯形)加減速時會使用此函式所設定時間做加速或是減速；如果是 S 型加減速類型，則真正使用時間會變成此值的兩倍。

注:此命令可在運動中使用，下一個命令改變加速度時間。

### 格式:

```
int32_t ECAT_McSetGroupAccTimeEx(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Time_ms)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Time_ms	uint16_t	IN	執行群組運動時使用的加速度時間 (單位: 毫秒)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Time_ms = 500;
ret = ECAT_McSetGroupAccTimeEx(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set group acceleration time:%d\n", ret);
}
else
{
    printf("Set group acceleration time successfully!\n");
}
```

---

## 7.6.9. ECAT\_McGetGroupAccTime

### 說明:

取得群組執行群組運動時使用的加速度時間設定。使用線性(梯形)加減速時會使用此函式所得的時間做加速或是減速；如果是 S 型加減速類型，則真正使用時間會變成此值的兩倍。

### 格式:

```
int32_t ECAT_McGetGroupAccTime(uint16_t DeviceNo, uint16_t GroupNo, uint16_t
*Time_ms)
```

大

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Time_ms	uint16_t*	OUT	執行群組運動時使用的加速度時間 (單位: 毫秒)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Time_ms;
ret = ECAT_McGetGroupAccTime(DeviceNo, GroupNo, &Time_ms);
if(ret < 0)
{
    printf("Failed to get group acceleration time:%d\n", ret);
}
else
{
    printf("group[%u] Acceleration Time(ms):%f\n", GroupNo, Time_ms);
}
```

---

## 7.6.10. ECAT\_McSetGroupAccDecType\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupAccDecType"  
設定群組執行群組運動時使用的加速度類型。

### 格式:

int32\_t ECAT\_McSetGroupAccDecType\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t Type)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Type	uint16_t	IN	執行群組運動時使用的加速度類型 1:T-Curve (線性) (default) 2:S-Curve

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type = 1; //T-Curve
ret = ECAT_McSetGroupAccDecType_Ex(DeviceNo, GroupNo, Type);
if(ret < 0)
{
    printf("Failed to set group AccDecType:%d\n", ret);
}
else
{
    printf("Set group AccDecType successfully!\n");
}
```

---

## 7.6.11. ECAT\_McGetGroupAccDecType

### 說明:

取得群組執行群組運動時使用的加速度類型。

### 格式:

```
int32_t ECAT_McGetGroupAccDecType(uint16_t DeviceNo, uint16_t GroupNo, uint16_t *Type)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Type	uint16_t*	OUT	執行群組運動時使用的加速度類型 1: T-Curve (線性) 2: S-Curve

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type;
ret = ECAT_McGetGroupAccDecType(DeviceNo, GroupNo, &Type);
if(ret < 0)
{
    printf("Failed to get group AccDecType:%d\n",ret);
}
else
{
    printf("group[%u] AccDecType:%f\n", GroupNo, Type);
}
```

---

## 7.6.12. ECAT\_McSetGroupBlendingPercent\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupBlendingPercent"

設定群組命令 BLENDING 模式的混合百分比，此命令立即執行。

### 格式:

int32\_t ECAT\_McSetGroupBlendingPercent\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t Value)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Value	uint16_t	IN	混合百分比 範圍:0~100 預設:100

### 回傳值:

0: 函式執行成功。

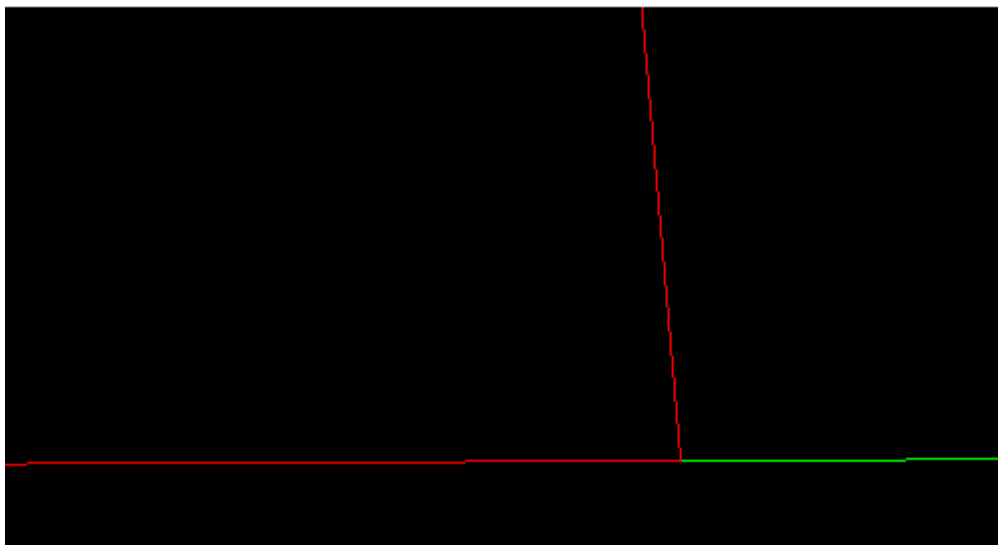
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

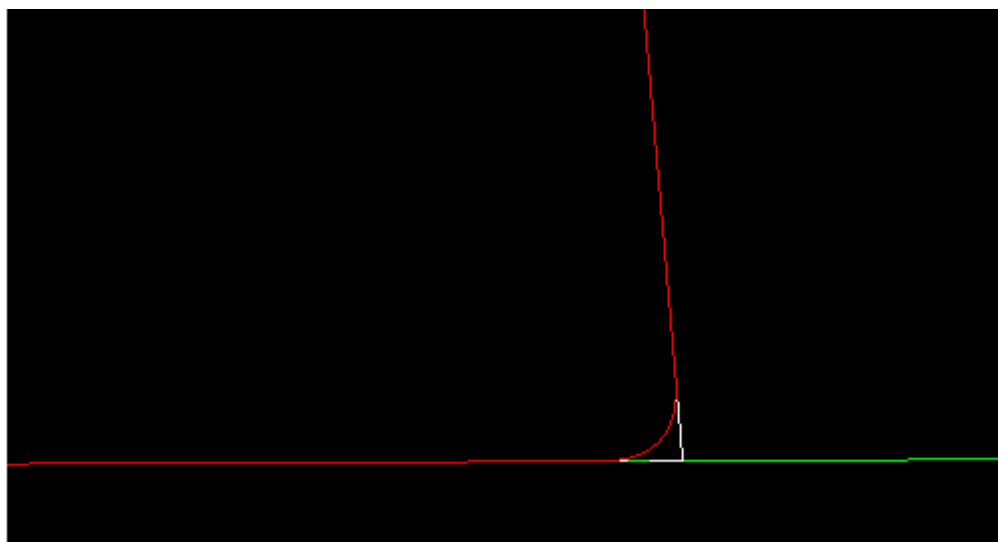
## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Value = 50;
ret = ECAT_McSetGroupBlendingPercent_Ex(DeviceNo, GroupNo, Value);
if(ret < 0)
{
    printf("Failed to set group blending percent:%d\n", ret);
}
else
{
    printf("Set group blending percent successfully!\n");
}
```

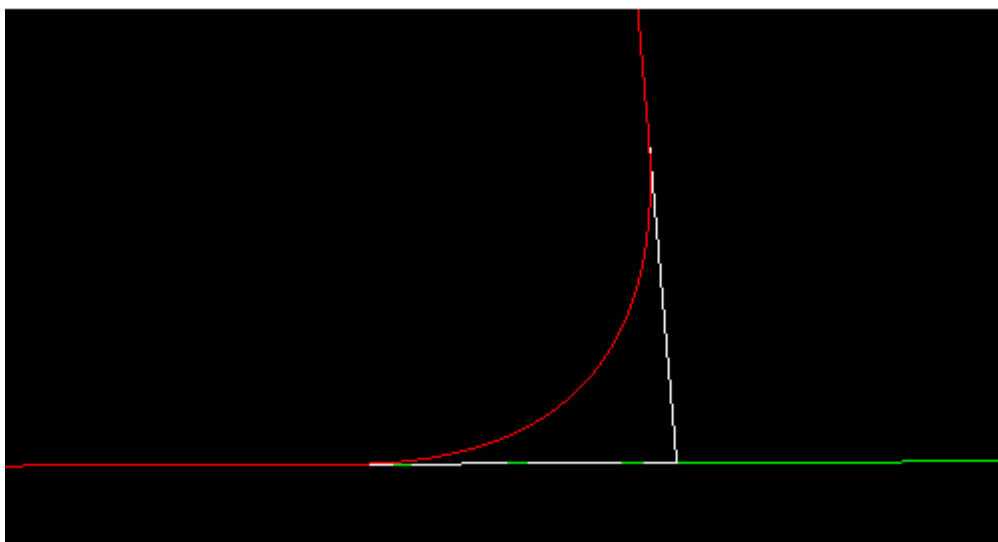
0%



50%



100%(default)





### 7.6.13. ECAT\_McSetGroupBlendingPercentEx\_Ex

#### 說明:

取代舊的函式"ECAT\_McSetGroupBlendingPercentEx"

設定群組命令 BLENDING 模式的混合百分比，此命令將排進命令緩衝區。

#### 格式:

```
int32_t ECAT_McSetGroupBlendingPercentEx_Ex(uint16_t DeviceNo, uint16_t
GroupNo, uint16_t Value)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Value	uint16_t	IN	混合百分比 範圍:0~100 預設:100

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t Time_ms = 999;
double Pos[MC_AXIS_NO_MAX]={ 0};
double Vel = 5;

// Add Axis
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
    printf("Failed to add axis to group:%d\n", ret);
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
    printf("Failed to add axis to group:%d\n", ret);
// Set Acctime
ret = ECAT_McSetGroupAccTime(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
    printf("Failed to set group acceleration time:%d\n", ret);
// Set blending mode
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, MS_GRP_CM_BLENDED);
if(ret < 0)
    printf("Failed to set group command mode:%d\n", ret);

// Start
ret = ECAT_McSetGroupBlendingPercentEx_Ex(DeviceNo,GrpNo, 80);
if(ret < 0)    printf("Failed to set group blending percent:%d\n", ret);
Pos[0] = 2;
Pos[1] = 0;
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);
if(ret < 0)    printf("Failed to group move abs%d\n", ret);

```

---

```
ret = ECAT_McSetGroupBlendingPercentEx_Ex(DeviceNo,GrpNo, 60);  
if(ret < 0)    printf("Failed to set group blending percent:%d\n", ret);  
Pos[0] = 2;  
Pos[1] = 2;  
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);  
if(ret < 0)    printf("Failed to group move abs%d\n", ret);
```

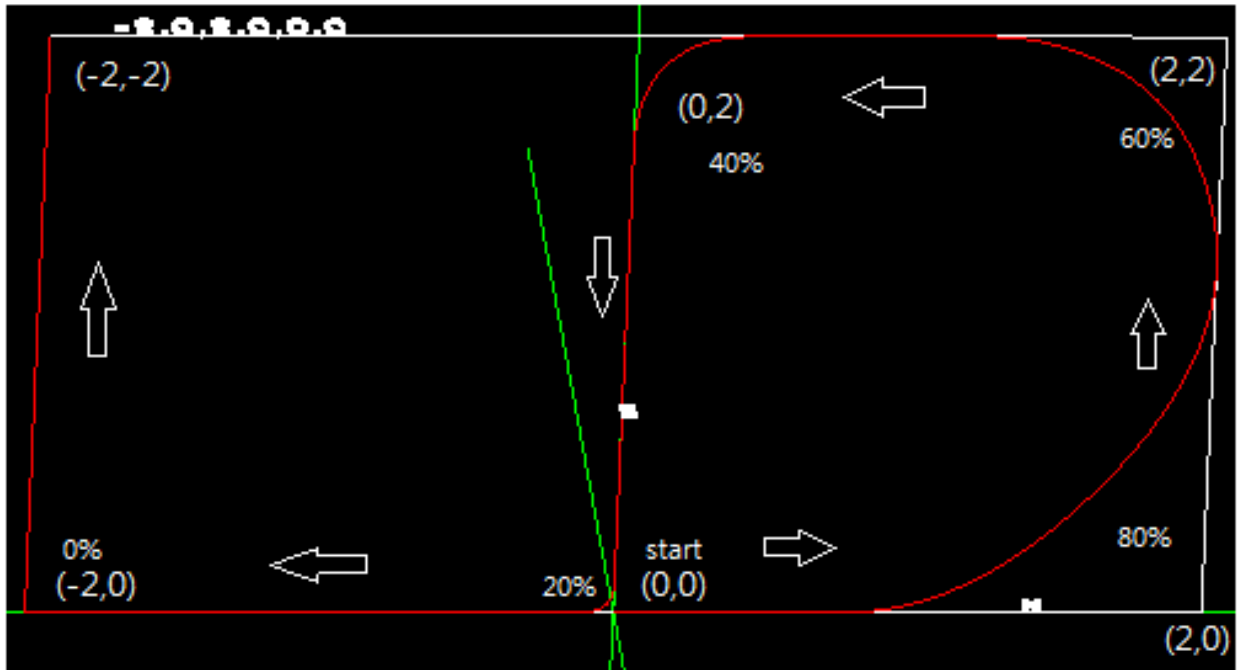
```
ret = ECAT_McSetGroupBlendingPercentEx_Ex(DeviceNo,GrpNo, 40);  
if(ret < 0)    printf("Failed to set group blending percent:%d\n", ret);  
Pos[0] = 0;  
Pos[1] = 2;  
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);  
if(ret < 0)    printf("Failed to group move abs%d\n", ret);
```

```
ret = ECAT_McSetGroupBlendingPercentEx_Ex(DeviceNo,GrpNo, 20);  
if(ret < 0)    printf("Failed to set group blending percent:%d\n", ret);  
Pos[0] = 0;  
Pos[1] = 0;  
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);  
if(ret < 0)    printf("Failed to group move abs%d\n", ret);
```

```
ret = ECAT_McSetGroupBlendingPercentEx_Ex(DeviceNo,GrpNo, 0);  
if(ret < 0)    printf("Failed to set group blending percent:%d\n", ret);  
Pos[0] = -2;  
Pos[1] = 0;  
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);  
if(ret < 0)    printf("Failed to group move abs%d\n", ret);
```

```
Pos[0] = -2;  
Pos[1] = 2;  
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);  
if(ret < 0)    printf("Failed to group move abs%d\n", ret);
```

---



## 7.6.14. ECAT\_McSetGroupPvtDecEnable\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupPvtDecEnable"  
設定群組 PVT 運動執行至最後一個命令後是否減速。

### 格式:

int32\_t ECAT\_McSetGroupPvtDecEnable\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t Enable)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Enable	uint16_t	IN	0:不減速(default) 1:減速

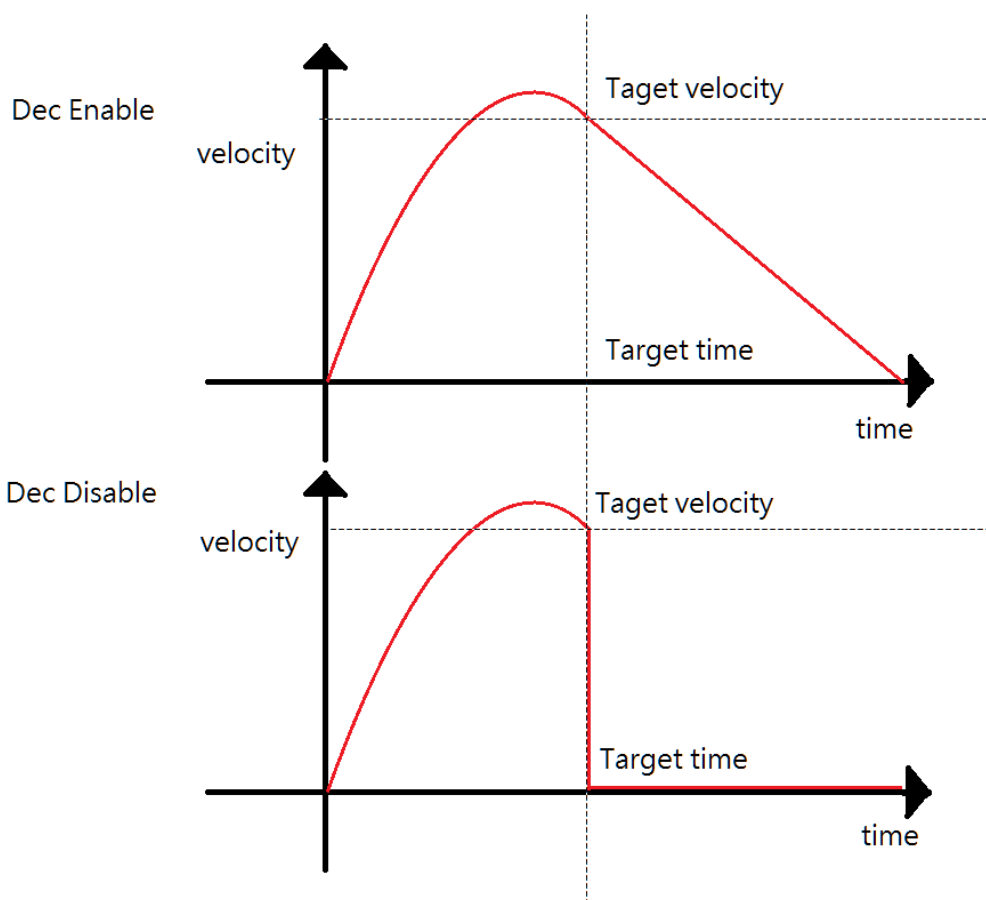
### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint16_t GroupNo = 0;  
uint16_t Enable= 1;  
ret = ECAT_McSetGroupPvtDecEnable_Ex(DeviceNo, GroupNo, Enable);  
if(ret < 0)  
{  
    printf("Failed to set group PvtDecEnable:%d\n", ret);  
}  
else  
{  
    printf("Set group PvtDecEnable successfully!\n");  
}
```



## 7.6.15. ECAT\_McGetGroupPvtDecEnable

### 說明:

取得群組 PVT 運動執行至最後一個命令後是否減速。

### 格式:

```
int32_t ECAT_McGetGroupPvtDecEnable(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t *Enable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Enable	uint16_t*	OUT	0:不減速 1:減速

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Enable;
ret = ECAT_McGetGroupPvtDecEnable(DeviceNo, GroupNo, &Enable);
if(ret < 0)
{
    printf("Failed to get group PvtDecEnable:%d\n",ret);
}
else
{
    printf("group[%u] PvtDecEnable:%f\n", GroupNo, Enable);
}
```

---



## 7.6.16. ECAT\_McSetGroupCoordinate\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupCoordinate"

設定群組運動控制是否執行座標轉換。

當進行座標轉換時，無法使用表 7.11 之函式，若要使用，先將座標轉換模式設為 MC\_DEFAULT\_INTERPOLATION 方可使用。

表 7.11

名稱	類型
ECAT_McSetAxisServoOn	單軸運動控制設定
ECAT_McSetAxisPPU	單軸運動控制設定
ECAT_McAxisHome	單軸自動原點復歸
ECAT_McAxisHomeEx	單軸自動原點復歸
7.5 單軸運動控制 中的所有函式	單軸運動控制
ECAT_McAddAxisToGroup	群組運動控制設定
ECAT_McRemoveAxisFromGroup	群組運動控制設定
ECAT_McUngroupAllAxes	群組運動控制設定
ECAT_McGroupMoveShaker	群組運動控制

### 格式:

```
int32_t ECAT_McSetGroupCoordinate_Ex(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t Type)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Type	uint16_t	IN	表 7.12: 群組座標轉換模式編號 預設: MC_DEFAULT_INTERPOLATION

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.12: 群組座標轉換模式編號

標頭檔定義名稱	數值	說明
MC_DEFAULT_INTERPOLATION	0	不進行座標轉換
MC_POLAR_INTERPOLATION	1	<p>命令為直角座標，輸出為極座標</p> <p>注意:</p> <ul style="list-style-type: none"> <li>➤ 在此模式時，機構應為一直線軸與一旋轉軸</li> <li>➤ 應避免直線軸長度為小於零</li> <li>➤ 需設定旋轉軸的單位為 rad，也就是說，需要將 PPU 設為旋轉 1rad 所需的 pulse 數，而直線軸的 PPU 可由使用者自行設定。</li> <li>➤ 在此模式下，命令位置為直角座標模式，單位與直線軸相同。</li> <li>➤ 直線軸需第一個加入群組當中，旋轉軸需第二個加入群組當中，請看以下範例。</li> </ul>

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type;
uint16_t AxisNo;
double PPU;

//先加入直線軸
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
//第二加入旋轉軸

```

---

```
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCoordinate_Ex(DeviceNo, GroupNo, Type);
if(ret < 0)
{
    printf("Failed to set group Coordinate:%d\n",ret);
}

```

---

## 7.6.17. ECAT\_McSetGroupCoordinateLimit\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupCoordinateLimit"  
設定群組座標轉換執行時的位置軟體極限。

### 格式:

```
int32_t ECAT_McSetGroupCoordinateLimit_Ex(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t Type, uint16_t Enable, double MIN_Value, double MAX_Value);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Type	uint16_t	IN	表 7.12: 群組座標轉換模式編號
Enable	uint16_t	IN	啟用/禁用 軟體極限 預設值: 禁用 0: 禁用 1: 啟用
MIN_Value	double	IN	軟體極限最小值
MAX_Value	double	IN	軟體極限最大值

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type = MC_POLAR_INTERPOLATION;
uint16_t Enable = 1;
double Min = 0;
double Max = 10;
ret = ECAT_McSetGroupCoordinateLimit_Ex(DeviceNo, GroupNo, Type, Enable, Min, Max);
if(ret < 0)
{
    printf("Failed to set group coordinate limit:%d\n",ret);
}
```

---

## 7.6.18. ECAT\_McGetGroupCoordinateLimit\_Ex

### 說明:

取代舊的函式"ECAT\_McGetGroupCoordinateLimit"

取得群組座標轉換執行時的位置軟體極限。

### 格式:

```
int32_t ECAT_McGetGroupCoordinateLimit_Ex(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t Type, uint16_t *Enable, double *MIN_Value, double *MAX_Value);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Type	uint16_t	IN	表 7.12: 群組座標轉換模式編號
Enable	uint16_t*	OUT	啟用/禁用 軟體極限 0: 禁用 1: 啟用
MIN_Value	double*	OUT	軟體極限最小值
MAX_Value	double*	OUT	軟體極限最大值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type = MC_POLAR_INTERPOLATION;
uint16_t Enable;
double Min;
double Max;
ret = ECAT_McGetGroupCoordinateLimit_Ex(DeviceNo, GroupNo, Type, &Enable, &Min, &Max);
if(ret < 0)
{
    printf("Failed to get group coordinate limit:%d\n",ret);
}
else
{
    printf("Enable: %d, min: %f, max: %f\n", Enable, Min, Max);
}
```

---



## 7.7. 群組運動狀態

### 7.7.1. ECAT\_McGetGroupState

**說明:**

取得指定群組當前群組狀態。

**格式:**

```
int32_t ECAT_McGetGroupState(uint16_t DeviceNo, uint16_t GroupNo, uint32_t *State)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
State	uint32_t*	OUT	取得指定群組當前狀態編號(如表 7.13 所示)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.13: 群組狀態編號

標頭檔定義名稱	數值	說明
MC_GS_DISABLED	0	群組尚未啟用
MC_GS_STANDBY	1	群組啟用且為停止狀態，準備接收新的運動命令
MC_GS_ERRORSTOP	2	群組目前出現錯誤且為停止狀態
MC_GS_STOPPING	3	群組目前在停止運動中
MC_GS_HOMING	4	保留無使用
MC_GS_MOVING	5	群組目前群組運動中

## 使用範例:

[C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
char buf[512];
uint32_t State;
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(ret < 0)
{
    printf("Failed to get group state:%d\n", ret);
}
else
{
    switch(State)
    {
        case MC_GS_DISABLED:
            sprintf(buf,"Disabled");
            break;
        case MC_GS_STANDBY:
            sprintf(buf,"Standby");
            break;
        case MC_GS_ERRORSTOP:
            sprintf(buf,"ErrorStop");
            break;
    }
}

```

---

```
case MC_GS_STOPPING:
    sprintf(buf,"Stopping");
    break;
case MC_GS_HOMING:
    sprintf(buf,"Homing");
    break;
case MC_GS_MOVING:
    sprintf(buf,"Moving");
    break;
default:
    sprintf(buf,"Invalid");
}
printf("Group State:%s\n", buf);
}
```

---

## 7.7.2. ECAT\_McGetGroupPauseState

### 說明:

取得指定群組當前群組 Pause 狀態。

### 格式:

int32\_t ECAT\_McGetGroupPauseState(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t\* State)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
State	uint32_t*	OUT	0:Pausing 1:完成 Pause , 運動已停止

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
char buf[512];
uint32_t State;
ret = ECAT_McGetGroupPauseState(DeviceNo, GroupNo, &State);
if(ret < 0)
{
    printf("Failed to get group state:%d\n", ret);
}
else
{
    switch(State)
    {
        case 0:
            sprintf(buf,"Pausing");
            break;
        case 1:
            sprintf(buf,"Pause Done");
            break;
        default:
            sprintf(buf,"Invalid");
    }
    printf("Group Pause State:%s\n", buf);
}
```

### 7.7.3. ECAT\_McGetGroupCmdBuffer

#### 說明:

取得群組中在命令緩衝區等待被執行的命令數量。如果需要連續運動，使用此資訊可以決定是否要馬上新增命令到緩衝區。

#### 格式:

```
int32_t ECAT_McGetGroupCmdBuffer(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t *Buffer)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Buffer	uint16_t*	OUT	在命令緩衝區等待被執行的命令數量

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t BufferCnt;
ret = ECAT_McGetGroupCmdBuffer(DeviceNo, GroupNo, &BufferCnt);
if(ret < 0)
{
    printf("Failed to get group command buffer:%d\n", ret);
}
else
{
    printf("Group command buffer:%u\n", BufferCnt);
}
```

---

## 7.7.4. ECAT\_McSetGroupVelLimitStatus\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupVelLimitStatus"

設定指定群組當前群組速度限制狀態。注意，僅對 *ECAT\_McGroupMoveLineAbs\_PT* 以及 *ECAT\_McGroupMoveLineRel\_PT* 有效。

### 格式:

```
int32_t ECAT_McSetGroupVelLimitStatus_Ex(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t Status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Status	uint16_t	IN	指定群組當前速度限制狀態 0:未啟用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t State = 1;
ret = ECAT_McSetGroupVelLimitStatus_Ex(DeviceNo, GroupNo, State);
if(ret < 0)
{
    printf("Failed to Set group velocity limit status:%d\n", ret);
}
```

---

## 7.7.5. ECAT\_McGetGroupVelLimitStatus

### 說明:

取得指定群組當前群組速度限制狀態。

### 格式:

```
int32_t ECAT_McGetGroupVelLimitStatus(uint16_t DeviceNo, uint16_t GroupNo,
uint16_t *Status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Status	uint16_t*	OUT	指定群組當前速度限制狀態 0:未啟用 1:啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t State;
ret = ECAT_McGetGroupVelLimitStatus(DeviceNo, GroupNo, &State);
if(ret < 0)
{
    printf("Failed to get group group velocity limit status:%d\n", ret);
}
else
{
    printf("Group velocity limit status:%u\n", State);
}
```

---

## 7.7.6. ECAT\_McSetGroupVelLimitValue\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupVelLimitValue"

設定指定群組當前群組每一軸的最大速度限制。

注意, 僅 *ECAT\_McGroupMoveLineAbs\_PT* 以及 *ECAT\_McGroupMoveLineRel\_PT* 有效, 此函式會以加長時間的方式來達成每一軸最大速度限制的要求。

### 格式:

```
int32_t ECAT_McSetGroupVelLimitValue_Ex(uint16_t DeviceNo, uint16_t GroupNo,
double Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Value	double	IN	指定群組當前每一軸的最大速度限制

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
double Value = 100;
ret = ECAT_McSetGroupVelLimitValue_Ex(DeviceNo, GroupNo, Value);
if(ret < 0)
{
    printf("Failed to Set group velocity limit value:%d\n", ret);
}
```

---

### 7.7.7. ECAT\_McGetGroupVelLimitValue

#### 說明:

取得指定群組當前群組每一軸的最大速度限制。

注意, 僅對 *ECAT\_McGroupMoveLineAbs\_PT* 以及 *ECAT\_McGroupMoveLineRel\_PT* 有效, 此函式會以加長時間的方式來達成每一軸最大速度限制的要求。

#### 格式:

int32\_t ECAT\_McGetGroupVelLimitValue(uint16\_t DeviceNo, uint16\_t GroupNo, double \*Value)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Value	double*	OUT	指定群組當前每一軸的最大速度限制

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
double Value;
ret = ECAT_McGetGroupVelLimitValue(DeviceNo, GroupNo, &Value);
if(ret < 0)
{
    printf("Failed to get group group velocity limit value:%d\n", ret);
}
else
{
    printf("Group velocity limit value:%f\n", Value);
}
```

---

## 7.8. 群組運動控制

### 7.8.1. ECAT\_McGroupMoveLineAbs\_Ex

**說明:**

取代舊的函式"ECAT\_McGroupMoveLineAbs"  
指定位置和速度，執行指定群組的絕對位置線性補間移動。

**格式:**

int32\_t ECAT\_McGroupMoveLineAbs\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Pos[], double Vel)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Vel	double	IN	移動速度(單位: user unit/s)

**回傳值:**

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
```

---

```
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupVel = 5;
ret = ECAT_McGroupMoveLineAbs_Ex(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupVel = 10;
ret = ECAT_McGroupMoveLineAbs_Ex(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

## 7.8.2. ECAT\_McGroupMoveLineRel\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveLineRel"

指定位置和速度，執行指定群組的相對距離線性補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveLineRel_Ex(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

```

---

```
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupVel = 5;
ret = ECAT_McGroupMoveLineRel_Ex(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupVel = 10;
ret = ECAT_McGroupMoveLineRel_Ex(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

### 7.8.3. ECAT\_McGroupMoveLineAbs\_PT\_Ex

#### 說明:

取代舊的函式"ECAT\_McGroupMoveLineAbs\_PT"  
指定位置和時間，執行指定群組的絕對位置線性補間移動。

#### 格式:

int32\_t ECAT\_McGroupMoveLineAbs\_PT\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo,  
double Pos[], double Time)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位: second)

#### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

```

---

```
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupTime = 5;
ret = ECAT_McGroupMoveLineAbs_PT_Ex(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupTime = 10;
ret = ECAT_McGroupMoveLineAbs_PT_Ex(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---



## 7.8.4. ECAT\_McGroupMoveLineRel\_PT\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveLineRel\_PT"  
指定位置和時間，執行指定群組的相對距離線性補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveLineRel_PT_Ex(uint16_t DeviceNo, uint16_t GroupNo,
double Pos[], double Time)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位: second)

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

```

---

```
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupTime = 5;
ret = ECAT_McGroupMoveLineRel_PT_Ex(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupTime = 10;
ret = ECAT_McGroupMoveLineRel_PT_Ex(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

## 7.8.5. ECAT\_McGroupMoveLineAbs\_PVT

### 說明:

執行指定群組的絕對位置 PVT 運動控制。

### 格式:

```
int32_t ECAT_McGroupMoveLineAbs_PVT(uint16_t DeviceNo, uint16_t GroupNo,
double Pos[], double Vel[], double Time)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的絕對位置(單位:user unit)
Vel	double[]	IN	群組內各軸的速度(單位: user unit/s)
Time	double	IN	絕對移動時間，相對於第一點位置 (單位: second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double AxisPos[MC_AXIS_NO_MAX];
double AxisVel[MC_AXIS_NO_MAX];
double Time;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1

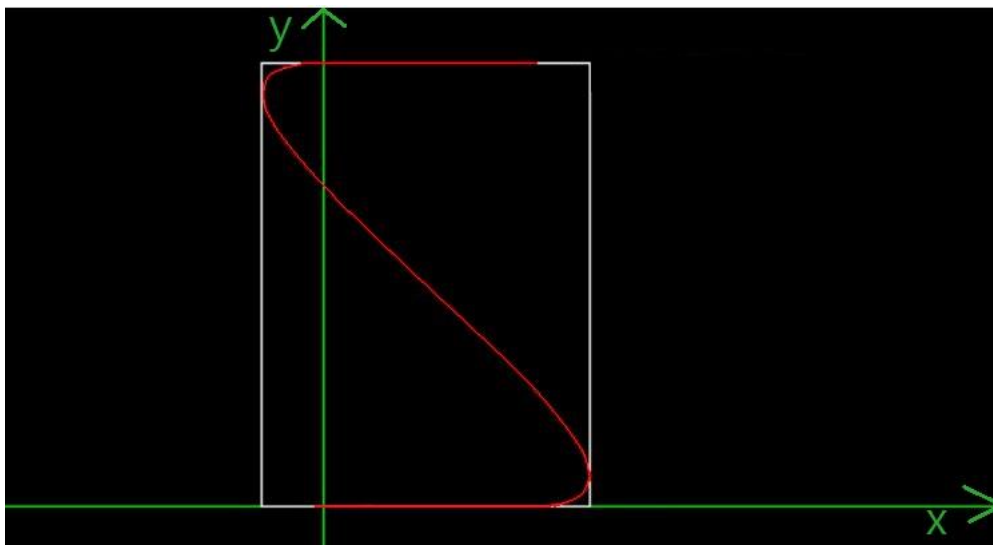
```

```
AxisPos[0] = 0.0;
AxisPos[1] = 0.0;
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 0;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 2
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 1.5;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 3
AxisPos[0] = 0.0;
AxisPos[1] = 10.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 3.0;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 4
AxisPos[0] = 5.0;
AxisPos[1] = 10.0;
```

```
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 4.5;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```



## 7.8.6. ECAT\_McGroupMoveLineRel\_PVT

### 說明:

執行指定群組的相對位置 PVT 運動控制。

### 格式:

int32\_t ECAT\_McGroupMoveLineRel\_PVT(uint16\_t DeviceNo, uint16\_t GroupNo, double Pos[], double Vel[], double Time)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的相對位置(單位:user unit)
Vel	double[]	IN	群組內各軸的速度(單位: user unit/s)
Time	double	IN	絕對移動時間，相對於第一點位置 (單位: second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double AxisPos[MC_AXIS_NO_MAX];
double AxisVel[MC_AXIS_NO_MAX];
double Time;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1

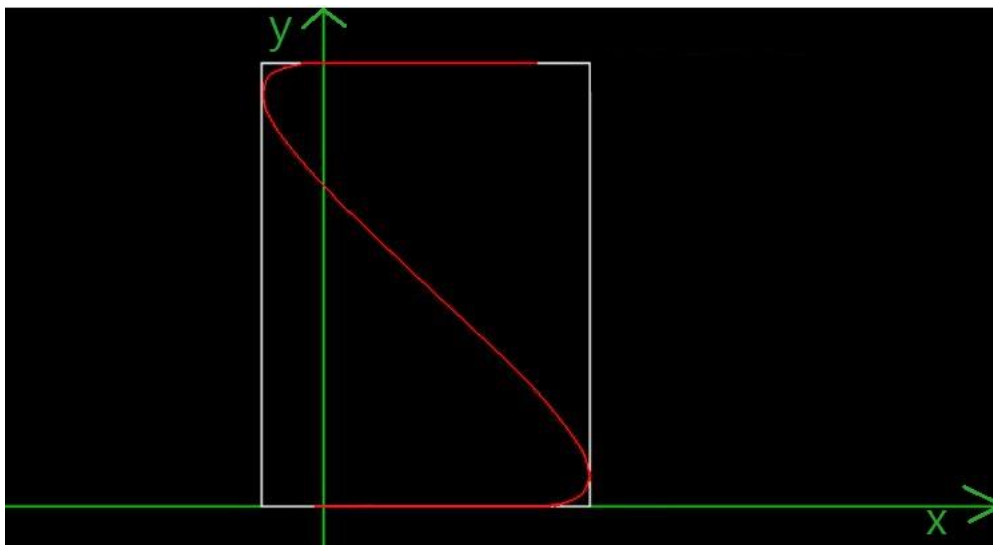
```

```
AxisPos[0] = 0.0;
AxisPos[1] = 0.0;
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 0;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 2
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 1.5;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 3
AxisPos[0] = -5.0;
AxisPos[1] = 10.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 3.0;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 4
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
```

```
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 4.5;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```



### 7.8.7. ECAT\_McGroupMoveLineAbs\_P2P

**說明:**

指定位置，執行指定群組內各軸絕對位置運動控制。

注:使用個軸最大速度執行(*ECAT\_McSetAxisMaxVelocity*)。

**格式:**

int32\_t ECAT\_McGroupMoveLineAbs\_P2P(uint16\_t DeviceNo, uint16\_t GroupNo, double Pos[])

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double MaxVelocity = 100;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}
```

---

```
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    ret = ECAT_McGroupMoveLineAbs_P2P(DeviceNo, GroupNo, GroupPos);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
}
```

---

### 7.8.8. ECAT\_McGroupMoveLineRel\_P2P

**說明:**

指定位置，執行指定群組內各軸相對位置運動控制。

注:使用個軸最大速度執行(*ECAT\_McSetAxisMaxVelocity*)。

**格式:**

```
int32_t ECAT_McGroupMoveLineRel_P2P(uint16_t DeviceNo, uint16_t GroupNo,  
double Pos[])
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double MaxVelocity = 100;

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}

```

```

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}

```



---

```
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
}
```

---

### 7.8.9. ECAT\_McGroupMoveLineAbs\_PTexT

#### 說明:

工廠內部測試用函式。

指定位置和時間，執行指定群組的絕對位置線性補間移動。

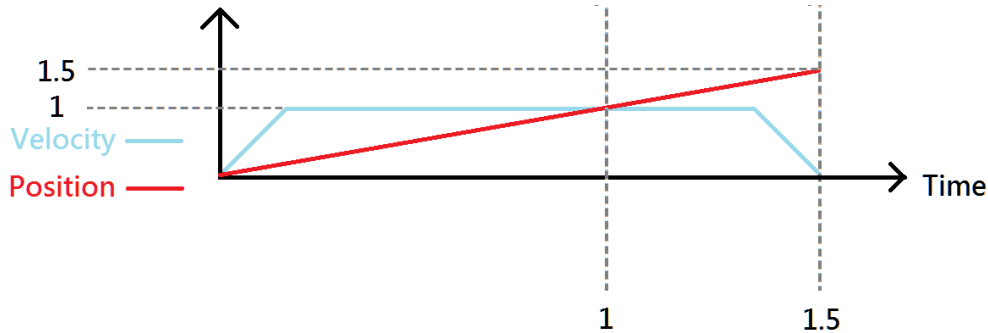
使用根據 Time(移動時間)所算出來的速度，再多運動 exTime(延長時間)秒

例:軸 0 目前位置為 0，目標位置為 1，移動時間為 1 秒，延長時間為 0.5 秒

根據計算，運動速度為目標位置 / 移動時間 =>  $1/1 = 1$ ，運動速度為 1

移動時間為 1 秒，延長時間為 0.5 秒，總共執行時間為 1.5 秒

總共運動距離為運動速度 \* 執行時間 =>  $1 * 1.5 = 1.5$ ，實際運動距離為 1.5



#### 格式:

```
int32_t ECAT_McGroupMoveLineAbs_PTexT(uint16_t DeviceNo, uint16_t GroupNo,
double Pos[], double Time, double exTime)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位:second)
exTime	double	IN	延長時間(單位:second)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;
double extandTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby

```

```
{  
    //Command 1  
    GroupPos[0] = 1;  
    GroupPos[1] = 0;  
    GroupTime = 1;  
    extendTime = 0.5;  
    ret = ECAT_McGroupMoveLineAbs_PText (DeviceNo, GroupNo, GroupPos, GroupTime);  
    if(ret < 0)  
    {  
        printf("Failed to add group move line command:%d\n", ret);  
    }  
  
    do  
    {  
        sleep(1);  
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);  
  
    }while(State == MC_GS_MOVING) //Moving  
  
    if(State == MC_GS_STANDBY) //Standby  
        printf("Group move line successfully!\n");  
    else if(State == MC_GS_ERRORSTOP) //ErrorStop  
    {  
        printf("Group error stop\n");  
    }  
}
```

### 7.8.10. ECAT\_McGroupMoveLineRel\_PTexT

**說明:**

工廠內部測試用函式。

指定位置和時間，執行指定群組的相對距離線性補間移動。

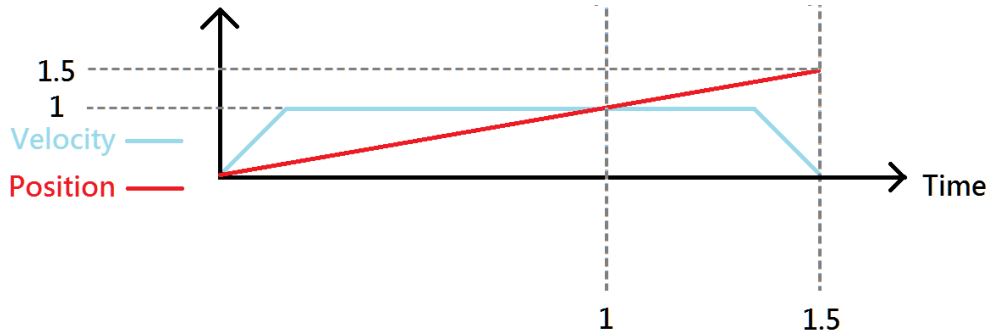
使用根據 Time(移動時間)所算出來的速度，再多運動 exTime(延長時間)秒

例:軸 0 目前位置為 0，目標位置為 1，移動時間為 1 秒，延長時間為 0.5 秒

根據計算，運動速度為目標位置 / 移動時間 => 1/1 = 1，運動速度為 1

移動時間為 1 秒，延長時間為 0.5 秒，總共執行時間為 1.5 秒

總共運動距離為運動速度 \* 執行時間 => 1\*1.5 = 1.5，實際運動距離為 1.5



**格式:**

```
int32_t ECAT_McGroupMoveLineRel_PTexT(uint16_t DeviceNo, uint16_t GroupNo,
double Pos[], double Time, double exTime)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位: second)
exTime	double	IN	延長時間(單位: second)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;
double extendTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby

```



```
{  
    //Command 1  
    GroupPos[0] = 1;  
    GroupPos[1] = 0;  
    GroupTime = 1;  
    extendTime = 0.5;  
  
    ret = ECAT_McGroupMoveLineRel_PText(DeviceNo, GroupNo, GroupPos, GroupTime, extendTime);  
    if(ret < 0)  
    {  
        printf("Failed to add group move line command:%d\n", ret);  
    }  
  
    do  
    {  
        sleep(1);  
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);  
  
    }while(State == MC_GS_MOVING) //Moving  
  
    if(State == MC_GS_STANDBY) //Standby  
        printf("Group move line successfully!\n");  
    else if(State == MC_GS_ERRORSTOP) //ErrorStop  
    {  
        printf("Group error stop\n");  
    }  
}
```

### 7.8.11. ECAT\_McGroupMoveLineAbs\_PPT

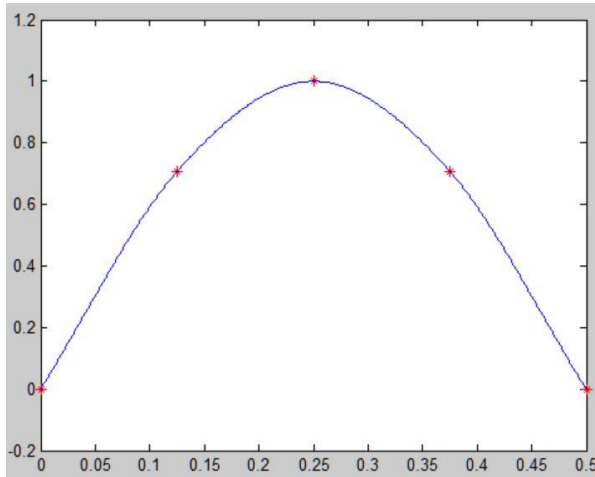
#### 說明:

指定位置和時間，執行指定群組的絕對位置曲線補間移動。

此命令會將移動軌跡平滑化。

此命令為三次平滑曲線，且會經過目標點。

注:僅供在 Blending 模式下使用。



#### 格式:

```
int32_t ECAT_McGroupMoveLineAbs_PPT(uint16_t DeviceNo, uint16_t GroupNo,
double Pos[], double Time)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位: second)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

```

---

```
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupTime = 5;
ret = ECAT_McGroupMoveLineAbs_PPT (DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupTime = 10;
ret = ECAT_McGroupMoveLineAbs_PPT(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

## 7.8.12. ECAT\_McGroupMoveLineRel\_PPT

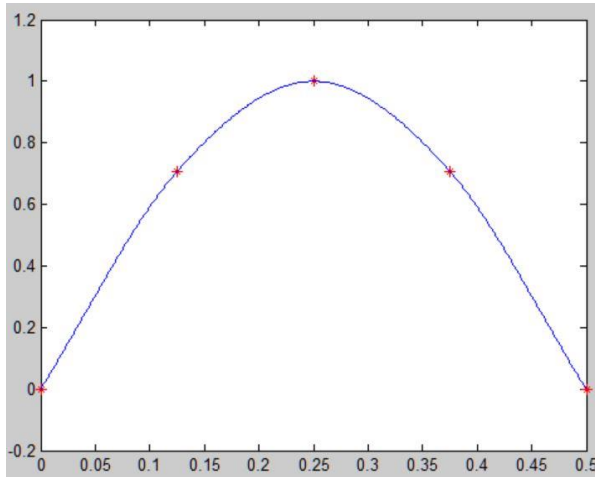
### 說明:

指定位置和時間，執行指定群組的相對距離曲線補間移動。

此命令會將移動軌跡平滑化。

此命令為三次平滑曲線，且會經過目標點。

注:僅供在 Blending 模式下使用。



### 格式:

```
int32_t ECAT_McGroupMoveLineRel_PPT(uint16_t DeviceNo, uint16_t GroupNo,
double Pos[], double Time)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Pos	double[]	IN	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位: second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

```



---

```
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupTime = 5;
ret = ECAT_McGroupMoveLineRel_PPT(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupTime = 10;
ret = ECAT_McGroupMoveLineRel_PPT(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

### 7.8.13. ECAT\_McGroupMoveCircularAbs\_CP\_Angle\_Ex

**說明:**

取代舊的函式"ECAT\_McGroupMoveCircularAbs\_CP\_Angle"

根據中心位置及旋轉角度，開始指定群組絕對位置 2D 圓弧補間移動。

**格式:**

```
int32_t ECAT_McGroupMoveCircularAbs_CP_Angle_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, double Angle, double AuxPos[])
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定旋轉角度(單位:degree)
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo,GroupNo, &State);

```

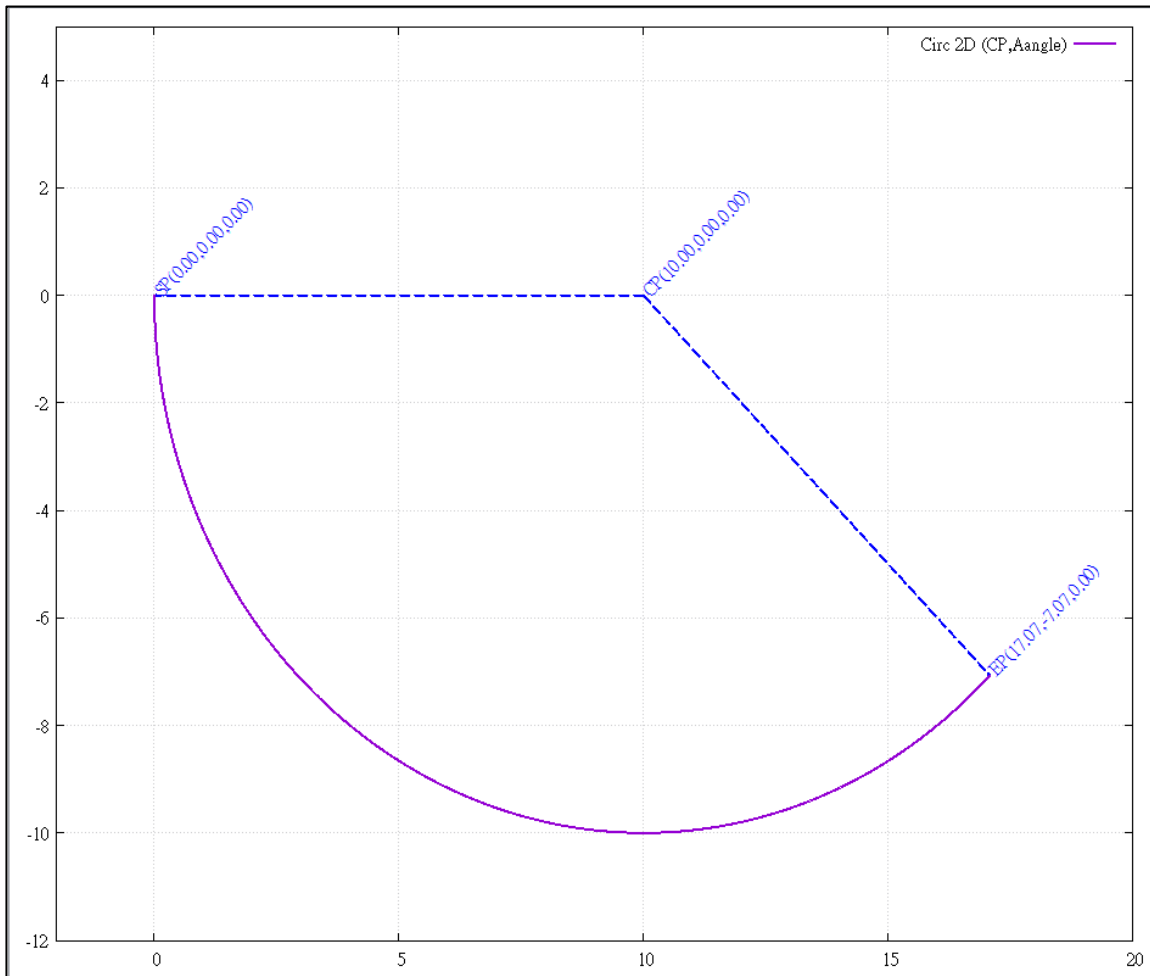
```
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos [0] = 10.0; //Center Position
    CircAuxPos [1] = 0.0; //Center Position
    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMoveCircularAbs_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel, CircAngle
        , CircAuxPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

使用範例的 2D 圓弧補間移動軌跡:



## 7.8.14. ECAT\_McGroupMoveCircularRel\_CP\_Angle\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveCircularRel\_CP\_Angle"

根據中心位置及旋轉角度，開始指定群組相對距離 2D 圓弧補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveCircularRel_CP_Angle_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, double Angle, double AuxPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定旋轉角度(單位:degree)
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

---

```
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 10.0; //Center Position
    CircAuxPos [1] = 0.0; //Center Position
    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMoveCircularRel_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel, CircAngle
        , CircAuxPos);

    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

---



## 7.8.15. ECAT\_McGroupMoveCircularAbs\_CP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveCircularAbs\_CP\_EP"

根據中心位置及終點位置，開始指定群組絕對位置 2D 圓弧補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveCircularAbs_CP_EP_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
EndPos	double[]	IN	絕對座標終點位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = 7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularAbs_CP_EP_Ex(DeviceNo, GroupNo, GroupVel, CircDir
        , CircAuxPos, CircEndPos);
    if(ret < 0)
        printf("Failed to add group move circular command:%d\n",ret);

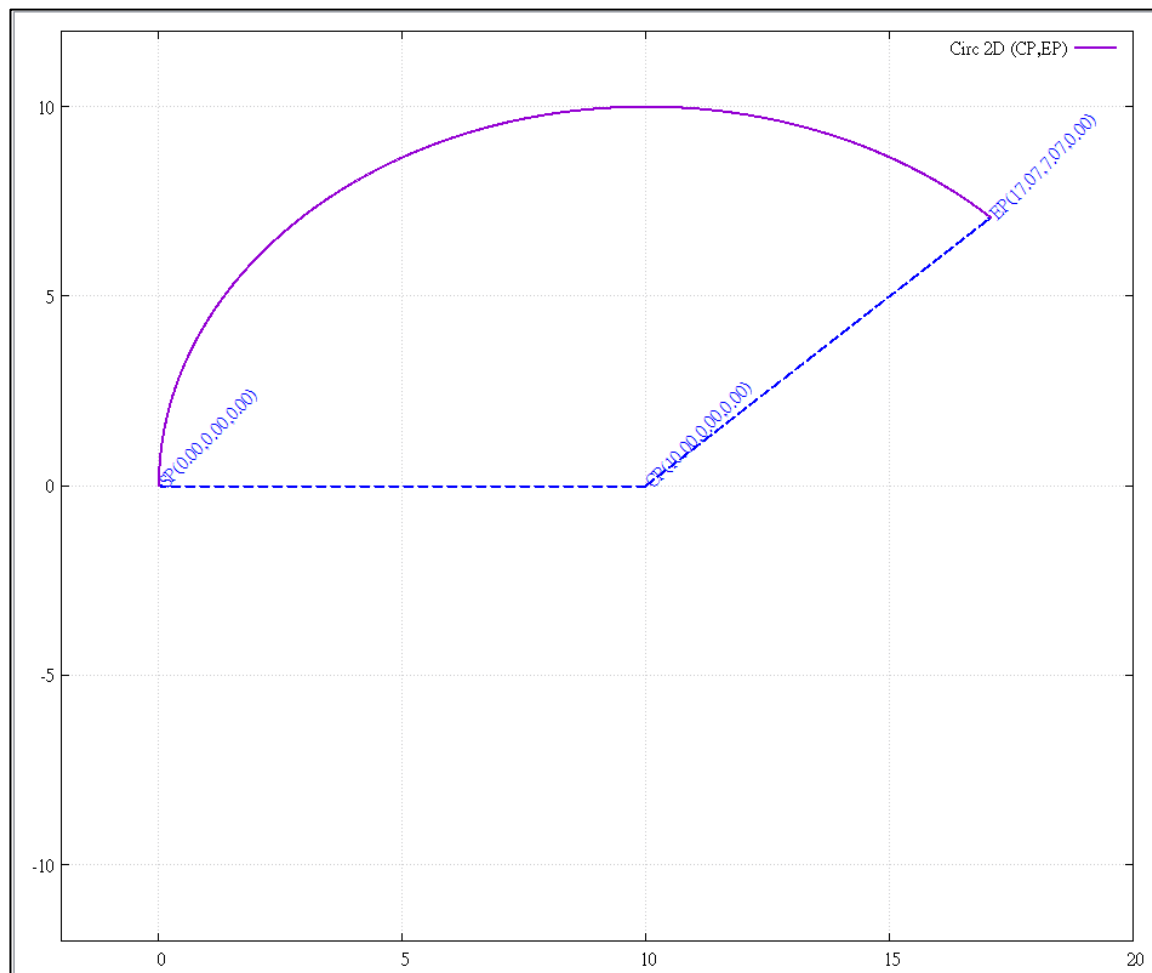
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

}

}

使用範例的 2D 圓弧補間移動軌跡:



## 7.8.16. ECAT\_McGroupMoveCircularRel\_CP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveCircularRel\_CP\_EP"

根據中心位置及終點位置，開始指定群組相對距離 2D 圓弧補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveCircularRel_CP_EP_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
EndPos	double[]	IN	相對距離終點位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

---

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = 7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularRel_CP_EP_Ex(DeviceNo, GroupNo, GroupVel, CircDir
        , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

---

## 7.8.17. ECAT\_McGroupMoveCircularAbs\_BP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveCircularAbs\_BP\_EP"

根據圓弧上一點位置及終點位置，開始指定群組絕對位置 2D 圓弧補間移動。注意：起始點與這兩點不能位於同一直線上。

### 格式:

```
int32_t ECAT_McGroupMoveCircularAbs_BP_EP_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	絕對座標圓弧上一點位置 (單位:user unit)
EndPos	double[]	IN	絕對座標終點位置 (單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos[0] = 17.071; //Border Position
    CircAuxPos[1] = 7.071; //Border Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = -7.071 // End Position

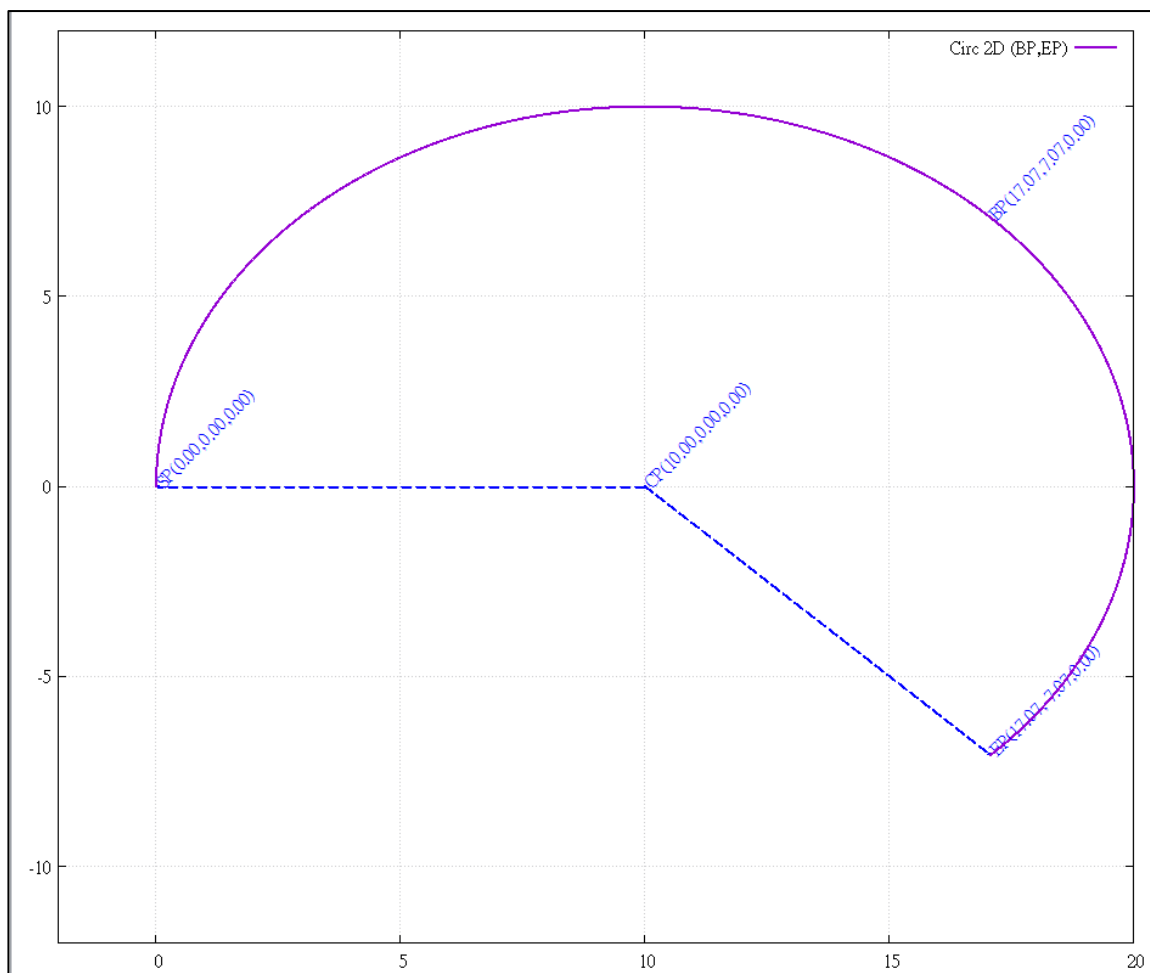
    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularAbs_BP_EP_Ex(DeviceNo, GroupNo, GroupVel, CircDir
        , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
```

```
printf("Group error stop\n");  
}  
}
```

使用範例的 2D 圓弧補間移動軌跡:



## 7.8.18. ECAT\_McGroupMoveCircularRel\_BP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveCircularRel\_BP\_EP"

根據圓弧上一點位置及終點位置，開始指定群組相對距離 2D 圓弧補間移動。注意：起始點與這兩點不能位於同一直線上。

### 格式:

```
int32_t ECAT_McGroupMoveCircularRel_BP_EP_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	相對距離圓弧上一點位置(單位:user unit)
EndPos	double[]	IN	相對距離終點位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos[0] = 17.071; //Border Position
    CircAuxPos[1] = 7.071; //Border Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = -7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularRel_BP_EP_Ex(DeviceNo, GroupNo, GroupVel, CircDir
        , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

## 7.8.19. ECAT\_McGroupMove3DCircularAbs\_CP\_Angle\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DCircularAbs\_CP\_Angle"  
根據中心位置及旋轉角度，開始指定群組絕對位置 3D 圓弧補間移動。

### 格式:

int32\_t ECAT\_McGroupMove3DCircularAbs\_CP\_Angle\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Vel, double Angle, double AuxPos[], double NV[])

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定旋轉角度(單位:degree)
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
NV	double[]	IN	法向量

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupPos[2] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

    NV [0] = -0.7071; //Normal Vector
    NV [1] = 0.7071; //Normal Vector
    NV [2] = 0.0;     //Normal Vector

    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularAbs_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel
        , CircAngle, CircAuxPos, NV);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

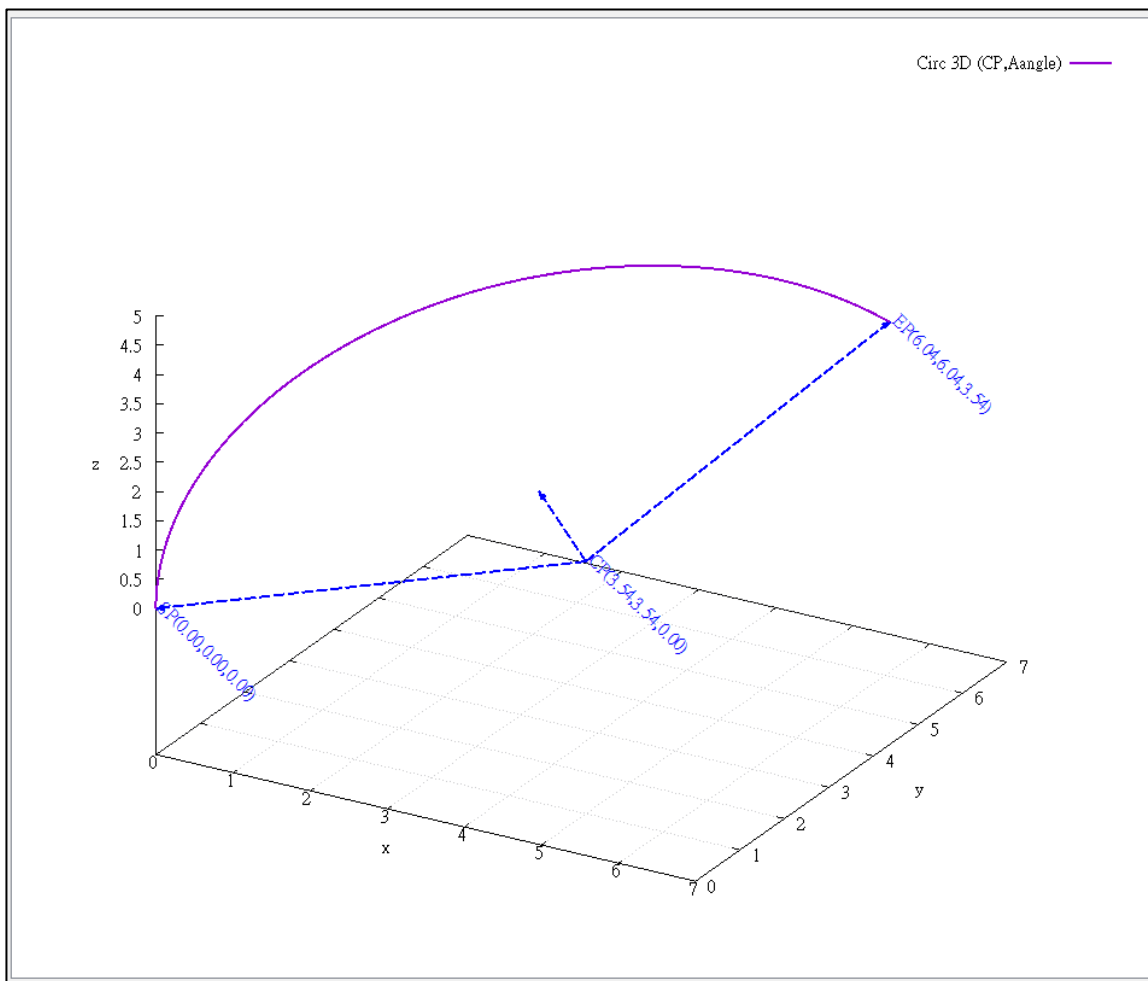
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
```

```
printf("Group move circular successfully!\n");  
else if(State == MC_GS_ERRORSTOP) //ErrorStop  
{  
    printf("Group error stop\n");  
}  
}
```

使用範例的 3D 圓弧補間移動軌跡:



## 7.8.20. ECAT\_McGroupMove3DCircularRel\_CP\_Angle\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DCircularRel\_CP\_Angle"

根據中心位置及旋轉角度，開始指定群組相對距離 3D 圓弧補間移動。

### 格式:

```
int32_t ECAT_McGroupMove3DCircularRel_CP_Angle_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, double Angle, double AuxPos[], double NV[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定旋轉角度(單位:degree)
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
NV	double[]	IN	法向量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

---

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

    NV [0] = -0.7071; //Normal Vector
    NV [1] = 0.7071; //Normal Vector
    NV [2] = 0.0;     //Normal Vector

    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularRel_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel
        , CircAngle, CircAuxPos, NV);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

---

## 7.8.21. ECAT\_McGroupMove3DCircularAbs\_CP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DCircularAbs\_CP\_EP"

根據中心位置及終點位置，開始指定群組絕對位置 3D 圓弧補間移動。

### 格式:

```
int32_t ECAT_McGroupMove3DCircularAbs_CP_EP_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
EndPos	double[]	IN	絕對座標終點位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
}

```

```
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupPos[2] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

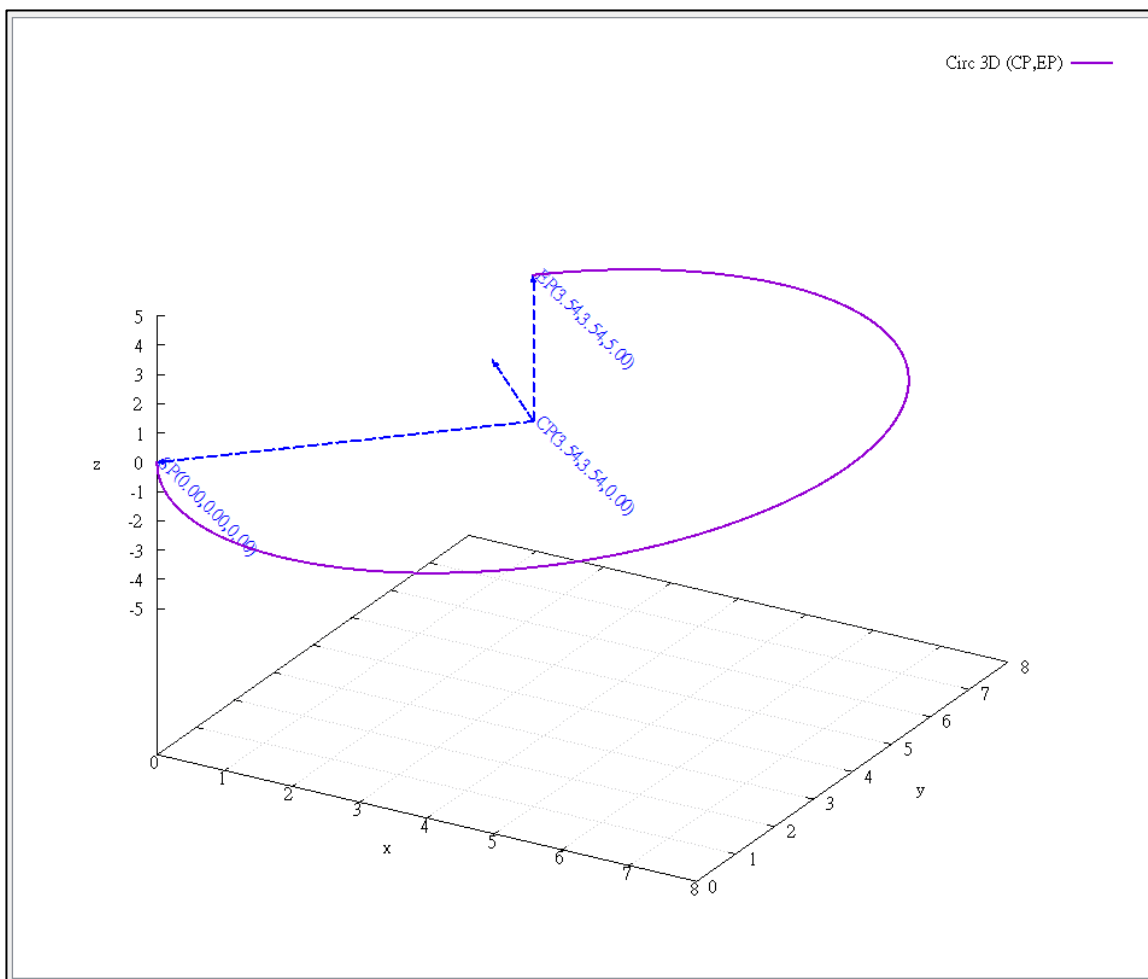
    CircEndPos[0] = 3.5355 // End Position
    CircEndPos[1] = 3.5355 // End Position
    CircEndPos[2] = 5.0    // End Position
    CircDir = 0; // CW
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularAbs_CP_EP_Ex(EcatDeviceID, GroupNo, GroupVel
        , CircDir, CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving
```



```
if(State == MC_GS_STANDBY) //Standby
    printf("Group move circular successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

使用範例的 3D 圓弧補間移動軌跡:



## 7.8.22. ECAT\_McGroupMove3DCircularRel\_CP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DCircularRel\_CP\_EP"

根據中心位置及終點位置，開始指定群組相對距離 3D 圓弧補間移動。

### 格式:

```
int32_t ECAT_McGroupMove3DCircularRel_CP_EP_Ex(uint16_t DeviceNo, uint16_t
GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
EndPos	double[]	IN	相對距離終點位置(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
}

```

```
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;      //Center Position

    CircEndPos[0] = 3.5355    // End Position
    CircEndPos[1] = 3.5355    // End Position
    CircEndPos[2] = 5.0      // End Position
    CircDir = 0; // CW
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularRel_CP_EP_Ex(EcatDeviceID, GroupNo, GroupVel
        , CircDir, CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

### 7.8.23. ECAT\_McGroupMove3DCircularAbs\_BP\_EP\_Ex

#### 說明:

取代舊的函式"ECAT\_McGroupMove3DCircularAbs\_BP\_EP"

根據圓弧上一點位置及終點位置，開始指定群組絕對位置 3D 圓弧補間移動。注意：起始點與這兩點不能位於同一直線上。

#### 格式:

int32\_t ECAT\_McGroupMove3DCircularAbs\_BP\_EP\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Vel, uint8\_t Dir, double AuxPos[], double EndPos[])

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	絕對座標圓弧上一點位置 (單位:user unit)
EndPos	double[]	IN	絕對座標終點位置 (單位:user unit)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
}

```

```
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupPos[2] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

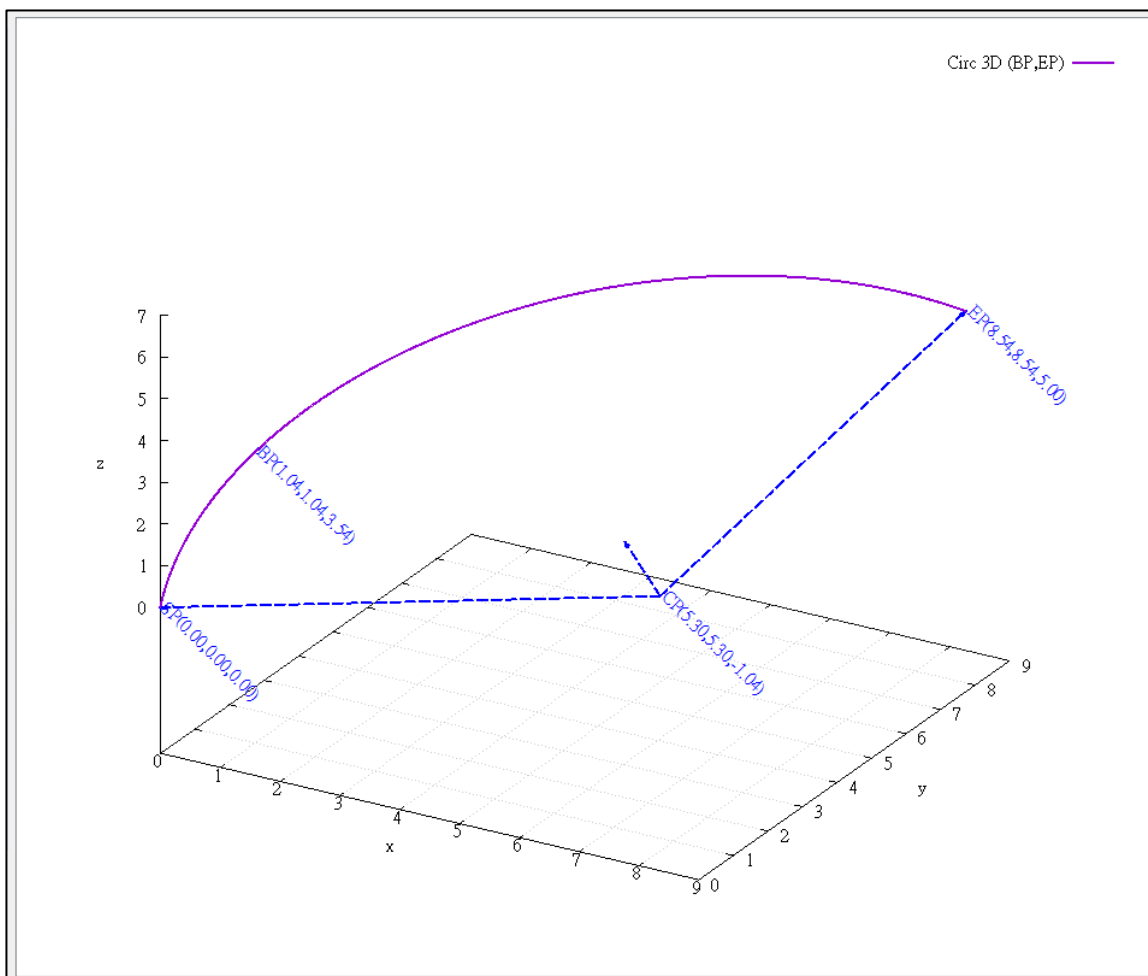
    //Command 2
    CircAuxPos [0] = 1.036; //Border Position
    CircAuxPos [1] = 1.036; //Border Position
    CircAuxPos [2] = 3.5355; //Border Position

    CircEndPos[0] = 8.53656 // End Position
    CircEndPos[1] = 8.53656 // End Position
    CircEndPos[2] = 5.0 // End Position
    CircDir = 1; // CCW
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularAbs_BP_EP_Ex(EcatDeviceID, GroupNo, GroupVel
        , CircDir, CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving
```

```
if(State == MC_GS_STANDBY) //Standby
    printf("Group move circular successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

使用範例的 3D 圓弧補間移動軌跡:





## 7.8.24. ECAT\_McGroupMove3DCircularRel\_BP\_EP\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DCircularRel\_BP\_EP"

根據圓弧上一點位置及終點位置，開始指定群組相對距離 3D 圓弧補間移動。注意：起始點與這兩點不能位於同一直線上。

### 格式:

```
int32_t ECAT_McGroupMove3DCircularRel_BP_EP_Ex(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度 (單位: user unit/s)
Dir	uint8_t	IN	旋轉方向 0: CW 1: CCW
AuxPos	double[]	IN	相對距離圓弧上一點位置 (單位:user unit)
EndPos	double[]	IN	相對距離終點位置 (單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

```

```

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
}

```

```
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 1.036; //Border Position
    CircAuxPos [1] = 1.036; //Border Position
    CircAuxPos [2] = 3.5355; //Border Position

    CircEndPos[0] = 8.53656 // End Position
    CircEndPos[1] = 8.53656 // End Position
    CircEndPos[2] = 5.0 // End Position
    CircDir = 1; // CCW
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularRel_BP_EP_Ex(EcatDeviceID, GroupNo, GroupVel
        , CircDir, CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

## 7.8.25. ECAT\_McGroupMoveHelicalAbs\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveHelicalAbs"

開始指定群組絕對距離螺旋補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveHelicalAbs_Ex(uint16_t DeviceNo, uint16_t GroupNo,
double Angle, double AuxPos[], double Pitch, double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Angle	double	IN	指定螺旋圈數(單位:degree)
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

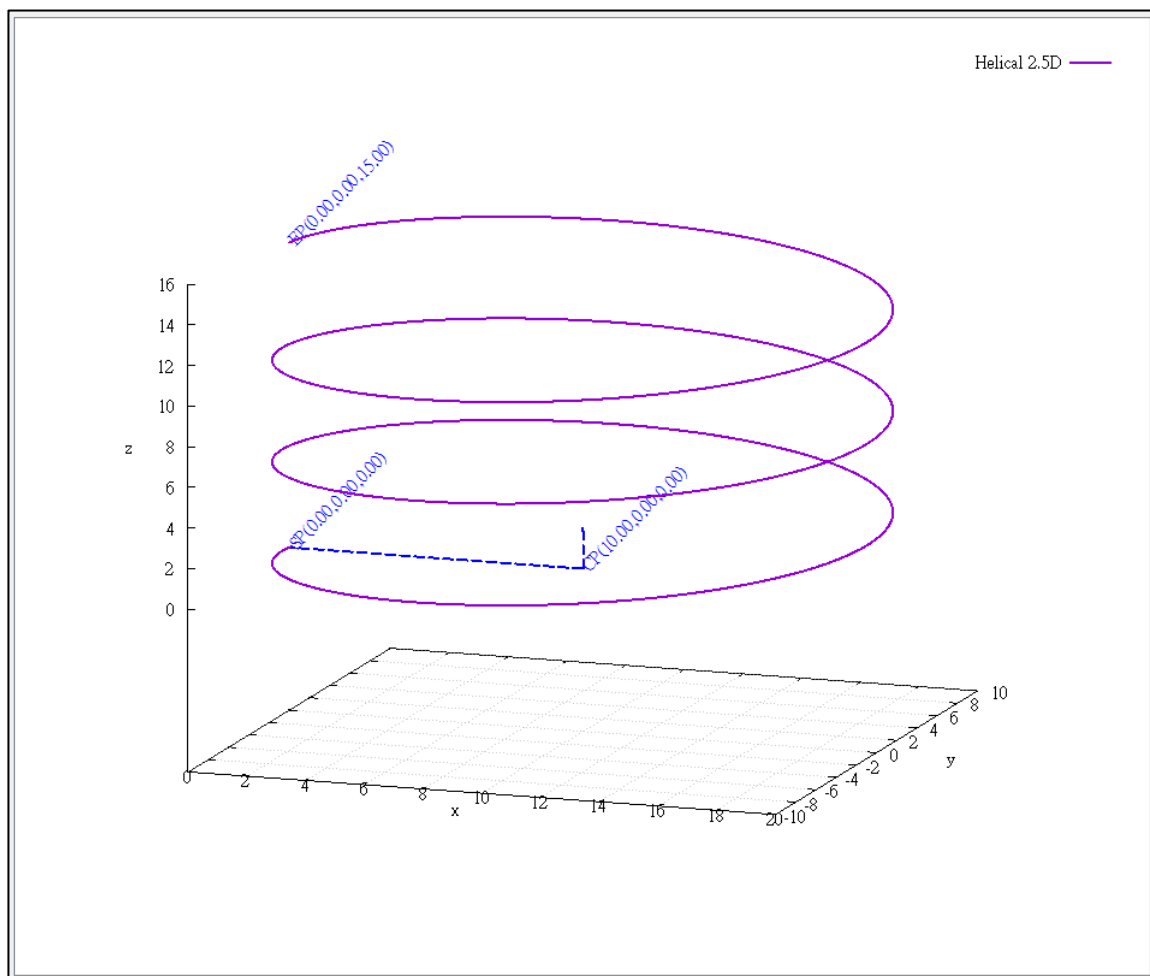
    //Command 2
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMoveHelicalAbs_Ex(DeviceNo,GroupNo
        , CircAngle, CircAuxPos, HelicalPitch, GroupVel);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

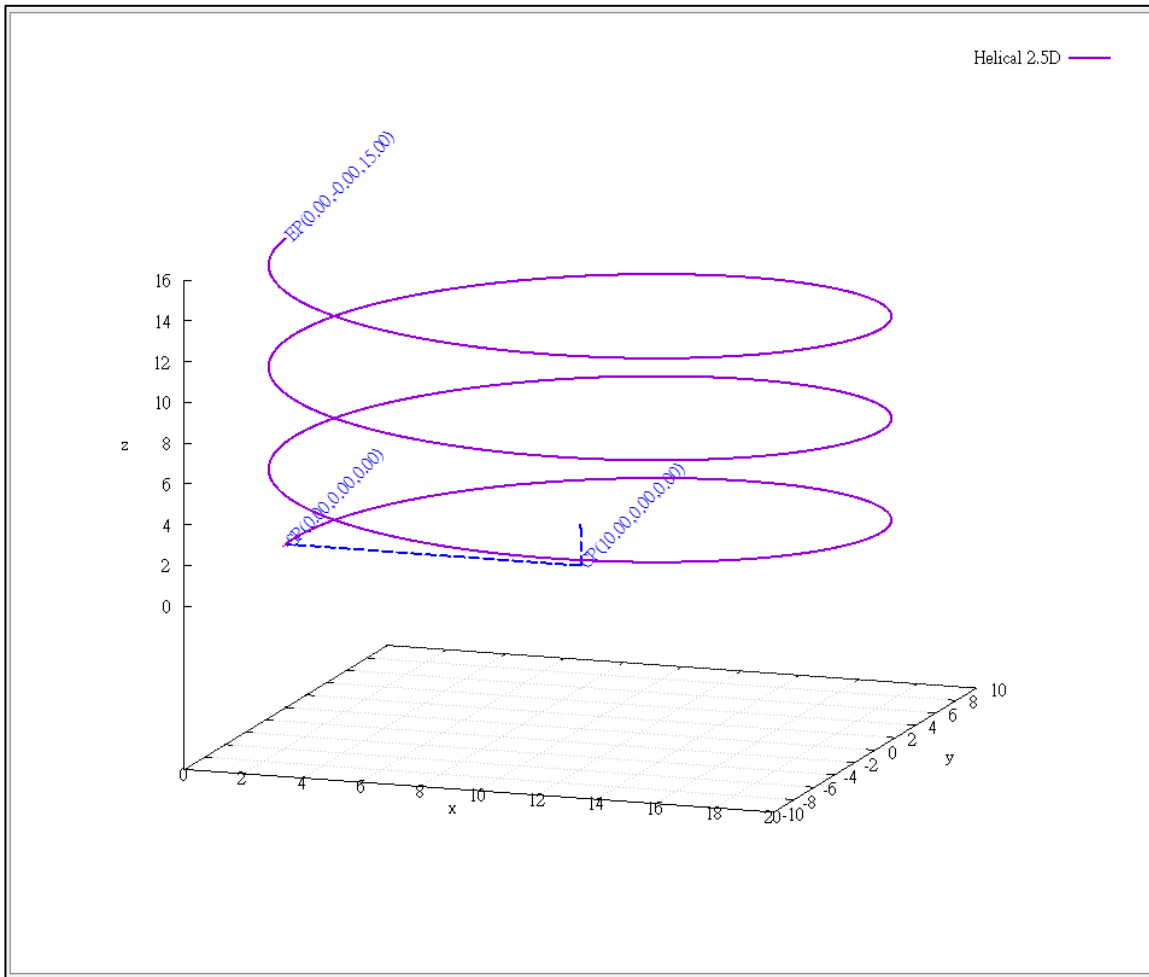
使用範例的螺旋補間移動軌跡(右螺旋):



**使用範例的螺旋補間移動軌跡(左螺旋):**

若旋轉角度參數設定成反方向，則移動軌跡更改為左螺旋

**CircAngle = -1080;**





## 7.8.26. ECAT\_McGroupMoveHelicalRel\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveHelicalRel"

開始指定群組相對距離螺旋補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveHelicalRel_Ex(uint16_t DeviceNo, uint16_t GroupNo,
double Angle, double AuxPos[], double Pitch, double Vel)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Angle	double	IN	指定螺旋圈數(單位:degree)
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

```

---

```
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMoveHelicalRel_Ex(DeviceNo,GroupNo
        , CircAngle, CircAuxPos, HelicalPitch, GroupVel);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

---

## 7.8.27. ECAT\_McGroupMove3DHelicalAbs\_CP\_Angle\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DHelicalAbs\_CP\_Angle"  
根據中心位置及旋轉角度，開始指定群組絕對位置 3D 螺旋補間移動。

### 格式:

int32\_t ECAT\_McGroupMove3DHelicalAbs\_CP\_Angle\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[])

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定螺旋圈數(單位:degree)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
NV	double[]	IN	法向量

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

    NV [0] = -0.7071; //Normal Vector
    NV [1] = 0.7071; //Normal Vector
    NV [2] = 0.0;     //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DHelicalAbs_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel
        , CircAngle, HelicalPitch, CircAuxPos, NV);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n",ret);

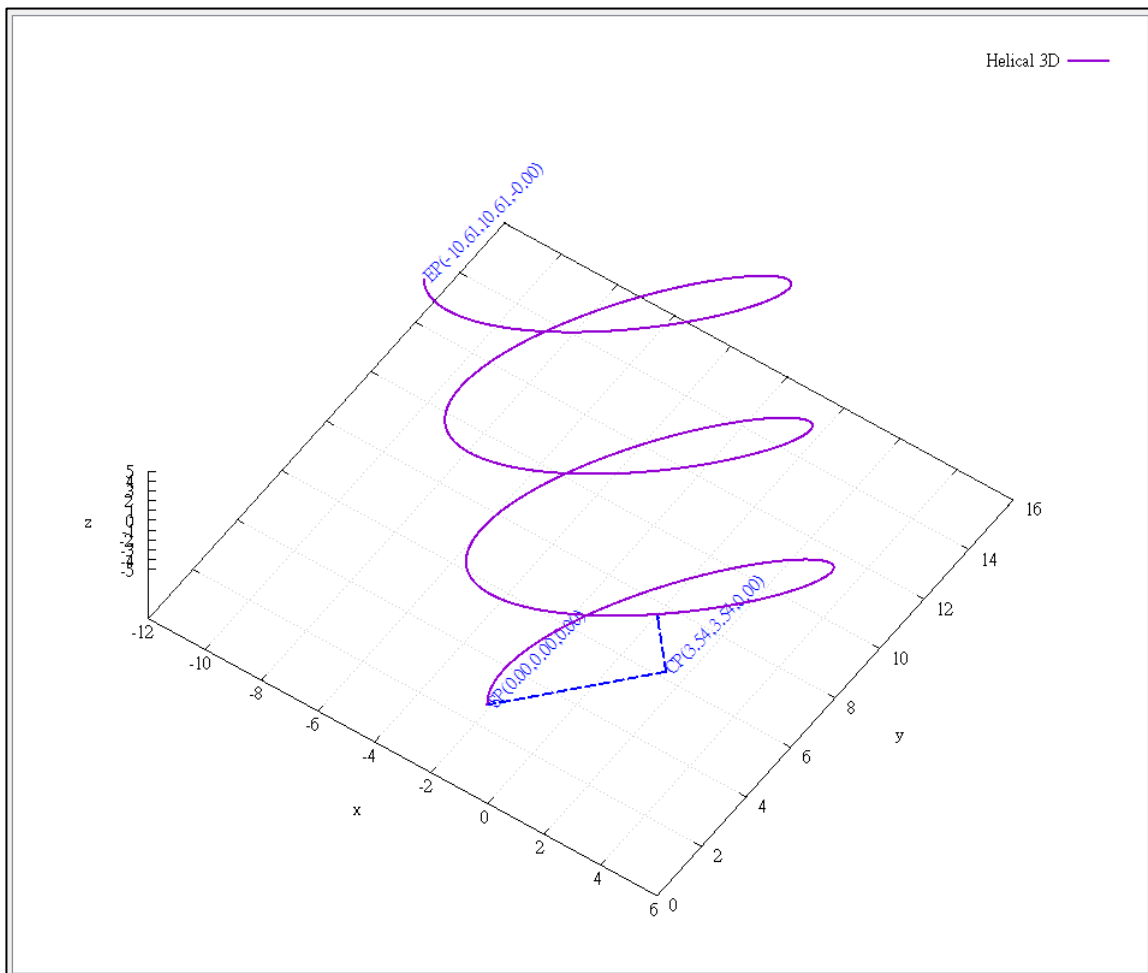
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
```

```
{  
    printf("Group error stop\n");  
}  
}
```

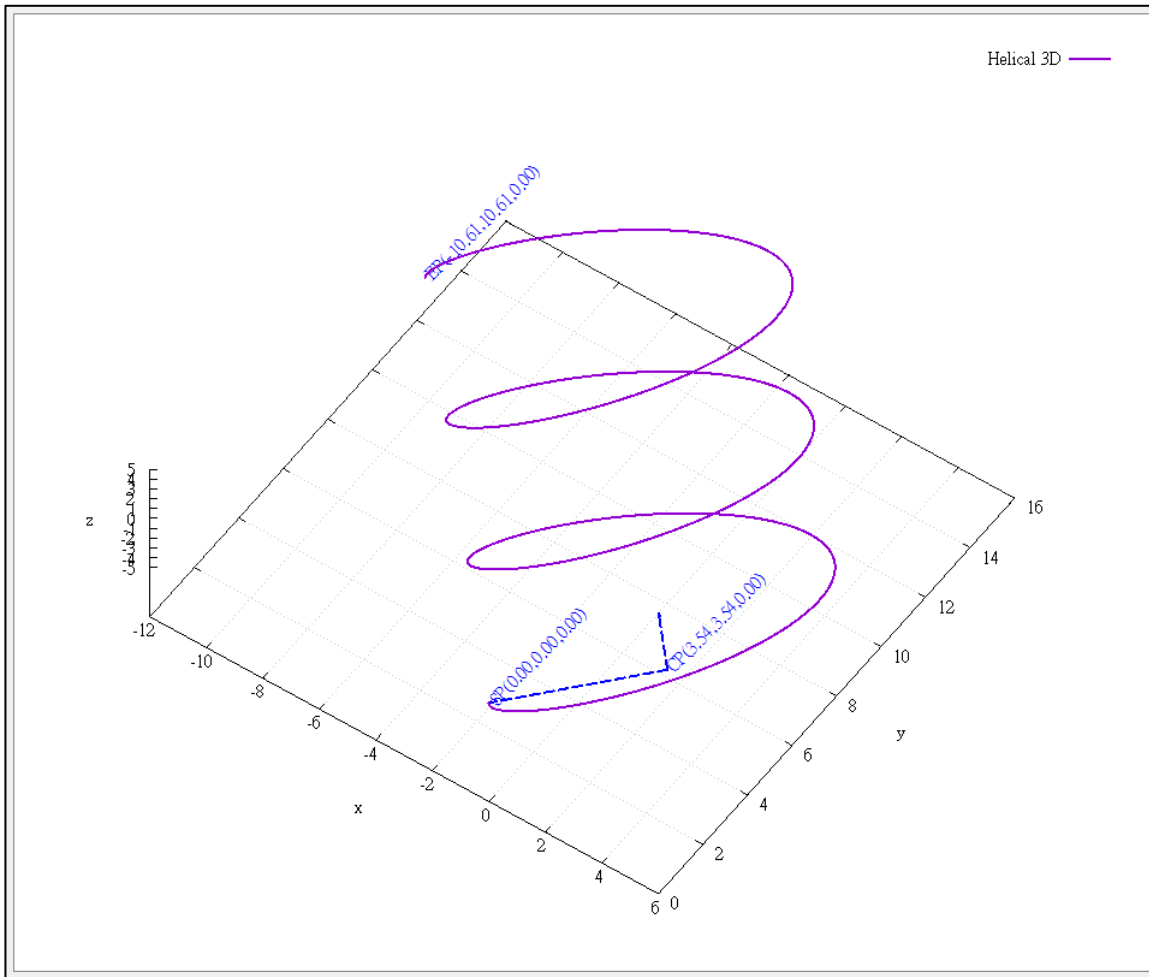
使用範例的 3D 螺旋補間移動軌跡(右螺旋):



**使用範例的螺旋補間移動軌跡(左螺旋):**

若旋轉角度參數設定成反方向，則移動軌跡更改為左螺旋

**CircAngle = -1080;**





## 7.8.28. ECAT\_McGroupMove3DHelicalRel\_CP\_Angle\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DHelicalRel\_CP\_Angle"  
根據中心位置及旋轉角度，開始指定群組相對距離 3D 螺旋補間移動。

### 格式:

int32\_t ECAT\_McGroupMove3DHelicalRel\_CP\_Angle\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[])

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定螺旋圈數(單位:degree)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
NV	double[]	IN	法向量

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

---

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

    NV [0] = -0.7071; //Normal Vector
    NV [1] = 0.7071; //Normal Vector
    NV [2] = 0.0;     //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DHelicalRel_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel
        , CircAngle, HelicalPitch, CircAuxPos, NV);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

---

## 7.8.29. ECAT\_McGroupMoveConicalHelixAbs\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveConicalHelixAbs"  
開始指定群組絕對距離圓錐螺旋補間移動。

### 格式:

int32\_t ECAT\_McGroupMoveConicalHelixAbs\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel, double EndRadius)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Angle	double	IN	指定螺旋圈數(單位:degree)
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)
EndRadius	double	IN	結束位置半徑(單位: user unit)

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

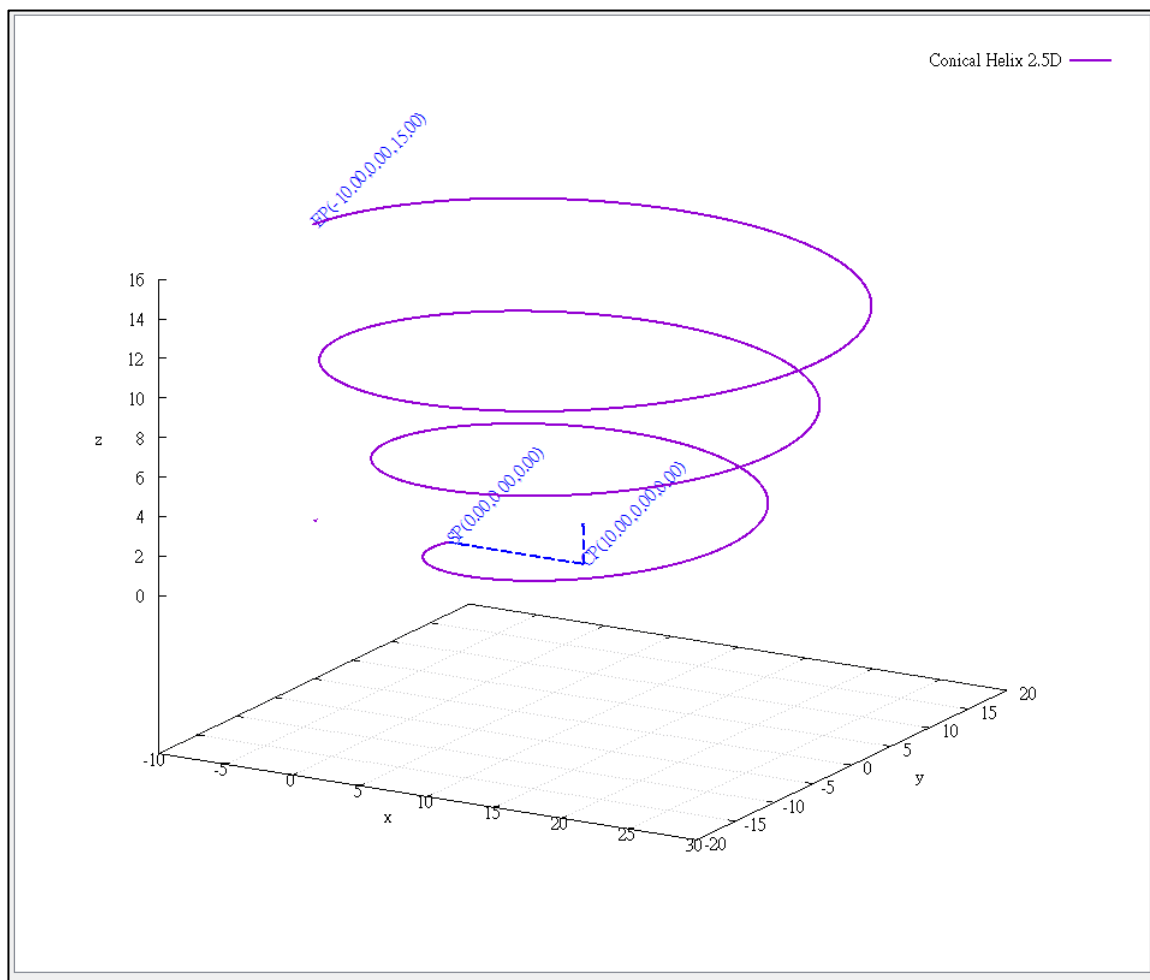
    //Command 1
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMoveConicalHelixAbs_Ex(DeviceNo, GroupNo
        , CircAngle, CircAuxPos, HelicalPitch, GroupVel, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

使用範例的圓錐螺旋補間移動軌跡:



## 7.8.30. ECAT\_McGroupMoveConicalHelixRel\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveConicalHelixRel"  
開始指定群組相對距離圓錐螺旋補間移動。

### 格式:

```
int32_t ECAT_McGroupMoveConicalHelixRel_Ex(uint16_t DeviceNo, uint16_t GroupNo,
double Angle, double AuxPos[], double Pitch, double Vel, double EndRadius)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Angle	double	IN	指定螺旋圈數(單位:degree)
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
Vel	double	IN	移動速度(單位: user unit/s)
EndRadius	double	IN	結束位置半徑(單位: user unit)

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

```

---

```
CircAuxPos[0] = 10.0; //Center Position
CircAuxPos[1] = 0.0; //Center Position
CircAngle = 1080;
HelicalPitch = 5;
GroupVel = 5;
EndRadius = 20;
ret = ECAT_McGroupMoveConicalHelixRel_Ex(DeviceNo, GroupNo
    , CircAngle, CircAuxPos, HelicalPitch, GroupVel, EndRadius);
if(ret < 0)
    printf("Failed to add group move conical helix command:%d\n",ret);

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move circular successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

### 7.8.31. ECAT\_McGroupMove3DConicalHelixAbs\_CP\_Angle\_Ex

#### 說明:

取代舊的函式"ECAT\_McGroupMove3DConicalHelixAbs\_CP\_Angle"

開始指定群組絕對位置 3D 圓錐螺旋補間移動。

#### 格式:

int32\_t ECAT\_McGroupMove3DConicalHelixAbs\_CP\_Angle\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[], double EndRadius)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定螺旋圈數(單位:degree)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
AuxPos	double[]	IN	絕對座標中心點位置(單位:user unit)
NV	double[]	IN	法向量
EndRadius	double	IN	結束位置半徑(單位: user unit)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    //Command 2
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

    NV [0] = -0.7071; //Normal Vector
    NV [1] = 0.7071; //Normal Vector
    NV [2] = 0.0;     //Normal Vector

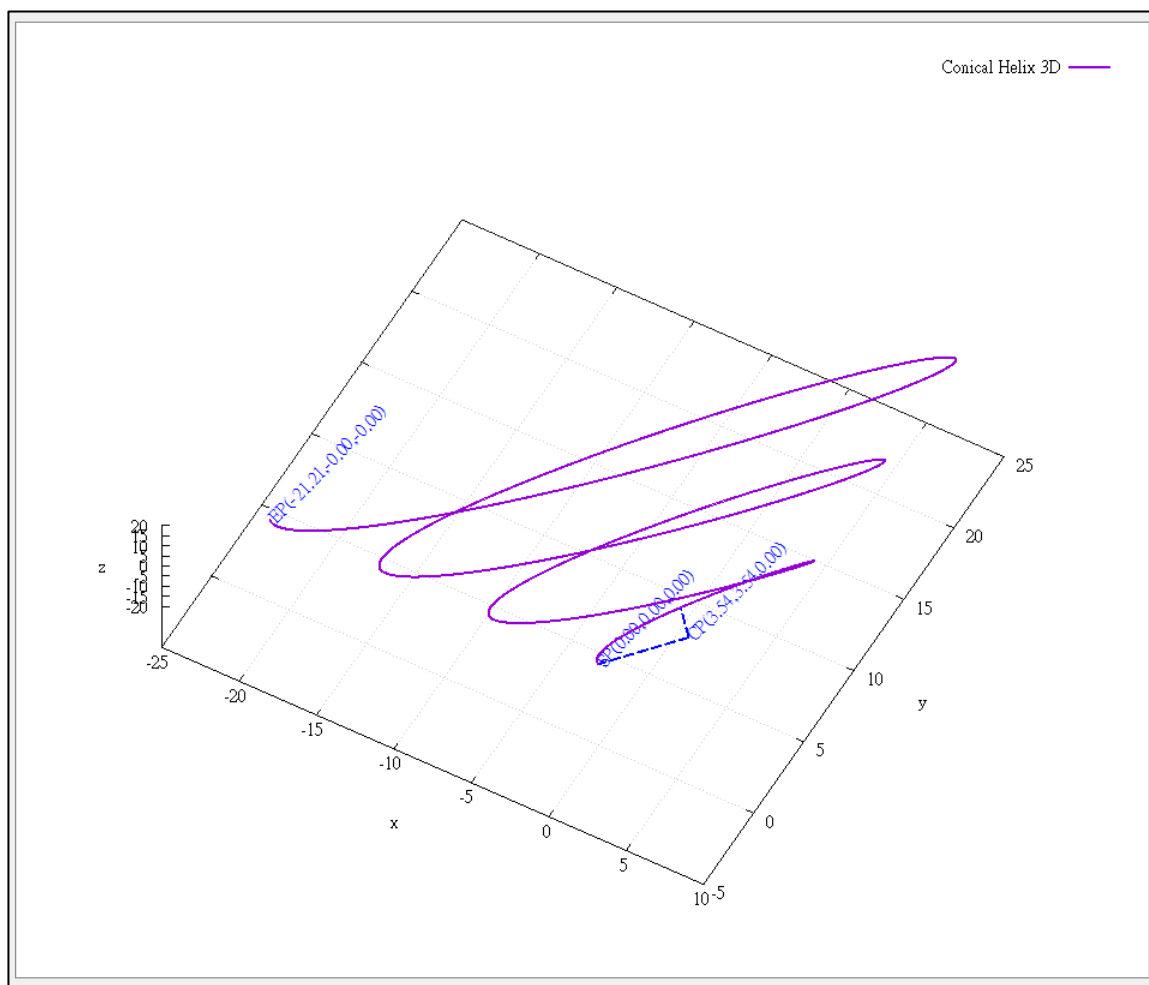
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMove3DConicalHelixAbs_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel
        , CircAngle, HelicalPitch, CircAuxPos, NV, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
```

```
printf("Group move circular successfully!\n");  
else if(State == MC_GS_ERRORSTOP) //ErrorStop  
{  
    printf("Group error stop\n");  
}  
}
```

使用範例的圓錐螺旋補間移動軌跡:



## 7.8.32. ECAT\_McGroupMove3DConicalHelixRel\_CP\_Angle\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMove3DConicalHelixRel\_CP\_Angle"

開始指定群組相對距離 3D 圓錐螺旋補間移動。

### 格式:

```
int32_t ECAT_McGroupMove3DConicalHelixRel_CP_Angle_Ex(uint16_t DeviceNo,
uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[],
double EndRadius)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Vel	double	IN	移動速度(單位: user unit/s)
Angle	double	IN	指定螺旋圈數(單位:degree)
Pitch	double	IN	指定螺旋螺距(單位: user unit)
AuxPos	double[]	IN	相對距離中心點位置(單位:user unit)
NV	double[]	IN	法向量
EndRadius	double	IN	結束位置半徑(單位: user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355; //Center Position
    CircAuxPos [1] = 3.5355; //Center Position
    CircAuxPos [2] = 0.0;     //Center Position

    NV [0] = -0.7071; //Normal Vector
    NV [1] = 0.7071; //Normal Vector
    NV [2] = 0.0;     //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMove3DConicalHelixRel_CP_Angle_Ex(DeviceNo, GroupNo, GroupVel
        , CircAngle, HelicalPitch, CircAuxPos, NV, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

### 7.8.33. ECAT\_McGroupMoveProfile\_Ex

#### 說明:

取代舊的函式"ECAT\_McGroupMoveProfile"  
開始執行指定群組 Profile 運動控制。

#### 格式:

int32\_t ECAT\_McGroupMoveProfile\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t ProfileNo[], uint16\_t TotalStep)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
ProfileNo	uint16_t[]	IN	Profile 編號，可用編號範圍 0~15
TotalStep	uint16_t	IN	總移動筆數

#### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t ProfileNo[MC_AXIS_NO_MAX];
uint16_t TotalStep = 1000;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    ProfileNo[0] = 0;
    ProfileNo[1] = 1;
    ret = ECAT_McGroupMoveProfile_Ex(DeviceNo, GroupNo, ProfileNo, TotalStep);
    if(ret < 0)
    {
        printf("Failed to start group move profile:%d\n",ret);
    }
}

do

```

---

```
{  
    sleep(1);  
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);  
  
}while(State == MC_GS_MOVING) //Moving  
  
if(State == MC_GS_STANDBY) //Standby  
    printf("Group move profile successfully!\n");  
else if(State == MC_GS_ERRORSTOP) //ErrorStop  
{  
    printf("Group error stop\n");  
}  
}
```

---

## 7.8.34. ECAT\_McGroupMoveProfileCSV\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveProfileCSV"

開始執行指定群組 Profile 運動控制，格式如圖 7.1 所示。

### 格式:

int32\_t ECAT\_McGroupMoveProfileCSV\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t ProfileNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
ProfileNo	uint16_t	IN	Profile 編號，可用編號範圍 0~15

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t ProfileNo = 0;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    ret = ECAT_McGroupMoveProfileCSV_Ex(DeviceNo, GroupNo, ProfileNo);
    if(ret < 0)
    {
        printf("Failed to start group move profile CSV:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }
}

```

```
}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move profile successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---

## 7.8.35. ECAT\_McGroupMoveDwell\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveDwell"  
群組連續運動中加入暫停指定時間。

### 格式:

int32\_t ECAT\_McGroupMoveDwell\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint32\_t Cnt)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Cnt	Uint32_t	IN	暫停時間，週期時間的倍數

### 回傳值:

0: 函式執行成功。  
其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint32_t DwellTime;

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {

```

---

```
    printf("Failed to add group move line command:%d\n",ret);
}
DwellTime = 500; //Wait 500ms, If cycletime = 1ms
ret = ECAT_McGroupMoveDwell_Ex(DeviceNo, GroupNo, DwellTime);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n",ret);
}

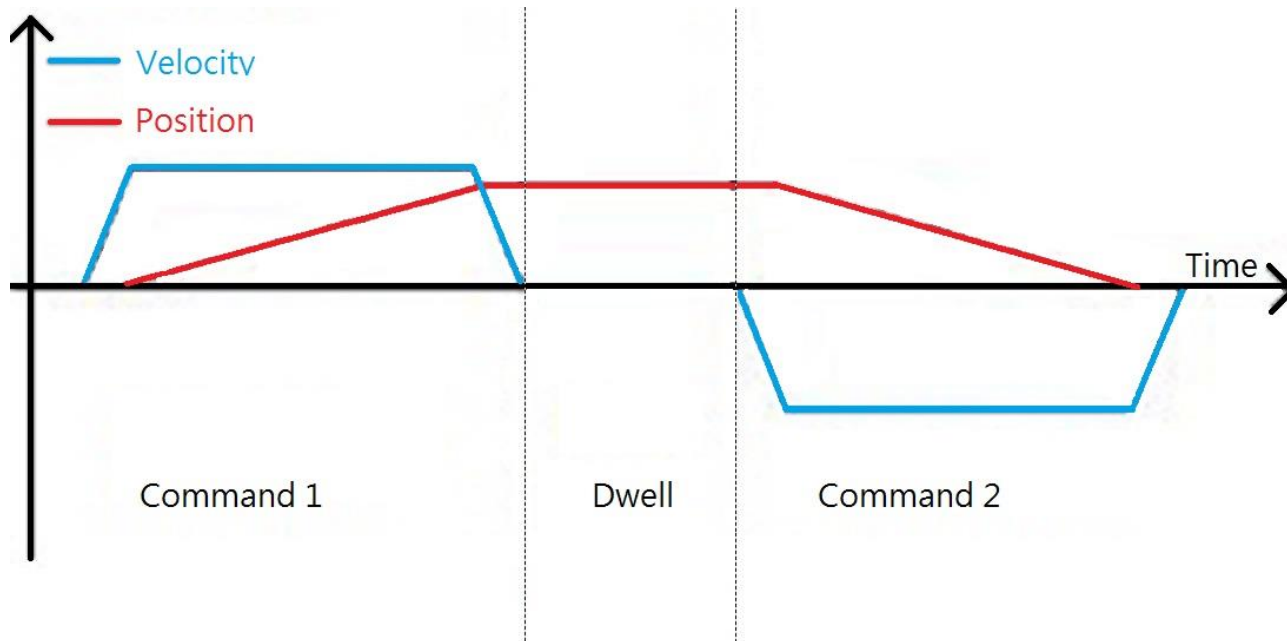
//Command 2
GroupPos[0] = 0;
GroupVel = 10;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n",ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---



## 7.8.36. ECAT\_McGroupMoveDO

### 說明:

群組連續運動中加入指定從站 DO 輸出命令。

### 格式:

```
int32_t ECAT_McGroupMoveDO(uint16_t DeviceNo, uint16_t GroupNo, uint16_t SlaveNo, uint16_t BitNo, uint32_t Value)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
SlaveNo	uint16_t	IN	從站編號
BitNo	uint16_t	IN	指定位元
Value	uint32_t	IN	輸出埠資料

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t SlaveNo, BitNo, Value;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby

```

```
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    SlaveNo = 3;
    BitNo = 1;
    Value = 1;
    DwellTime = 1000; //Wait 1s, If cycletime = 1ms
    ret = ECAT_McGroupMoveDO(DeviceNo, GroupNo, SlaveNo, BitNo, Value);
    if(ret < 0)
    {
        printf("Failed to add group move DO command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```



## 7.8.37. ECAT\_McGroupMoveAO

### 說明:

群組連續運動中加入指定從站 AO 輸出命令。

注:RunMode 為 0 :使用者用給定 16 位元數值方式來設定從站模組類比輸出的電壓值。

RunMode 為 1:使用者以輸入電壓範圍內的浮點數的方式來設定從站模組類比輸出的電壓值(需先使用 [ECAT\\_SetSlaveAoProperty](#) 進行設定)。

### 格式:

```
int32_t ECAT_McGroupMoveAO(uint16_t DeviceNo, uint16_t GroupNo, uint16_t SlaveNo, uint32_t RunMode, uint16_t ChannelNo, uint16_t RawData, double VoltData)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
SlaveNo	uint16_t	IN	從站編號
RunMode	uint32_t	IN	輸出模式
ChannelNo	uint16_t	IN	通道編號
RawData	uint16_t	IN	16 位元電壓值
VoltData	double	IN	浮點電壓值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t SlaveNo, ChannelNo, RawData;
double VoltData = 0;
uint32_t RunMode = 0;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    SlaveNo = 3;
    ChannelNo = 0;
    RawData = 32767;
    DwellTime = 1000; //Wait 1s, If cycletime = 1ms
    ret = ECAT_McGroupMoveAO(DeviceNo, GroupNo, SlaveNo, RunMode, ChannelNo, RawData,
    VoltData);
    if(ret < 0)
    {
        printf("Failed to add group move AO command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```



## 7.8.38. ECAT\_McGroupMoveBlendingSync\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveBlendingSync"

當群組運動命令為 blending mode 時，加入此命令，會等待減速停止完成後，再執行下一個運動命令。

### 格式:

int32\_t ECAT\_McGroupMoveBlendingSync\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BLENDING; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint32_t DwellTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 5.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {

```

---

```
    printf("Failed to add group move line command:%d\n",ret);
}
//Command 2
ret = ECAT_McGroupMoveBlendingSync_Ex(DeviceNo, GroupNo);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n",ret);
}

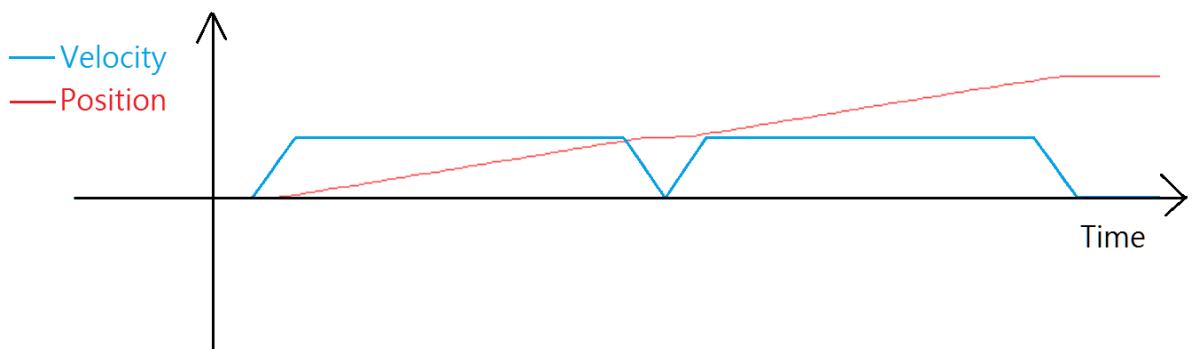
//Command 3
GroupPos[0] = 10.0;
GroupVel = 5;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n",ret);
}

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---



## 7.8.39. ECAT\_McGroupStop

### 說明:

停止群組運動，執行此命令將減速後停止運動。

### 格式:

```
int32_t ECAT_McGroupStop(uint16_t DeviceNo, uint16_t GroupNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

nt32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_DISABLED) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }

    sleep(1000);
    ret = ECAT_McGroupStop(DeviceNo,GroupNo);
    if(ret < 0)
    {
        printf("Failed to stop group move:%d\n",ret);
    }
    return;
}

```

---

```
    }  
    else  
    {  
        do  
        {  
            sleep(1);  
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);  
        }while(State == MC_GS_STOPPING) //Stopping  
  
        if(State == MC_GS_STANDBY) //Standby  
            printf("Group move stop successfully!\n");  
        else if(State == MC_GS_ERRORSTOP) //ErrorStop  
        {  
            printf("Group error stop\n");  
        }  
    }  
}
```

---

## 7.8.40. ECAT\_McGroupQuickStop

### 說明:

快速停止群組運動，執行此命令將立即停止運動。

### 格式:

```
int32_t ECAT_McGroupQuickStop(uint16_t DeviceNo, uint16_t GroupNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

nt32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_DISABLED) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }

    sleep(1000);
    ret = ECAT_McGroupQuickStop(DeviceNo, GroupNo);
    if(ret < 0)
    {
        printf("Failed to stop group move:%d\n",ret);
    }
    return;
}

```

---

```
    }  
    else  
    {  
        do  
        {  
            sleep(1);  
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);  
        }while(State == MC_GS_STOPPING) //Stopping  
  
        if(State == MC_GS_STANDBY) //Standby  
            printf("Group move stop successfully!\n");  
        else if(State == MC_GS_ERRORSTOP) //ErrorStop  
        {  
            printf("Group error stop\n");  
        }  
    }  
}
```

---

## 7.8.41. ECAT\_McSetGroupHold\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupHold"

暫停群組運動，執行此命令，群組狀態變為 MC\_GS\_HOLD，當前命令執行完後暫停群組運動，若當前無命令運行中，則之後下達的群組命令不會立即執行，直到解除群組暫停狀態。

注:PVT 運動命令與其他群組運動命令不可混用。

### 格式:

```
int32_t ECAT_McSetGroupHold_Ex(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Status	uint16_t	IN	0:解除群組暫停狀態 1:暫停群組命令

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending

```

```

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    ret = ECAT_McSetGroupHold_Ex(DeviceNo, GroupNo, 1); // hold
    if(ret < 0)
    {
        printf("Failed to set group hold:%d\n",ret);
    }

    //Command 1
    GroupPos[0] = 5.0;
    GroupVel = 10;

```

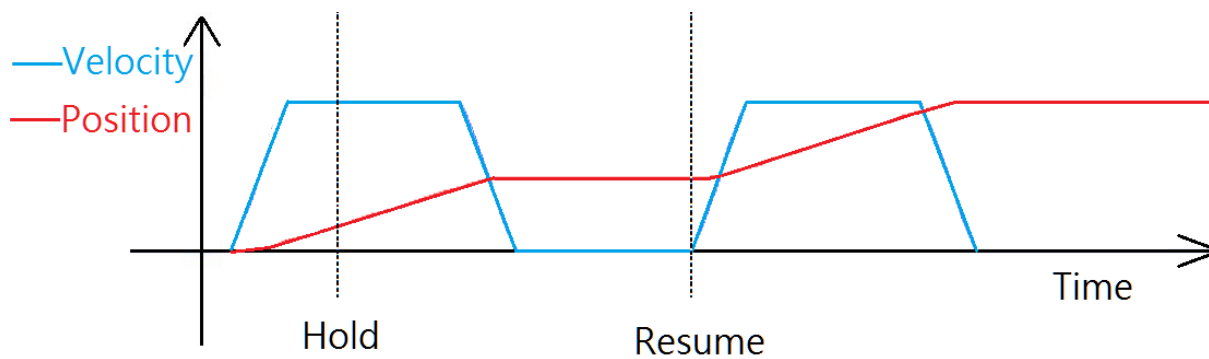
```
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
    printf("Failed to add group move line command:%d\n",ret);
//Command 2
GroupPos[0] = 10.0;
GroupVel = 10;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
if(ret < 0)
    printf("Failed to add group move line command:%d\n",ret);
usleep(200000); //sleep 200 ms

ret = ECAT_McSetGroupHold_Ex(DeviceNo, GroupNo, 1 ); //Hold
if(ret < 0)
{
    printf("Failed to set group hold:%d\n",ret);
}
usleep(800000); //sleep 800 ms

ret = ECAT_McSetGroupHold_Ex(DeviceNo, GroupNo, 0 ); //Resume
if(ret < 0)
{
    printf("Failed to set group resume:%d\n",ret);
}
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```





## 7.8.42. ECAT\_McSetGroupPause\_Ex

### 說明:

取代舊的函式"ECAT\_McSetGroupPause"

暫停群組運動，在群組運動中執行此命令，群組狀態變為 **MC\_GS\_PAUSE**，當前命令立即減速暫停群組運動，直到解除群組暫停狀態。

### 格式:

int32\_t ECAT\_McSetGroupPause\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t Status)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Status	uint16_t	IN	0:解除群組暫停狀態 1:暫停群組命令

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint16_t CmdMode = MS_GRP_CM_ABORTING; //0: Aborting, 1: Buffered, 2: Blending

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 5.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);
    usleep(200000); //sleep 200 ms
    ret = ECAT_McSetGroupPause_Ex(DeviceNo, GroupNo, 1 ); //pause
    if(ret < 0)

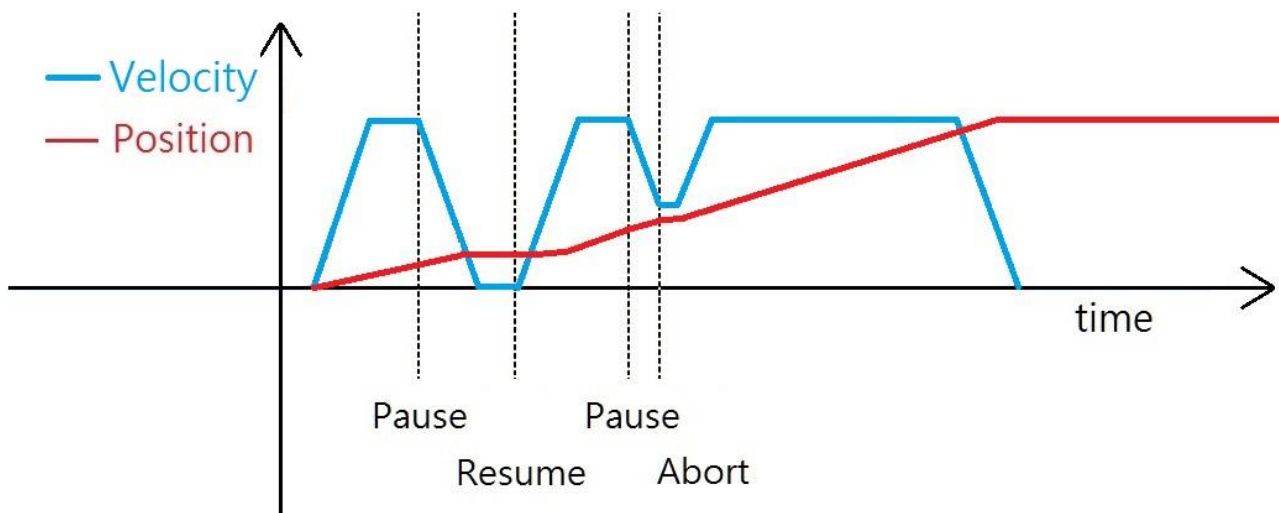
```

```
{
    printf("Failed to set group pause:%d\n",ret);
}
usleep(200000); //sleep 200 ms
ret = ECAT_McSetGroupPause_Ex(DeviceNo, GroupNo, 0 ); //resume
if(ret < 0)
    printf("Failed to set group resume:%d\n",ret);
usleep(200000); //sleep 200 ms
ret = ECAT_McSetGroupPause_Ex(DeviceNo, GroupNo, 1 ); //pause
if(ret < 0)
{
    printf("Failed to set group pause:%d\n",ret);
}
usleep(50000); //sleep 50ms

GroupPos[0] = 10.0;
GroupVel = 10;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel); //Abort
if(ret < 0)
    printf("Failed to add group move line command:%d\n",ret);

do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //Error Stop
{
    printf("Group error stop\n");
}
}
```



## 7.8.43. ECAT\_McAddPathData

### 說明:

加入一筆 Path 資料到指定佇列編號中。

### 格式:

```
int32_t ECAT_McAddPathData(uint16_t DeviceNo, uint16_t PathDataNo, uint32_t
CmdType, uint8_t AbsMove, double EndPos[], double AuxPos[], double Args[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PathDataNo	uint16_t	IN	佇列編號
CmdType	uint32_t	IN	命令種類(如表 7.14 所示)
AbsMove	uint8_t	IN	絕對移動設定 0: 相對移對 1: 絕對移動
EndPos	double []	IN	結束點位置
AuxPos	double []	IN	輔助位置
Args	double []	IN	參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.14: Path Data 命令種類定義

標頭檔定義名稱	數值	說明
MC_PATH_CMD_TYPE_MOVE_LINE	1	線性補間移動
MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE	2	2D 圓弧補間移動, 根據中心位置及旋轉角度參數決定
MC_PATH_CMD_TYPE_MOVE_CIRC_CP_EP	3	2D 圓弧補間移動, 根據中心位置及終點位置參數決定
MC_PATH_CMD_TYPE_MOVE_CIRC_BP_EP	4	2D 圓弧補間移動, 根據圓弧上一點位置及終點位置參數決定
MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_ANGLE	5	3D 圓弧補間移動, 根據中心位置及旋轉角度參數決定
MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_EP	6	3D 圓弧補間移動, 根據中心位置及終點位置參數決定
MC_PATH_CMD_TYPE_MOVE_3D_CIRC_BP_EP	7	3D 圓弧補間移動, 根據圓弧上一點位置及終點位置參數決定
MC_PATH_CMD_TYPE_MOVE_HELICAL	8	螺旋補間移動, 根據中心位置及旋轉角度參數決定
MC_PATH_CMD_TYPE_MOVE_3D_HELICAL	9	3D 螺旋補間移動, 根據中心位置及旋轉角度參數決定
MC_PATH_CMD_TYPE_MOVE_CONICAL_HELIX	10	圓錐螺旋補間移動, 根據中心位置及旋轉角度參數決定
MC_PATH_CMD_TYPE_MOVE_3D_CONICAL_HELIX	11	3D 圓錐螺旋補間移動, 根據中心位置及旋轉角度參數決定
MC_PATH_CMD_TYPE_SET_ACCDEC_TIME	12	設定加速度時間
MC_PATH_CMD_TYPE_SET_ACCDEC_TYPE	13	設定加速度種類
MC_PATH_CMD_TYPE_SET_CMD_MODE	14	設定混合模式
MC_PATH_CMD_TYPE_SET_BLEND_PERCENT	15	設定混合百分比
MC_PATH_CMD_TYPE_DWELL	16	等待指定時間

MC_PATH_CMD_TYPE_SET_DO	17	輸出 DO
MC_PATH_CMD_TYPE_SET_AO_VOLT	18	輸出 AO volt
MC_PATH_CMD_TYPE_TANGENT_IN	19	執行指定軸號切線跟隨運動控制
MC_PATH_CMD_TYPE_TANGENT_OUT	20	停止指定軸號切線跟隨運動控制
MC_PATH_CMD_TYPE_BLENDINGSYNC	21	等待減速停止完成後，再執行下一個運動命令

### 使用範例:

#### [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double EndPos[MC_AXIS_NO_MAX];
double AuxPos[3];
double Args[MC_PATH_DATA_ARGS_MAX];

/*****Move Line*****/
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                        AbsMove, EndPos, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move Circular (CP, ANGLE)*****/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 315; //Angle

```



```

AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE,
AbsMove, NULL, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move Circular (CP, EP)*****/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
EndPos[0] = 8.535533; //End Position
EndPos[1] = 3.535533; //End Position
Args[0] = 4; //Velocity
Args[1] = 1; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move Circular (BP, EP)*****/
AuxPos[0] = 5; //Border Position
AuxPos[1] = 5; //Border Position
EndPos[0] = 8.535533; //End Position
EndPos[1] = 3.535533; //End Position
Args[0] = 4; //Velocity
Args[1] = 0; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_BP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move 3D Circular (CP, ANGLE)*****/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = -0.7071; //Normal vector
EndPos[1] = 0.7071; //Normal vector

```

```
EndPos[2] = 0; //Normal vector
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo,
MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_ANGLE, AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move 3D Circular (CP, EP)*****/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = 1.03552; //End Position
EndPos[1] = 1.03552; //End Position
EndPos[2] = -3.53547; //End Position
Args[0] = 4; //Velocity
Args[1] = 0; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move 3D Circular (CP, EP)*****/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = 1.03552; //End Position
EndPos[1] = 1.03552; //End Position
EndPos[2] = -3.53547; //End Position
Args[0] = 4; //Velocity
Args[1] = 0; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);
```

```
/******Move 3D Circular (BP, EP)******/
AuxPos[0] = 1.036; //Border Position
AuxPos[1] = 1.036; //Border Position
AuxPos[2] = 3.5355; //Border Position
EndPos[0] = 6.035; //End Position
EndPos[1] = 6.035; //End Position
EndPos[2] = -3.53547; //End Position
Args[0] = 4; //Velocity
Args[1] = 1; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_CIRC_BP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/******Move Helical******/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_HELICAL,
AbsMove, NULL, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/******Move 3D Helical******/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = -0.7071; //Normal vector
EndPos[1] = 0.7071; //Normal vector
EndPos[2] = 0; //Normal vector
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
```

```
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_HELICAL,
                          AbsMove, EndPos, AuxPos, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move Conical Helix*****/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
Args[3] = 10; //End Radius
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CONICAL_HELIX,
                          AbsMove, NULL, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move 3D Conical Helix*****/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = -0.7071; //Normal vector
EndPos[1] = 0.7071; //Normal vector
EndPos[2] = 0; //Normal vector
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
Args[3] = 10; //End Radius
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo,
                          MC_PATH_CMD_TYPE_MOVE_3D_CONICAL_HELIX, AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Set Acc Time*****/
rgs[0] = 900; //ms
```

```

ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_ACCDEC_TIME,
                          0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Set Acc Type*****/
Args[0] = 2; // 1:T-Curve, 2:S-Curve
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_ACCDEC_TYPE,
                          0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Set Cmd Mode*****/
Args[0] = MS_GRP_CM_BLENDING; //MS_GRP_CM_BUFFERED or MS_GRP_CM_BLENDING
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_CMD_MODE,
                          0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Set Bledn Percent*****/
Args[0] = 50; //50%
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_BLEND_PERCENT,
                          0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Execute Dwell*****/
Args[0] = 2000; //Cycle time tick
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_DWELL,
                          0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Output DO*****/
Args[0] = 3; //SlaveNo
Args[1] = 1; //BitNo
Args[2] = 0; //Do Value
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_DO,

```

---

```
                                0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Output AO*****/
Args[0] = 4; //SlaveNo
Args[1] = 1; //ChannelNo
Args[2] = 7.5; //Volt
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_AO_VOLT,
                                0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Tangent In*****/
Args[0] = 2; //AxisNo
Args[1] = 90; //Angle
Args[2] = 90; //Velocity
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_TANGENT_IN,
                                0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Tangent Out*****/
Args[0] = 2; //AxisNo
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_TANGENT_OUT,
                                0, NULL, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Blending Sync*****/
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_BLENDINGSYNC,
                                0, NULL, NULL, NULL);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);
```

---

## 7.8.44. ECAT\_McSetPathData

### 說明:

修改指定索引 Path 資料。

### 格式:

```
int32_t ECAT_McSetPathData(uint16_t DeviceNo, uint16_t PathDataNo, uint16_t
DataIndex, uint32_t CmdType, uint8_t AbsMove, double EndPos[], double AuxPos[], double
Args[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PathDataNo	uint16_t	IN	佇列編號
DataIndex	uint16_t	IN	資料索引
CmdType	uint32_t	IN	命令種類(如表 7.14 所示)
AbsMove	uint8_t	IN	絕對移動設定 0: 相對移對 1: 絕對移動
EndPos	double []	IN	結束點位置
AuxPos	double []	IN	輔助位置
Args	double []	IN	參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double   EndPos[MC_AXIS_NO_MAX];
double   AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];

/*****Move Line*****/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                          AbsMove, EndPos, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move Circular *****/
//data index 1
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE,
                          AbsMove, NULL, AuxPos, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

//modify data index 1
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position

```



---

```
EndPos[0] = 8.535533; //End Position
EndPos[1] = 3.535533; //End Position
Args[0] = 4; //Velocity
Args[1] = 1; //Dir
AbsMove = 0;
ret = ECAT_McSetPathData(DeviceNo, PathDataNo, 1, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_EP,
                          AbsMove, EndPos, AuxPos, Args);

if(ret != 0)
    printf("Failed to set path data:%d\n",ret);
```

---

## 7.8.45. ECAT\_McGetPathData

### 說明:

取得指定索引 Path 資料。

### 格式:

```
int32_t ECAT_McGetPathData(uint16_t DeviceNo, uint16_t PathDataNo, uint16_t
DataIndex, uint32_t *CmdType, uint8_t *AbsMove, double EndPos[], double AuxPos[],
double Args[])
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PathDataNo	uint16_t	IN	佇列編號
DataIndex	uint16_t	IN	資料索引
CmdType	uint32_t *	OUT	命令種類(如表 7.14 所示)
AbsMove	uint8_t *	OUT	絕對移動設定 0: 相對移對 1: 絕對移動
EndPos	double []	OUT	結束點位置
AuxPos	double []	OUT	輔助位置
Args	double []	OUT	參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double   EndPos[MC_AXIS_NO_MAX];
double   AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];
uint32_t CmdType;
uint8_t AbsMove;
/*****Move Line*****/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                        AbsMove, EndPos, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

ret = ECAT_McGetPathData(DeviceNo, PathDataNo, 0, &CmdType, &AbsMove, EndPos, AuxPos, Args);
if(ret != 0) {
    printf("Failed to get path data:%d\n",ret);
} else {
    printf("Cmd Type:%u\n", CmdType);
    printf("Abs. Move:%u\n", AbsMove);
    printf("EndPos[0]:%u\n", EndPos[0]);
    printf("EndPos[1]:%u\n", EndPos[1]);
    printf("Args[0]:%u\n", Args[0]);
}

```

## 7.8.46. ECAT\_McClearPathData

## 說明:

清除指定佇列編號 Path 資料。

**格式:**

```
int32_t ECAT_McClearPathData(uint16_t DeviceNo, uint16_t PathDataNo)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PathDataNo	uint16_t	IN	佇列編號

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double   EndPos[MC_AXIS_NO_MAX];
double   AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];
/*****Move Line*****/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                        AbsMove, EndPos, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

ret = McClearPathData(DeviceNo, PathDataNo);
if(ret != 0) {
    printf("Failed to clear path data:%d\n",ret);
}
```

## 7.8.47. ECAT\_McGetPathDataSize

### 說明:

取得指定佇列編號已儲存 Path 資料筆數。

### 格式:

```
int32_t ECAT_McGetPathDataSize(uint16_t DeviceNo, uint16_t PathDataNo, uint16_t *Size)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PathDataNo	uint16_t	IN	佇列編號
Size	uint16_t *	OUT	Path 資料筆數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double   EndPos[MC_AXIS_NO_MAX];
double   AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];
uint16_t Size;

/*****Move Line*****/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                        AbsMove, EndPos, NULL, Args);

if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

ret = ECAT_McGetPathDataSize(DeviceNo, PathDataNo, &Size);
if(ret != 0) {
    printf("Failed to get path data size:%d\n",ret);
} else {
    printf("path data size:%u\n", Size);
}

```

## 7.8.48. ECAT\_McGroupMovePath

### 說明:

開始執行指定群組 Path 運動控制。

### 格式:

int32\_t ECAT\_McGroupMovePath(uint16\_t DeviceNo, uint16\_t GroupNo, uint16\_t PathDataNo, uint8\_t Restart, uint16\_t DataIndex, uint8\_t Repeat)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PathDataNo	uint16_t	IN	佇列編號
Restart	uint8_t	IN	重新開始 0: 若 DataIndex 為 0 則從最後停止索引值開始執行，若 DataIndex 不為 0 則從指定索引值開始執行 1: 從索引 0 開始執行
DataIndex	uint16_t	IN	資料索引
Repeat	uint8_t	IN	重複循環執行

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double EndPos[MC_AXIS_NO_MAX];
double AuxPos[3];
double Args[MC_PATH_DATA_ARGS_MAX];

/*****Move Line*****/
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                        AbsMove, EndPos, NULL, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Move Circular (CP, ANGLE)*****/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE,
                        AbsMove, NULL, AuxPos, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*****Group Move Path*****/
ret = ECAT_McGroupMovePath(DeviceNo, GroupNo, PathDataNo, 1, 0, 1);
if(ret != 0)
    printf("Group move path failed:%d\n",ret);

```



## 7.8.49. ECAT\_McGroupMoveLineAbsAdv\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveLineAbsAdv"

執行指定群組的絕對位置線性補間移動。

### 格式:

int32\_t ECAT\_McGroupMoveLineAbsAdv\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double EndPos[], double StartVel, double ReqVel, double FinalVel, double Accel, double Decel, uint8\_t AccDecMode)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
EndPos	Double *	IN	目標位置(單位: user unit)
StartVel	double	IN	初始速度(單位: user unit/s)
ReqVel	double	IN	移動速度(單位: user unit/s)
FinalVel	double	IN	結束速度(單位: user unit/s)
Accel	double	IN	加速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
Decel	double	IN	減速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
AccDecMode	uint8_t	IN	加減速模式: 0: 加減速度率(user unit/s <sup>2</sup> ) 1: 加減速度時間(second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
```

```
final_vel = 2;
accel = 4;
decel = 4;
end_pos[0] = 1.5;
end_pos[1] = 1.5;

ret = ECAT_McGroupMoveLineAbsAdv_Ex(DeviceNo, GroupNo, end_pos,
    start_vel, req_vel, final_vel, accel, decel, 0);
if (ret != 0)
    printf("Failed to add group move line command:%d\n", ret);
```

#### //Command 2

```
start_vel = 2;
req_vels = 2.5;
final_vel = 1.5;
accel = 6;
decel = 6;
end_pos[0] = 3.5;
end_pos[1] = 3.5;

ret = ECAT_McGroupMoveLineAbsAdv_Ex(DeviceNo, GroupNo, end_pos,
    start_vel, req_vel, final_vel, accel, decel, 0);
if (ret != 0)
    printf("Failed to add group move line command:%d\n", ret);
```

#### //Command 3

```
start_vel = 1.5;
req_vels = 1.5;
final_vel = 0;
accel = 2;
decel = 2;
end_pos[0] = 5;
end_pos[1] = 5;

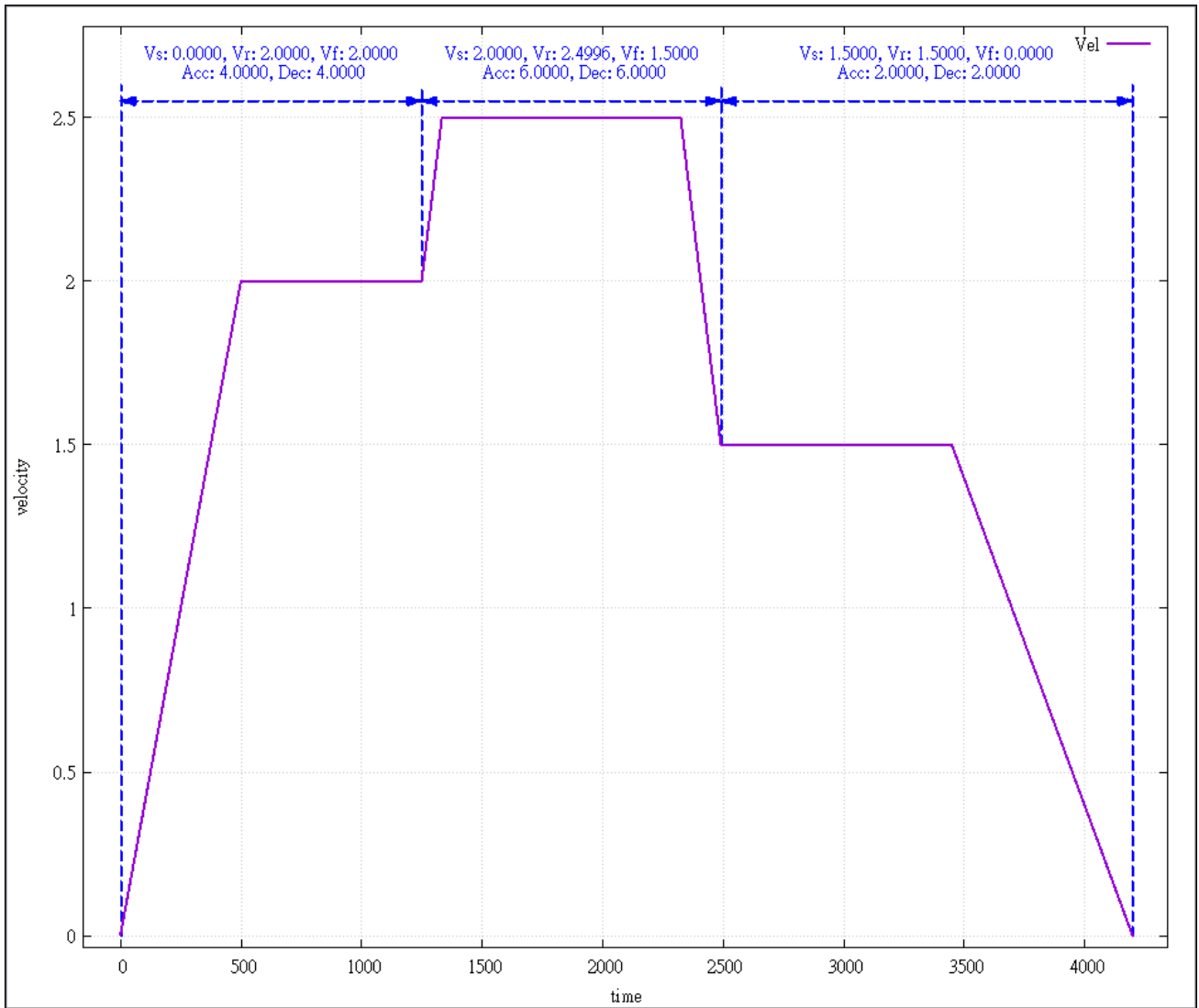
ret = ECAT_McGroupMoveLineAbsAdv_Ex(DeviceNo, GroupNo, end_pos,
    start_vel, req_vel, final_vel, accel, decel, 0);
if (ret != 0)
    printf("Failed to add group move line command:%d\n", ret);
```

```
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---



## 7.8.50. ECAT\_McGroupMoveLineRelAdv\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveLineRelAdv"

執行指定群組的相對位置線性補間移動。

### 格式:

int32\_t ECAT\_McGroupMoveLineRelAdv\_Ex(uint16\_t DeviceNo, uint16\_t GroupNo, double EndPos[], double StartVel, double ReqVel, double FinalVel, double Accel, double Decel, uint8\_t AccDecMode)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
EndPos	Double *	IN	目標位置(單位: user unit)
StartVel	double	IN	初始速度(單位: user unit/s)
ReqVel	double	IN	移動速度(單位: user unit/s)
FinalVel	double	IN	結束速度(單位: user unit/s)
Accel	double	IN	加速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
Decel	double	IN	減速度率 (user unit/s <sup>2</sup> ) 或加速度時間 (second)
AccDecMode	uint8_t	IN	加減速模式: 0: 加減速度率(user unit/s <sup>2</sup> ) 1: 加減速度時間(second)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
```

```
final_vel = 2;
accel = 4;
decel = 4;
end_pos[0] = 1.5;
end_pos[1] = 1.5;

ret = ECAT_McGroupMoveLineAbsAdv_Ex(DeviceNo, GroupNo, end_pos,
    start_vel, req_vel, final_vel, accel, decel, 0);
if (ret != 0)
    printf("Failed to add group move line command:%d\n", ret);
```

### //Command 2

```
start_vel = 2;
req_vels = 2.5;
final_vel = 1.5;
accel = 6;
decel = 6;
end_pos[0] = 3;
end_pos[1] = 3;

ret = ECAT_McGroupMoveLineRelAdv_Ex(DeviceNo, GroupNo, end_pos,
    start_vel, req_vel, final_vel, accel, decel, 0);
if (ret != 0)
    printf("Failed to add group move line command:%d\n", ret);
```

### //Command 3

```
start_vel = 1.5;
req_vels = 1.5;
final_vel = 0;
accel = 2;
decel = 2;
end_pos[0] = 1.5;
end_pos[1] = 1.5;

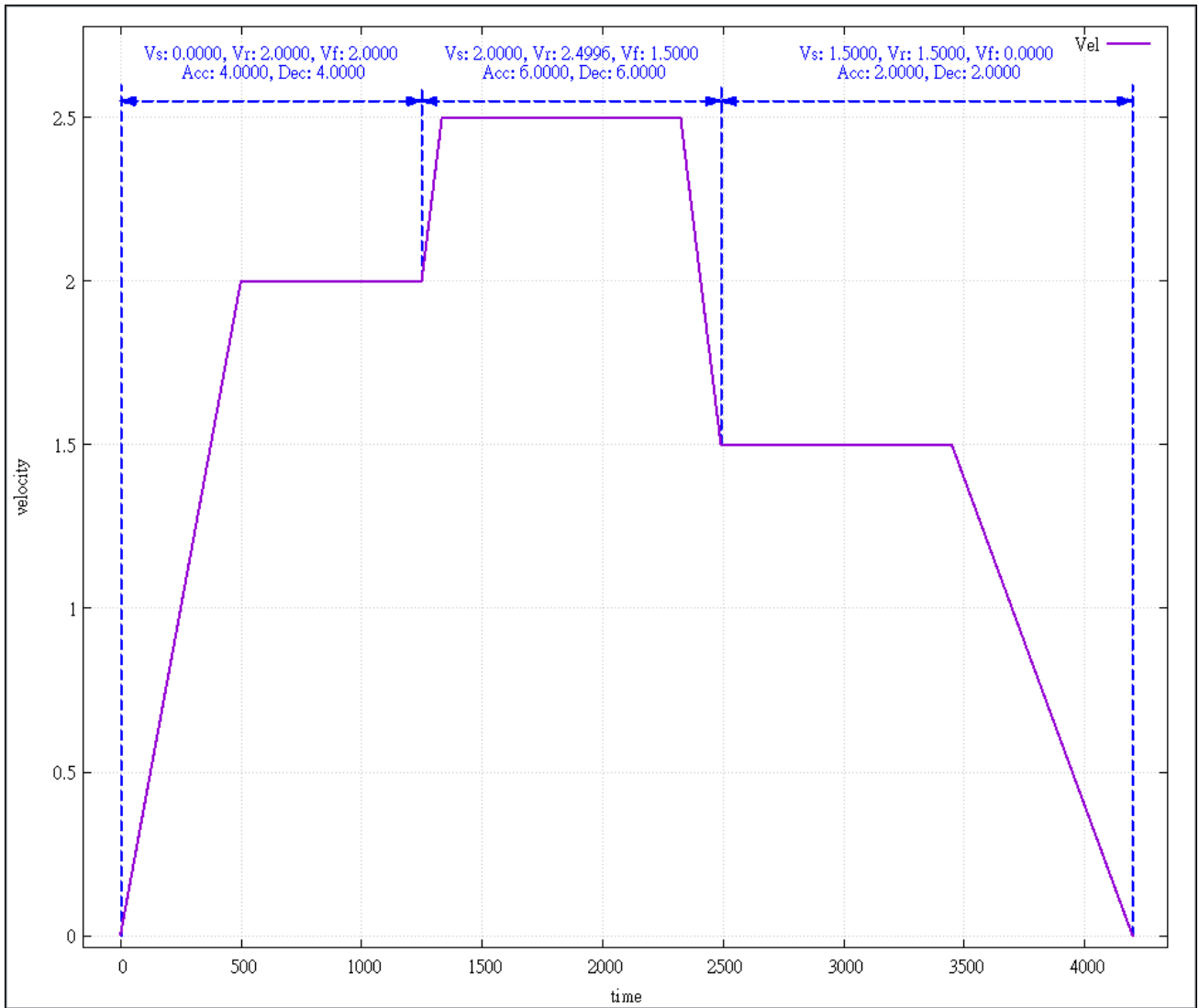
ret = ECAT_McGroupMoveLineRelAdv_Ex(DeviceNo, GroupNo, end_pos,
    start_vel, req_vel, final_vel, accel, decel, 0);
if (ret != 0)
    printf("Failed to add group move line command:%d\n", ret);
```

```
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move line successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

---



## 7.8.51. ECAT\_McGroupMoveShaker\_Ex

### 說明:

取代舊的函式"ECAT\_McGroupMoveShaker"

此功能僅供工廠測試使用

執行指定群組的相對正弦曲線移動。

注:(1)可與部分群組命令同時使用。

(2)不支援 blending 或 buffer 模式，在 blending 或 buffer 模式下可以使用，但會 abort 正在運行中的相對正弦曲線移動。

$$Y = \text{Amp} * \sin(2 * \pi * \text{Freq} * t + \text{phase})$$

t = 0 to Time

### 格式:

```
int32_t ECAT_McGroupMoveShaker_Ex(uint16_t DeviceNo, uint16_t GroupNo, double Amp, double phase[], double Freq, double Time);
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
GroupNo	uint16_t	IN	群組編號
Amp	double	IN	波幅(單位:user unit)
phase	double[]	IN	相位角(單位:degree)
Freq	double	IN	頻率(單位:hz)
Time	double	IN	移動時間(單位:秒)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double Amp;
double phase[MC_AXIS_NO_MAX];
double Freq;
double Time;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

```

---

```
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    Amp = 0.001;
    phase[0] =180;
    Freq = 20;
    Time = 1;
    ret = ECAT_McGroupMoveShaker_Ex(DeviceNo, GroupNo, Amp, phase, Freq, Time);
    if(ret < 0)
    {
        printf("Failed to add group move shaker command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

---

## 7.8.52. ECAT\_McAxisTangentInGroup

### 說明:

開始執行指定軸號切線跟隨運動控制。

注: (1) 請將此軸 PPU 設為一轉所需的 pulse 數

(2) 請將此軸啟用無限旋轉功能，位置最小軟體極限設為 0，位置最大軟體極限設為 1

### 格式:

int32\_t ECAT\_McAxisTangentInGroup(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t GroupNo, double Angle, double Vel)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
GroupNo	uint16_t	IN	群組編號
Angle	int16_t	IN	切入角度(單位:degree)
Vel	uint16_t	IN	切入角度執行速度(單位: degree/s)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t TangentInAxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
double AxisAngle, AxisVel, CircAngle;
uint32_t task_index;
bool task_stop;

/*****/
int32_t check_grp_state(void)
{
    int32_t ret;
    uint32_t State;
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    if (ret != 0) {
        printf("Failed to get group state:%d\n", ret);
        return -1;
    } else {
        if(State == MC_GS_ERRORSTOP) {
            printf("Group error stop\n");
            return -1;
        } else if(State == MC_GS_STANDBY) {
            return 0;
        } else
            return 1;
    }
}
/*****/
int main()
{

```

```
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return -1;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return -1;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return -1;
}

ret = check_grp_state();
if (ret == -1)
    return -1;

task_index = 0;
task_stop = false;
TangentInAxisNo = 2;

while(!task_stop) {
    switch(task_index) {
        case 0:
            GroupPos[0] = 0.0;
            GroupPos[1] = 0.0;
            GroupVel = 5;
            ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
            if(ret != 0) {
                printf("Failed to add group move line command:%d\n", ret);
            }
        }
    }
}
```

```
        task_stop = true;
    }

    AxisAngle = 90;
    AxisVel = 90;
    ret = ECAT_McAxisTangentInGroup(DeviceNo, TangentInAxisNo, GroupNo, AxisAngle,
AxisVel);

    if (ret != 0) {
        printf("Axis tangent in failed:%d\n", ret);
        task_stop = true;
    }

    GroupPos[0] = 0.0;
    GroupPos[1] = 10.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret != 0) {
        printf("Failed to add group move line command:%d\n", ret);
        task_stop = true;
    }

    AxisAngle = 0;
    AxisVel = 90;
    ret = ECAT_McAxisTangentInGroup(DeviceNo, TangentInAxisNo, GroupNo, AxisAngle,
AxisVel);

    if (ret != 0) {
        printf("Axis tangent in failed:%d\n", ret);
        task_stop = true;
    }

    GroupPos[0] = 0.0;
    GroupPos[1] = -2.5;
    GroupVel = 0.5;
    CircAngle = -180;
    ret = ECAT_McGroupMoveCircularRel_CP_Angle(DeviceNo, GroupNo, GroupVel,
CircAngle, GroupPos);
    if (ret != 0) {
        printf("Group move circular failed:%d\n", ret);
    }
}
```

```

        task_stop = true;
    }

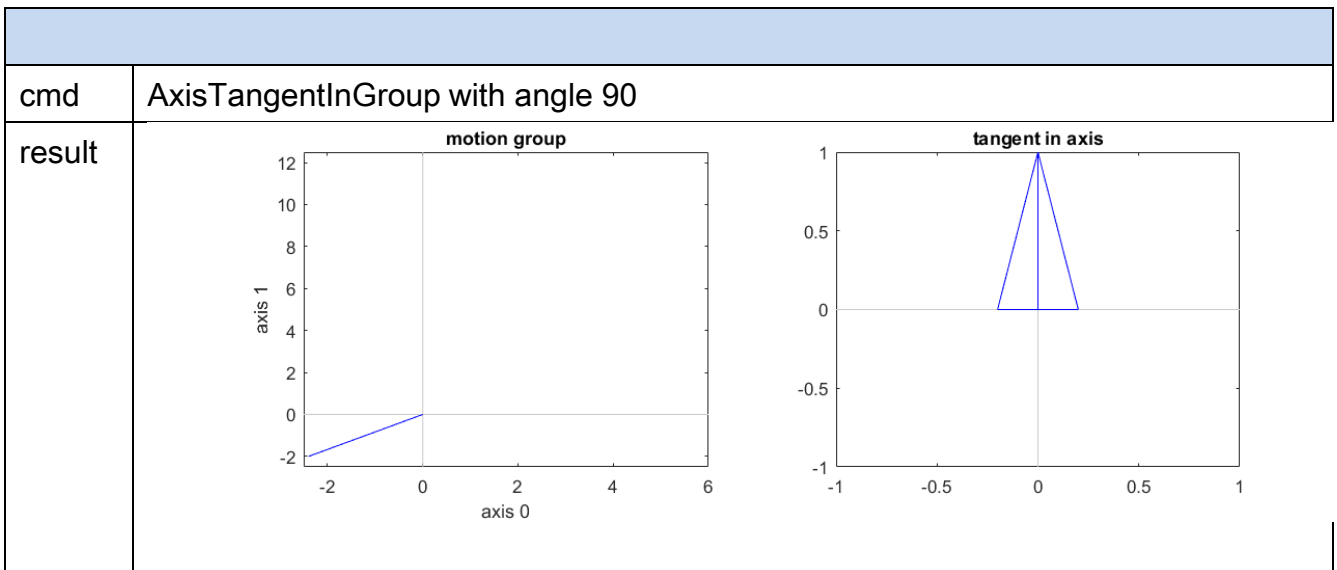
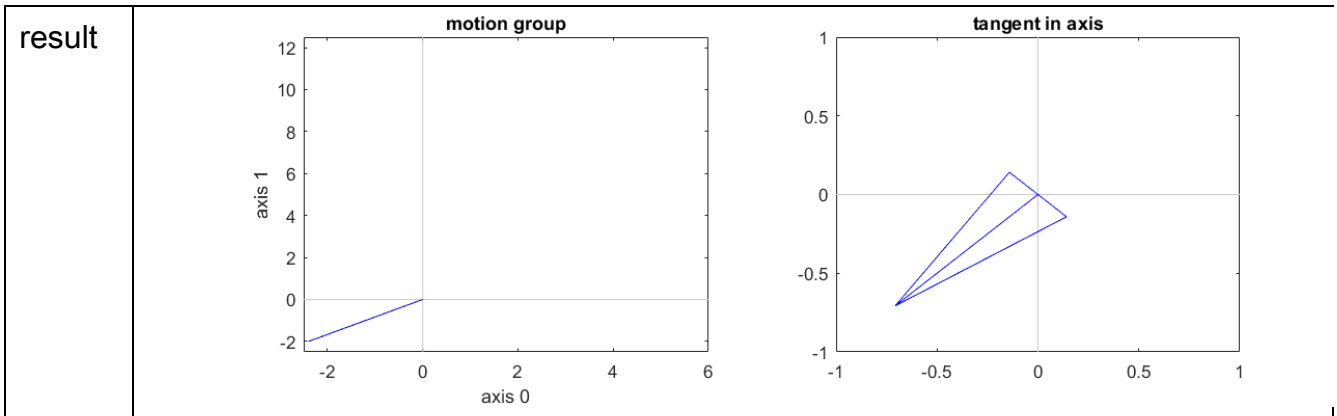
    ret = ECAT_McAxisTangentOut(DeviceNo, TangentInAxisNo, GroupNo);
    if (ret != 0) {
        printf("Axis tangent out failed:%d\n", ret);
        task_stop = true;
    } else
        task_index++;
    break;
case 1:
    ret = check_grp_state();
    if (ret == -1)
        task_stop = true;
    else if (ret == 0)
        task_stop = true;
    break;
default:
    task_stop = true;
    break;
}

    msleep(1);
}
return 0;
}

```

使用範例的切線跟隨移動軌跡:

cmd	GroupMoveLine to (0, 0)



<b>cmd</b>	<b>GroupMoveLine to (0, 10)</b>
<b>result</b>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>motion group</b></p> <p>axis 1</p> <p>axis 0</p> </div> <div style="text-align: center;"> <p><b>tangent in axis</b></p> </div> </div>

<b>cmd</b>	<b>AxisTangentInGroup with angle 0</b>
<b>result</b>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>motion group</b></p> <p>axis 1</p> <p>axis 0</p> </div> <div style="text-align: center;"> <p><b>tangent in axis</b></p> </div> </div>

cmd	GroupMoveCircular to angle -180	
result	<p style="text-align: center;"><b>motion group</b></p>	<p style="text-align: center;"><b>tangent in axis</b></p>
	<p style="text-align: center;"><b>motion group</b></p>	<p style="text-align: center;"><b>tangent in axis</b></p>
	<p style="text-align: center;"><b>motion group</b></p>	<p style="text-align: center;"><b>tangent in axis</b></p>
	<p style="text-align: center;"><b>motion group</b></p>	<p style="text-align: center;"><b>tangent in axis</b></p>





## 7.8.53. ECAT\_McAxisTangentOut

### 說明:

停止指定軸號切線跟隨運動控制。

### 格式:

int32\_t ECAT\_McAxisTangentOut(uint16\_t DeviceNo, uint16\_t AxisNo, uint16\_t GroupNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
AxisNo	uint16_t	IN	軸號
GroupNo	uint16_t	IN	群組編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

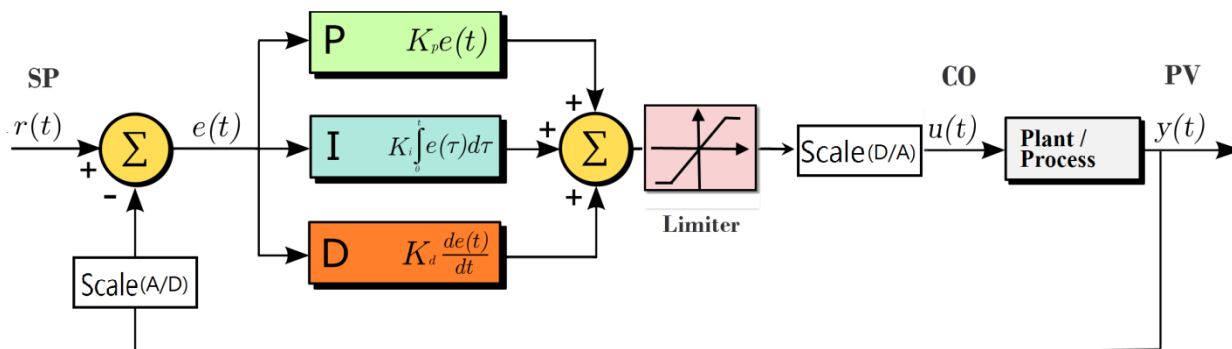
### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 2;
uint16_t GroupNo = 0;

ret = ECAT_McAxisTangentOut(DeviceNo, AxisNo, GroupNo);
if (ret != 0) {
    printf("Failed to tangent out:%d\n", ret);
}
```

---

## 7.9. PID 控制器



SP: SetPoint

CO: Controller Output

PV: Process Variable

e(t): SP-PV

Simulated Plant:

$$G(s) = \frac{1}{s+1}$$

Scale:

$$a \rightarrow \boxed{\text{Scale}} \rightarrow b$$

$$b = a * \text{ScaleGain} + \text{ScaleOffset}$$

## 7.9.1. ECAT\_PidGetSetPointValue

### 說明:

取得 PID 控制迴路的命令輸入設定值。

### 格式:

```
int32_t ECAT_PidGetSetPointValue(uint16_t DeviceNo, uint32_t PidNo, double *SetPointValue)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
SetPointValue	double *	OUT	PID 控制迴路的命令輸入設定值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidGetSetPointValue(DeviceNo, PidNo, &Setpoint);
if(ret != 0)
{
    printf("Failed to get Pid Set Point Value:%d\n",ret);
}
else
{
    printf("Pid Set Point Value %d\n", Setpoint);
}
```

## 7.9.2. ECAT\_PidSetSetPointValue

### 說明:

設定 PID 控制迴路的命令輸入設定值。

### 格式:

```
int32_t ECAT_PidSetSetPointValue(uint16_t DeviceNo, uint32_t PidNo, double SetPointValue)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
SetPointValue	double	IN	PID 控制迴路的命令輸入設定值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_SetPidStatus (DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)
{
    printf("Failed to set Pid Set Point Value:%d\n",ret);
}
```

### 7.9.3. ECAT\_PidGetProcessVariable

**說明:**

取得 PID 控制迴路的輸出測量值(處理過程變數)。

**格式:**

int32\_t ECAT\_PidGetProcessVariable(uint16\_t DeviceNo, uint32\_t PidNo, double \*ProcessVariable)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
ProcessVariable	double *	OUT	PID 控制迴路的處理過程變數(輸出測量值)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double ProcessVariable;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidGetProcessVariable(DeviceNo, PidNo, &ProcessVariable);
if(ret != 0)
{
    printf("Failed to get Pid Process Variable:%d\n",ret);
}
else
{
    printf("Pid Process Variable %d\n", ProcessVariable);
}
```

## 7.9.4. ECAT\_PidGetSampleTime

### 說明:

取得 PID 控制器執行的間隔時間，單位為通訊週期。

### 格式:

```
int32_t ECAT_PidGetSampleTime(uint16_t DeviceNo, uint32_t PidNo, uint32_t
*Interval)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
Interval	int32_t*	Output	間隔時間，單位為通訊週期

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
int32_t Interval = 0;

ret = ECAT_PidGetSampleTime(DeviceNo, PidNo, &Interval);
if(ret != 0)
{
    printf("Failed to Get Pid Controller:%d\n",ret);
}
else
{
    printf("Pid Interval %d\n", Interval);
}
```

---

## 7.9.5. ECAT\_PidSetSampleTime

### 說明:

設定 PID 控制器執行的間隔時間，單位為通訊週期。

### 格式:

```
int32_t ECAT_PidSetSampleTime(uint16_t DeviceNo, uint32_t PidNo, uint32_t Interval)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號(卡號)
PidNo	uint32_t	IN	PID 控制器編號
Interval	int32_t	IN	間隔時間，單位為通訊週期

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
int32_t Interval = 1;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0)
{
    printf("Failed to set Pid Controller:%d\n",ret);
}

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);
```

## 7.9.6. ECAT\_PidGetStatus

### 說明:

取得 PID 控制器是否啟用。

### 格式:

```
int32_t ECAT_PidGetStatus(uint16_t DeviceNo, uint32_t PidNo, uint8_t *status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號(卡號)
PidNo	uint32_t	IN	PID 控制器編號
status	uint8_t*	Output	狀態 0: 未啟用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint8_t Status= 0;

ret = ECAT_PidGetStatus(DeviceNo, PidNo, &Status);
if(ret != 0)
{
    printf("Failed to Get Pid Status:%d\n",ret);
}
else
{
    printf("Pid Status Value %d\n", Status);
}
```

---

## 7.9.7. ECAT\_PidSetStatus

### 說明:

設定 PID 控制器是否要啟用。注意，狀態由啟用變成未啟用時，PID 控制模組的輸出數值會保持最後輸出值而不會自動設為 0。如果要在未啟用時設定控制模組的輸出值，請呼叫 [ECAT\\_SetSlaveRxPdoData](#) 函式來設定。

### 格式:

```
int32_t ECAT_PidSetStatus(uint16_t DeviceNo, uint32_t PidNo, uint8_t status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
status	uint8_t	IN	狀態 0: 未啟用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)
{
    printf("Failed to Set Pid Status:%d\n",ret);
}
```

## 7.9.8. ECAT\_PidGetSimulateMode

### 說明:

取得 PID 控制迴路是否啟用模擬模型。當使用模擬模型時，AD/DA 模組並不會被使用來做控制。

### 格式:

```
int32_t ECAT_PidGetSimulateMode(uint16_t DeviceNo, uint32_t PidNo, uint8_t *status)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
status	uint8_t*	Output	狀態 0: 未啟用 1: 啟用

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint8_t Status= 0;

ret = ECAT_PidGetSimulateMode(DeviceNo, PidNo, &Status);
if(ret != 0)
{
    printf("Failed to Get Pid Simulate:%d\n",ret);
}
else
{
    printf("Pid Simulate %d\n", Status);
}
```

---

### 7.9.9. ECAT\_PidSetSimulateMode

#### 說明:

設定 PID 控制迴路是否啟用模擬模型來模擬受控系統。注意，當使用模擬模型時，AD/DA 模組並不會被使用來做控制，所以不會改變輸出模組的控制值。如果要設定輸出模組的值，請使用 [ECAT\\_SetSlaveRxPdoData](#) 函式。

#### 格式:

```
int32_t ECAT_PidSetSimulateMode(uint16_t DeviceNo, uint32_t PidNo, uint8_t status)
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
status	uint8_t	IN	狀態 0: 未啟用 1: 啟用

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)
{
    printf("Failed to Set Pid Simulate:%d\n",ret);
}

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);
```

---

## 7.9.10. ECAT\_PidGetParameter

### 說明:

取得 PID 控制器參數。

### 格式:

```
int32_t ECAT_PidGetParameter(uint16_t DeviceNo, uint32_t PidNo, double *kp, double *ki, double *kd)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
kp	double *	Output	PID 參數
ki	double *	Output	PID 參數
kd	double *	Output	PID 參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double kp= 0;
double ki= 0;
double kd= 0;
ret = ECAT_PidGetParameter(DeviceNo, PidNo, &kp, &ki, &kd)
if(ret != 0)
{
    printf("Failed to Get Pid Parameter:%d\n",ret);
}
else
{
    printf("Pid Parameter : kp:%f , ki:%f , kd:%f \n", kp, ki, kd);
}
```

---

## 7.9.11. ECAT\_PidSetParameter

### 說明:

設定 PID 控制器參數。

### 格式:

int32\_t ECAT\_PidSetParameter(uint16\_t DeviceNo, uint32\_t PidNo, double kp, double ki, double kd)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
kp	double	IN	PID 參數
ki	double	IN	PID 參數
kd	double	IN	PID 參數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)
{
    printf("Failed to Set Pid Parameter:%d\n",ret);
}

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n",ret);
```

## 7.9.12. ECAT\_PidGetProcessVariableModule

### 說明:

取得 PID 控制迴路量測模組之設定。使用 AD 模組量測到處理過程變數時，模組的 AD 通道只能提供一個帶符號整數數值（由 Bitlength 決定其範圍），令此數命名為 IN\_0。要轉成物理量 IN\_1 需要借助 ScaleGain 與 ScaleOffset 與以下的計算：

$$IN_1 = ScaleGain * IN_0 + ScaleOffset;$$

### 格式:

```
int32_t ECAT_PidGetProcessVariableModule(uint16_t DeviceNo, uint32_t PidNo,
uint16_t *SlaveNo, uint16_t *OffsetByte, uint16_t *Bitlength, double *ScaleGain, double
*ScaleOffset)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
SlaveNo	uint16*	Output	從站編號
OffsetByte	uint16*	Output	取得相對於從站模組 TxPdo Process Image 資料開始位置偏移量
Bitlength	uint16*	Output	設定的資料大小，單位:bit
ScaleGain	double *	Output	輸入增益
ScaleOffset	double *	Output	輸入偏移量

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;  
uint16_t DeviceNo = 0;  
uint32_t PidNo = 0;  
uint16_t SlaveNo = 0;  
uint16_t Offset = 0;  
uint16_t Bitsize = 16;  
double Scalegain = 1;  
double Scaleoffset = 0;
```

```
ret=ECAT_PidGetProcessVariableModule(DeviceNo, PidNo, &SlaveNo, &Offset, &Bitsize, &Scalegain,  
&Scaleoffset);  
if(ret != 0)  
{  
    printf("Failed to Get Pid Input:%d\n",ret);  
}
```

---

### 7.9.13. ECAT\_PidSetProcessVariableModule

#### 說明:

設定 PID 控制器量測模組之設定。使用 AD 模組量測到處理過程變數時，模組的 AD 通道只能提供一個帶符號整數數值（由 Bitlength 決定其範圍），令此數命名為 IN\_0。要轉成物理量 IN\_1 需要借助 ScaleGain 與 ScaleOffset 與以下的計算：

$$IN_1 = ScaleGain * IN_0 + ScaleOffset;$$

#### 格式:

int32\_t ECAT\_PidSetInput(uint16\_t DeviceNo, uint32\_t PidNo, uint16\_t SlaveNo, uint16\_t OffsetByte, uint16\_t Bitlength, double ScaleGain, double ScaleOffset)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號（卡號）
PidNo	uint32_t	IN	PID 控制器編號
SlaveNo	uint16	IN	從站編號
OffsetByte	uint16	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
Bitlength	uint16	IN	設定的資料大小，單位:bit
ScaleGain	double	IN	輸入增益
ScaleOffset	double	IN	輸入偏移量

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo_input = 0;
uint16_t Offset_input = 0;
uint16_t Bitsize_input = 16;
double Scalegain_input = 1;
double Scaleoffset_input = 0;
uint16_t SlaveNo_output = 0;
uint16_t Offset_output = 2;
uint16_t Bitsize_output = 16;
double Scalegain_output = 1;
double Scaleoffset_output = 0;
double Max_Value = 0;
double Min_Value = 0;
double Setpoint = 5;
uint8_t Simulate= 0;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;

ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret=ECAT_PidSetProcessVariableModule(DeviceNo, PidNo, SlaveNo_input, Offset_input, Bitsize_input,
Scalegain_input, Scaleoffset_input);
if(ret != 0)
```

---

```
{  
    printf("Failed to Set Pid Input:%d\n",ret);  
}  
  
ret= ECAT_PidSetControlOutputModule(DeviceNo, PidNo, SlaveNo_output, Offset_output, Bitsize_output,  
Scalegain_output, Scaleoffset_output, Max_Value, Min_Value);  
if(ret != 0) printf("Failed to Set Pid Output:%d\n",ret);  
  
ret = ECAT_SetPidStatus (DeviceNo, PidNo, enable);  
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);
```

---

### 7.9.14. ECAT\_PidGetControlOutputModule

**說明:**

取得 PID 控制器輸出模組之設定。當 PID 控制器算出一個控制量，此控制量經過一個限制器來限制其輸出範圍 (由 **Output\_Max\_Value** 與 **Output\_Min\_Value** 決定), 令此值為 **OUT\_0**。此輸出值要使用 DA 模組做控制輸出，但 DA 通道只能接受一個帶符號整數數值 (由 **Bitlength** 決定其範圍)，令此數命名為 **OUT\_1**。要將物理量轉成 **OUT\_1** 需要借助 **ScaleGain** 與 **ScaleOffset** 與以下的計算:

$$OUT_1 = ScaleGain * OUT_0 + ScaleOffset;$$

假設 **OUT\_1** 超出 **Bitlength** 所定義的有號整數範圍，那麼 **OUT\_1** 要依據此範圍做飽和處理。例如，**Bitlength** = 16，則比 32767 大時，輸出 32767；比 -32768 還小時，輸出 -32768。

$$\text{if } (OUT_1 > 32767) \quad OUT_1 = 32767;$$

$$\text{if } (OUT_1 < - 32768) \quad OUT_1 = -32768;$$

**格式:**

`int32_t ECAT_PidGetControlOutputModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t *SlaveNo, uint16_t *OffsetByte, uint16_t *Bitlength, double *ScaleGain, double *ScaleOffset, double *Output_Max_Value, double *Output_Min_Value)`

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
PidNo	uint32_t	IN	PID 控制器編號
SlaveNo	uint16*	Output	從站編號
OffsetByte	uint16*	Output	取得相對於從站模組 RxPdo Process Image 資料開始位置偏移量
Bitlength	uint16*	Output	設定的資料大小，單位:bit
ScaleGain	double *	Output	輸出增益
ScaleOffset	double *	Output	輸出位移量
Output_Max_Value	double *	Output	輸出最大值
Output_Min_Value	double *	Output	輸出最小值

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo = 0;
uint16_t Offset = 0;
uint16_t Bitsize = 16;
double Scalegain = 1;
double Scaleoffset = 0;
double Max_Value = 0;
double Min_Value = 0;

ret=ECAT_PidGetControlOutputModule(DeviceNo, PidNo, &SlaveNo, &Offset, &Bitsize, &Scalegain,
&Scaleoffset, &Max_Value, &Min_Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output:%d\n",ret);
}
```

---



## 7.9.15. ECAT\_PidSetControlOutputModule

### 說明:

設定 PID 控制器輸出模組之設定。當 PID 控制器算出一個控制量，此控制量經過一個限制器來限制其輸出範圍（由 **Output\_Max\_Value** 與 **Output\_Min\_Value** 決定），令此值為 **OUT\_0**。此輸出值要使用 DA 模組做控制輸出，但 DA 通道只能接受一個帶符號整數數值（由 **Bitlength** 決定其範圍），令此數命名為 **OUT\_1**。要將物理量轉成 **OUT\_1** 需要借助 **ScaleGain** 與 **ScaleOffset** 與以下的計算：

$$OUT_1 = ScaleGain * OUT_0 + ScaleOffset;$$

假設 **OUT\_1** 超出 **Bitlength** 所定義的有號整數範圍，那麼 **OUT\_1** 要依據此範圍做飽和處理。例如，**Bitlength** = 16，則比 32767 大時，輸出 32767；比 -32768 還小時，輸出 -32768。

$$\text{if } (OUT_1 > 32767) \quad OUT_1 = 32767;$$

$$\text{if } (OUT_1 < -32768) \quad OUT_1 = -32768;$$

### 格式:

`int32_t ECAT_PidSetControlOutputModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t Bitlength, double ScaleGain, double ScaleOffset, double Output_Max_Value, double Output_Min_Value)`

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
SlaveNo	uint16	IN	從站編號
OffsetByte	uint16	IN	設定相對於從站模組 RxPdo Process Image 資料開始位置偏移量
Bitlength	uint16	IN	設定的資料大小，單位:bit
ScaleGain	double	IN	輸出增益
ScaleOffset	double	IN	輸出位移量
Output_Max_Value	double	IN	輸出最大值
Output_Min_Value	double	IN	輸出最小值

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```

int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo_input = 0;
uint16_t Offset_input = 0;
uint16_t Bitsize_input = 16;
double Scalegain_input = 1;
double Scaleoffset_input = 0;
uint16_t SlaveNo_output = 0;
uint16_t Offset_output = 2;
uint16_t Bitsize_output = 16;
double Scalegain_output = 1;
double Scaleoffset_output = 0;
double Max_Value = 0;
double Min_Value = 0;
double Setpoint = 5;
uint8_t Simulate= 0;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;

ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

```

```
ret=ECAT_PidSetProcessVariableModule(DeviceNo, PidNo, SlaveNo_input, Offset_input, Bitsize_input,  
Scalegain_input, Scaleoffset_input);
```

```
if(ret != 0)printf("Failed to Set Pid Input:%d\n",ret);
```

```
ret=ECAT_PidSetControlOutputModule(DeviceNo, PidNo, SlaveNo_output, Offset_output, Bitsize_output,  
Scalegain_output, Scaleoffset_output, Max_Value, Min_Value);
```

```
if(ret != 0)
```

```
{
```

```
    printf("Failed to Set Pid Output:%d\n",ret);
```

```
}
```

```
ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
```

```
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);
```

---

## 7.9.16. ECAT\_PidGetControlOutputValue

### 說明:

取得 PID 控制器計算的控制輸出數值。

### 格式:

int32\_t ECAT\_PidGetControlOutputValue(uint16\_t DeviceNo, uint32\_t PidNo, double \*Output)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
Output	double*	Output	PID 控制器控制輸出數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Value= 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidGetControlOutputValue(DeviceNo, PidNo, &Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output Value:%d\n",ret);
}
else
{
    printf("Pid OutputValue :%f \n", Value);
}

```



## 7.9.17. ECAT\_PidGetSimulateFeedback

### 說明:

取得 PID 控制迴路之模擬模型的輸出數值。

### 格式:

```
int32_t ECAT_PidGetSimulateFeedback(uint16_t DeviceNo, uint32_t PidNo, double
*Feedback)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
Feedback	double*	Output	PID 控制迴路之模擬模型的輸出數值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Feedback = 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidGetSimulateFeedback (DeviceNo, PidNo, &Feedback);
if(ret != 0)
{
    printf("Failed to Get Pid Simulate Feedback:%d\n",ret);
}
else
{
    printf("Pid Simulate Feedback:%f \n", Feedback);
}
```





## 7.9.18. ECAT\_PidGet\_Sp\_Err\_Op\_Pv

### 說明:

同時取得 PID 控制迴路命令設定值、誤差值、控制輸出值以及處理過程變數量測值。

### 格式:

```
int32_t ECAT_PidGet_Sp_Err_Op_Pv(uint16_t DeviceNo, uint32_t PidNo, double *SetPointValue, double *Error, double *OutputValue, double *ProcessVariable)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
PidNo	uint32_t	IN	PID 控制器編號
SetPointValue	double*	Output	PID 控制迴路命令設定值
Error	double*	Output	PID 控制迴路命令設定值與回授測量值之差
OutputValue	double*	Output	PID 控制器計算之控制輸出數值
ProcessVariable	double*	Output	PID 控制迴路處理過程變數量測值

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Error= 0;
double ProcessVariable= 0;
double OutputValue= 0;
double Setpoint = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, Setpoint);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidGet_Sp_Err_Op_Pv(DeviceNo, PidNo, &Setpoint, &Error, &OutputValue, &ProcessVariable);
if(ret != 0)
{
    printf("Failed to Get Pid Sp_Err_Op_Pv:%d\n",ret);
}
else
{

```

---

```
printf("Pid Set Point Value :%f \n", Setpoint);  
printf("Pid Error :%f \n", Error);  
printf("Pid OutputValue :%f \n", OutputValue);  
printf("Pid ProcessVariable:%f \n", ProcessVariable);  
}
```

---

## 7.10. 史都華平台

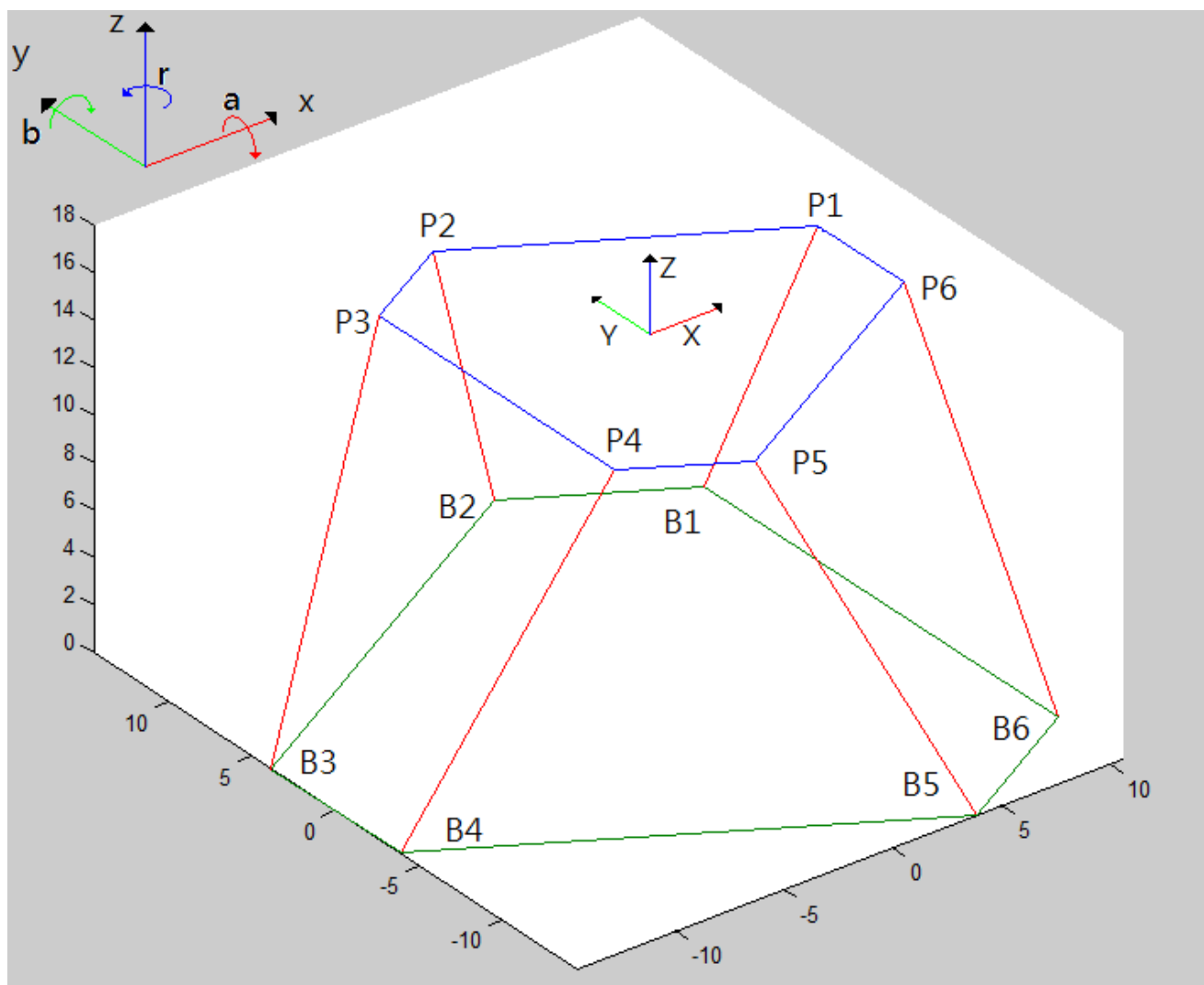
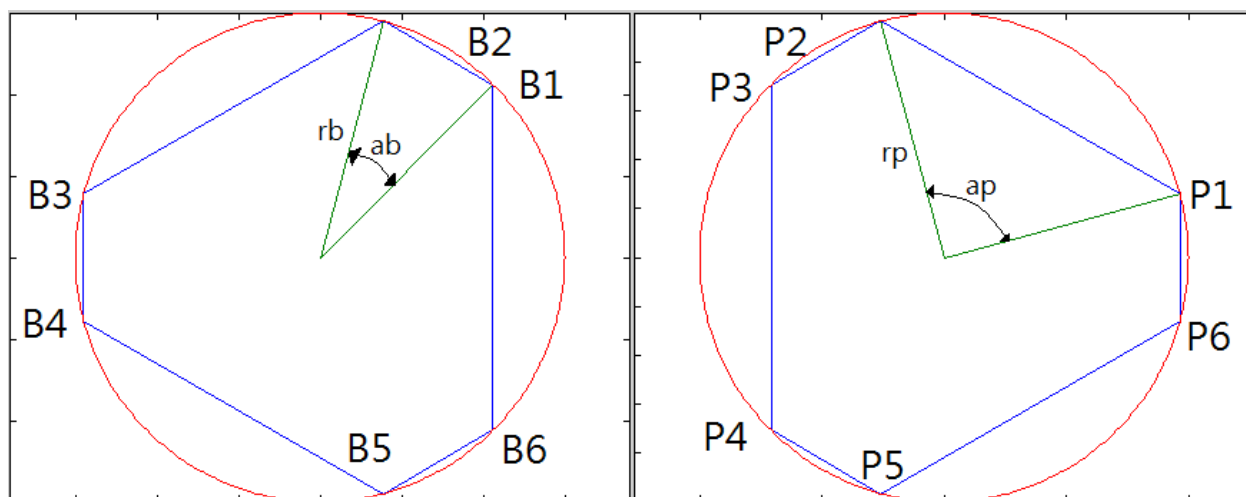
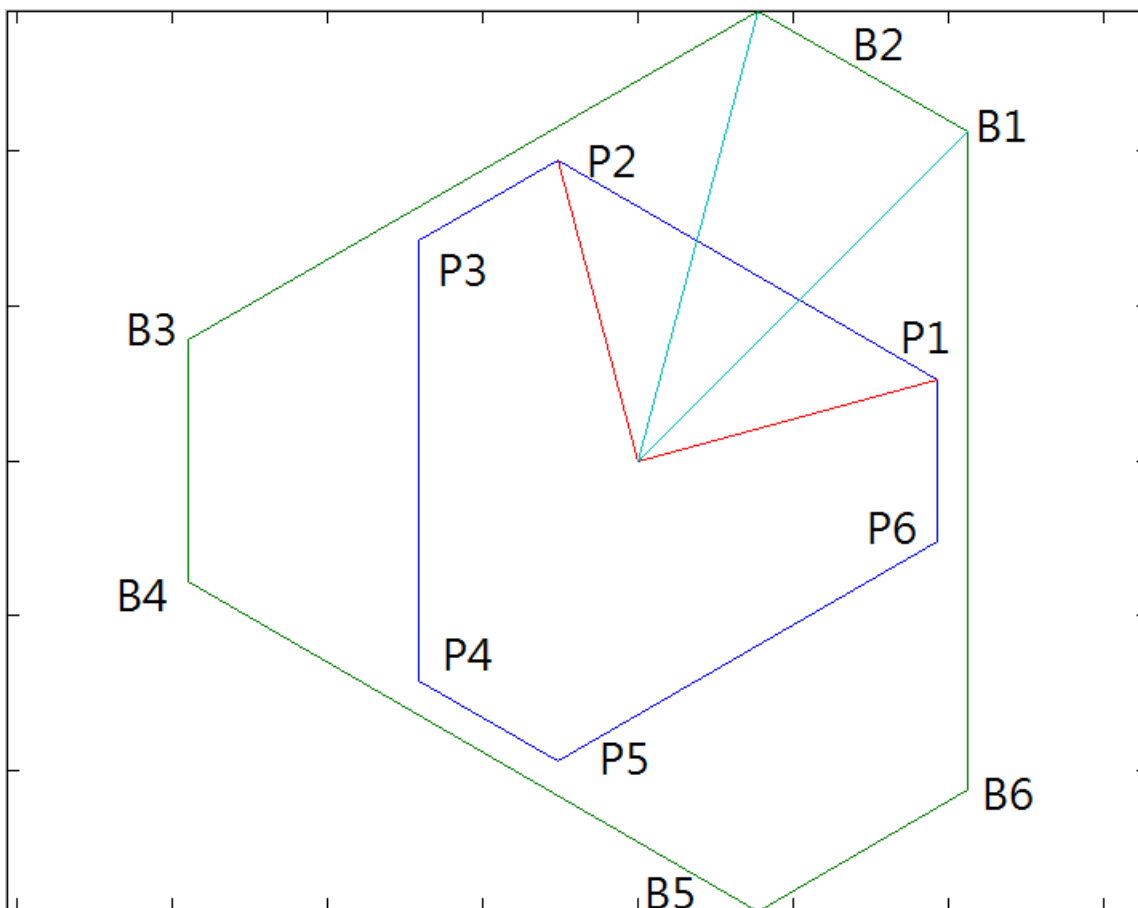


圖 7.2

上平台:6 個節點所形成的平面，P1~P6

下平台:6 個節點所形成的平面，B1~B6



rb:下平台之半徑

ab:下平板 B1、B2 與中心點的夾角

rp:上平台之半徑

ap:上平板 P1、P2 與中心點的夾角

### 7.10.1. ECAT\_McSetStewartPlatform\_M1

**說明:**

設定史都華平台之幾何參數(方法 1)。

**格式:**

int32\_t ECAT\_McSetStewartPlatform\_M1(uint16\_t DeviceNo, double radiusB, double angleB, double radiusP, double angleP, double RodLength, double Max\_RodLength)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
radiusB	double	IN	下平板之半徑，單位:公厘
angleB	double	IN	下平板 B1、B2 與中心點的夾角， 單位: 角度
radiusP	double	IN	上平板之半徑，單位:公厘
angleP	double	IN	上平板 P1、P2 與中心點的夾角， 單位: 角度
RodLength	double	IN	收縮時的桿長，單位:公厘
Max_RodLength	double	IN	伸到最長時的桿長，單位:公厘

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

**使用範例:****[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
double radiusB = 15;
double angleB = 30;
double radiusP = 10;
double angleP = 90;
double RodLength = 15;
double Max_RodLength = 30;

ret = ECAT_McSetStewartPlatform_M1(DeviceNo, radiusB, angleB, radiusP, angleP, RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n",ret);
    return;
}
```

---



## 7.10.2. ECAT\_McGetStewartPlatform\_M1

### 說明:

取得史都華平台之幾何參數(方法 1)。

### 格式:

```
int32_t ECAT_McGetStewartPlatform_M1(uint16_t DeviceNo, double *radiusB, double *angleB, double *radiusP, double *angleP, double *RodLength, double *Max_RodLength)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
radiusB	double*	OUT	下平板之半徑，單位:公厘
angleB	double*	OUT	下平板 B1、B2 與中心點的夾角，單位: 角度
radiusP	double*	OUT	上平板之半徑，單位:公厘
angleP	double*	OUT	上平板 P1、P2 與中心點的夾角，單位: 角度
RodLength	double*	OUT	收縮時的桿長，單位:公厘
Max_RodLength	double*	OUT	伸到最長時的桿長，單位:公厘

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
double radiusB = 0;
double angleB = 0;
double radiusP = 0;
double angleP = 0;
double RodLength = 0;
double Max_RodLength = 0;

ret = ECAT_McGetStewartPlatform_M1(DeviceNo, &radiusB, &angleB, &radiusP, &angleP, &RodLength,
&Max_RodLength);
if(ret < 0)
{
    printf("Failed to Get Stewart Platform:%d\n",ret);
    return;
}
```

---

### 7.10.3. ECAT\_McSetStewartPlatform\_M2

#### 說明:

設定史都華平台之幾何參數(方法 2)。

#### 格式:

```
int32_t ECAT_McSetStewartPlatform_M2(uint16_t DeviceNo, double Bx[], double By[],
double Px[], double Py[], double Z0, double RodLength[], double Max_RodLength[])
```

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Bx	double[]	IN	下平板之 X 軸座標，共六點，單位:公厘
By	double[]	IN	下平板之 Y 軸座標，共六點，單位:公厘
Px	double[]	IN	上平板之 X 軸座標，共六點，單位:公厘
Py	double[]	IN	上平板之 Y 軸座標，共六點，單位:公厘
Z0	double	IN	上平板相對於下平板的垂直高度， 單位: 公厘
RodLength	double[]	IN	收縮時的桿長，單位: 公厘
Max_RodLength	double[]	IN	伸到最長時的桿長，單位: 公厘

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
double Bx[6]= {10.6066, 3.8823, -14.4889, -14.4889, 3.8823, 10.6066};
double By[6]= {10.6066, 14.4889, 3.8823, -3.8823, -14.4889, -10.6066};
double Px[6] = {9.6593, -2.5882, -7.0711, -7.0711, -2.5882, 9.6593};
double Py[6] = {2.5882, 9.6593, 7.0711, -7.0711, -9.6593, -2.5882};
double Z0 = 14.1421;
double RodLength[6] = {15, 15, 15, 15, 15, 15};
double Max_RodLength[6] = {30, 30, 30, 30, 30, 30};

ret = ECAT_McSetStewartPlatform_M2(DeviceNo, Bx, By, Px, Py, Z0, RodLength, Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n",ret);
    return;
}
```

## 7.10.4. ECAT\_McGetStewartPlatform\_M2

### 說明:

取得史都華平台之幾何參數(方法 2)。

### 格式:

```
int32_t ECAT_McGetStewartPlatform_M2(uint16_t DeviceNo, double *Bx, double *By,
double *Px, double *Py, double *Z0, double *RodLength, double *Max_RodLength)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Bx	double*	OUT	下平板之 X 軸座標，共六點，單位:公厘
By	double*	OUT	下平板之 Y 軸座標，共六點，單位:公厘
Px	double*	OUT	上平板之 X 軸座標，共六點，單位:公厘
Py	double*	OUT	上平板之 Y 軸座標，共六點，單位:公厘
Z0	double*	OUT	上平板相對於下平板的垂直高度，單位:公厘
RodLength	double*	OUT	收縮時的桿長，單位:公厘
Max_RodLength	double*	OUT	伸到最長時的桿長，單位:公厘

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
double Bx[6]= {0};
double By[6]= {0};
double Px[6] = {0};
double Py[6] = {0};
double Z0 = 0;
double RodLength[6] = {0};
double Max_RodLength[6] = {0};

ret = ECAT_McGetStewartPlatform_M2(DeviceNo, &Bx, &By, &Px, &Py, &Z0, &RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Get Stewart Platform:%d\n",ret);
    return;
}
```

### 7.10.5. ECAT\_McStewartPlatformMoveAbs\_PT

#### 說明:

藉由輸入史都華平台的姿態和時間，開始指定群組絕對位置線性補間移動。

注意:線性補間移動是指馬達軸，而不是史都華平台在空間上直線運動。

#### 格式:

int32\_t ECAT\_McStewartPlatformMoveAbs\_PT(uint16\_t DeviceNo, uint16\_t GroupNo, double Pose[], double \*Pos, double Time)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
GroupNo	uint16_t	IN	群組編號
Pose	double[]	IN	史都華平台姿態 x: 沿 x 軸方向的位移，單位:公厘 y: 沿 y 軸方向的位移，單位:公厘 z: 沿 z 軸方向的位移，單位:公厘 a: 繞 x 軸的旋轉，單位:角度 b: 繞 y 軸的旋轉，單位:角度 r: 繞 z 軸的旋轉，單位:角度 位移方向及旋轉方向請看圖 7.2
Pos	double*	OUT	群組內各軸的目標位置(單位:user unit)
Time	double	IN	移動時間(單位:second)

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t i;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BLENDING; //0: Aborting, 1: Buffered, 2: Blending
double StewartPlatformPose[6]; //x y z a b r
double Pos[6]; //position of axis0~axis5
double GroupTime;
double radiusB = 15;
double angleB = 30;
double radiusP = 10;
double angleP = 90;
double RodLength = 15;
double Max_RodLength = 30;

ret = ECAT_McSetStewartPlatform_M1(DeviceNo, radiusB, angleB, radiusP, angleP, RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n",ret);
    return;
}

for(i=0;i<6;i++) //6-axis Stewart Platform
{
    ret = ECAT_McAddAxisToGroup(DeviceNo,GroupNo,i);
    if(ret < 0)
    {
        printf("Failed to add axis to group:%d\n",ret);
        return;
    }
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);

```



```
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    StewartPlatformPose [0] = 0; // x
    StewartPlatformPose [1] = 0; // y
    StewartPlatformPose [2] = 1; // z
    StewartPlatformPose [3] = 0; // a
    StewartPlatformPose [4] = 0; // b
    StewartPlatformPose [5] = 0; // r
    GroupTime = 1;
    ret = ECAT_McStewartPlatformMoveAbs_PT(DeviceNo,GroupNo, StewartPlatformPose, &Pos,
GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //Error Stop
    {
        printf("Group error stop\n");
    }
}
}
```



## 7.11. 運動記錄

### 7.11.1. ECAT\_McSetMotionRecord

**說明:**

設定是否記錄軸的真實位置、速度或是命令位置、速度，最多記錄 10 萬筆資料，每個通訊週期記錄乙次。

注意:此函式不會將記錄筆數設為零，如需將記錄筆數清零，則請使用函式 [ECAT\\_McClearMotionRecord](#)。

**格式:**

```
int32_t ECAT_McSetMotionRecord(uint16_t DeviceNo, uint16_t state)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
state	uint16_t	IN	設定 1 開始記錄，設定 0 停止記錄

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_McSetMotionRecord(DeviceNo,1);
if(ret < 0)
{
    printf("Failed to Set Motion Record:%d\n",ret);
}
else
{
    printf("Set Motion Record successfully! \n");
}
```

---

## 7.11.2. ECAT\_McGetMotionRecordState

### 說明:

讀取記錄狀態

### 格式:

```
int32_t ECAT_McGetMotionRecordState(uint16_t DeviceNo, uint16_t *state, uint32_t *count)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
state	uint16_t*	OUT	1 為記錄中，0 為暫停記錄中
count	uint32_t*	OUT	已記錄筆數

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t state;
uint32_t cnt;
ret = ECAT_McGetMotionRecordState(DeviceNo,&state,&cnt);
if(ret < 0)
{
    printf("Failed to Get Motion Record State:%d\n",ret);
}
else
{
    printf("State:%u , Count:%u \n", state, cnt);
}
```

---

### 7.11.3. ECAT\_McClearMotionRecord

**說明:**

將記錄筆數清零。

**格式:**

int32\_t ECAT\_McClearMotionRecord(uint16\_t DeviceNo)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
ret = ECAT_McClearMotionRecord(DeviceNo);  
if(ret < 0)  
{  
    printf("Failed to Clear Motion Record:%d\n",ret);  
}
```

---



### 7.11.4. ECAT\_McSetMotionRecordParam

**說明:**

設定運動記錄所要儲存的參數。位置、速度或是命令位置、速度，可任選兩種。

**格式:**

```
int32_t ECAT_McSetMotionRecordParam(uint16_t DeviceNo, uint16_t Value1,
uint16_t Value2)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
Value1	uint16_t	IN	運動記錄儲存參數編號(如表 7.15 所示)
Value2	uint16_t	IN	運動記錄儲存參數編號(如表 7.15 所示)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.15: 運動記錄儲存參數

標頭檔定義名稱	數值	說明
MC_RECORD_POSITION	0	軸當前位置(單位:user unit)
MC_RECORD_VELOCITY	1	軸當前速度(單位:user unit)
MC_RECORD_COMMAND_POSITION	2	軸命令位置(單位:user unit)
MC_RECORD_COMMAND_VELOCITY	3	軸命令速度(單位:user unit)

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t Value1= MC_RECORD_POSITION;
uint16_t Value2= MC_RECORD_VELOCITY;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Value1, Value2);
if(ret < 0)
{
    printf("Failed to set motion record parameters:%d\n",ret);
}
ret = ECAT_McSetMotionRecord(DeviceNo,1);
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n",ret);

```

### 7.11.5. ECAT\_McGetMotionRecordParam

**說明:**

取得運動記錄所要儲存的參數。

**格式:**

```
int32_t ECAT_McGetMotionRecordParam(uint16_t DeviceNo, uint16_t *Value1,
uint16_t *Value2)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號
Value1	uint16_t*	OUT	運動紀錄儲存參數編號(如表 7.15 所示)
Value2	uint16_t*	OUT	運動紀錄儲存參數編號(如表 7.15 所示)

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t *Value1;
uint16_t *Value2;

ret = ECAT_McGetMotionRecordParam(DeviceNo, &Value1, &Value2);
if(ret < 0)
{
    printf("Failed to get motion record parameters:%d\n",ret);
}
else
{
    printf("Value1:%u , Value2:%u \n", Value1, Value2);
}
```

---

## 7.11.6. ECAT\_McGetMotionRecordValue\_Ex

### 說明:

取代舊的函式"ECAT\_McGetMotionRecordValue"

取得指定記錄筆數及軸號之儲存數值。

注意:將 AxisNo 設為 65535 時，可以一次取得所有軸的儲存數值。

### 格式:

```
int32_t ECAT_McGetMotionRecordValue_Ex(uint16_t DeviceNo, uint32_t CountNo,
uint16_t AxisNo, float *Value1, float *Value2)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CountNo	uint32_t	IN	記錄筆數(索引號)
AxisNo	uint16_t	IN	軸號
Value1	float*	OUT	指定紀錄筆數及軸號之儲存數值(單位:user unit)
Value2	float*	OUT	指定紀錄筆數及軸號之儲存數值(單位:user unit)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 0;
uint16_t state;
uint32_t cnt;
int i;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1;
float Value2;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n",ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n",ret);

Sleep(1000); //wait for record something...

ret = ECAT_McSetMotionRecord(DeviceNo,0); // stop record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n",ret);

ret = ECAT_McGetMotionRecordState(DeviceNo,&state,&cnt); // get record count
if(ret < 0)
    printf("Failed to Get Motion Record State:%d\n",ret);
for(i=0;i< cnt;i++)
{
    ret = ECAT_McGetMotionRecordValue_Ex(DeviceNo, i , AxisNo, &Value1, &Value2);
    if(ret < 0)
    {
        printf("Failed to get motion record value:%d\n",ret);
    }
}

```

```

else
{
    printf("Axis Value1:%f , Value2:%f \n", Value1, Value2);
}
}
ret = ECAT_McClearMotionRecord(DeviceNo);
if(ret < 0)
    printf("Failed to Clear Motion Record:%d\n",ret);

}

```

### 使用範例:

#### [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 65535;
uint16_t state;
uint32_t cnt;
int i;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1[MC_AXIS_NO_MAX];
float Value2[MC_AXIS_NO_MAX];

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n",ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n",ret);

Sleep(1000); //wait for record something...

ret = ECAT_McSetMotionRecord(DeviceNo,0); // stop record
if(ret < 0)

```

---

```
printf("Failed to Set Motion Record:%d\n",ret);

ret = ECAT_McGetMotionRecordState(DeviceNo,&state,&cnt); // get record count
if(ret < 0)
    printf("Failed to Get Motion Record State:%d\n",ret);
for( i=0; i< cnt; i++)
{
    ret = ECAT_McGetMotionRecordValue_Ex(DeviceNo, i , AxisNo, Value1, Value2);
    if(ret < 0)
    {
        printf("Failed to get motion record value:%d\n",ret);
    }
    else
    {
        for( j=0; j< MC_AXIS_NO_MAX; j++)
        {
            printf("Axis Value1:%f , Value2:%f \n", Value1[ j ] , Value2[ j ]);
        }
    }
}
ret = ECAT_McClearMotionRecord(DeviceNo);
if(ret < 0)
    printf("Failed to Clear Motion Record:%d\n",ret);

}
```

---



### 7.11.7. ECAT\_McGetMotionRecordValueEx\_Ex

**說明:**

取代舊的函式"ECAT\_McGetMotionRecordValueEx"

取得指定記錄筆數及軸號之儲存數值。

**格式:**

int32\_t ECAT\_McGetMotionRecordValueEx\_Ex(uint16\_t DeviceNo, uint32\_t CountNo, uint16\_t Count, uint16\_t AxisNo, float \*Value1, float \*Value2, uint16\_t \*ActualCount)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
CountNo	uint32_t	IN	記錄筆數(索引號)
Count	uint16_t	IN	筆數，一次最多讀取 64 筆
AxisNo	uint16_t	IN	軸號
Value1	float*	OUT	指定紀錄筆數及軸號之儲存數值(單位:user unit)
Value2	float*	OUT	指定紀錄筆數及軸號之儲存數值(單位:user unit)
ActualCount	uint16_t*	OUT	實際讀取到的筆數

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 0;
uint16_t state;
uint32_t cnt;
int i,j;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1[RECORDDATA_GET_COUNT_MAX];
float Value2[RECORDDATA_GET_COUNT_MAX];
uint16_t *ActualGetCount;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n",ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n",ret);

Sleep(1000); //wait for record something...

ret = ECAT_McSetMotionRecord(DeviceNo,0); // stop record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n",ret);

ret = ECAT_McGetMotionRecordState(DeviceNo,&state,&cnt); // get record count
if(ret < 0)
    printf("Failed to Get Motion Record State:%d\n",ret);
for(i=0;i< cnt;i+= RECORDDATA_GET_COUNT_MAX)
{
    ret = ECAT_McGetMotionRecordValueEx_Ex(DeviceNo, i , RECORDDATA_GET_COUNT_MAX, AxisNo, Value1,
Value2, &ActualGetCount);
    if(ret < 0)
    {

```

---

```
        printf("Failed to get motion record value:%d\n",ret);
    }
    else
    {
        for(j=0; j< ActualGetCount; j++)
        {
            printf("Axis Value1:%f , Value2:%f \n", Value1[ j], Value2[ j]);
        }
    }
}
ret = ECAT_McClearMotionRecord(DeviceNo);
if(ret < 0)
    printf("Failed to Clear Motion Record:%d\n",ret);

}
```

---

## 7.12. 事件觸發

在 PC 上要檢查控制卡內的某些狀態變化的方式是將資料讀回來再檢查。但這樣的方式比較浪費 PC 的 CPU 運算時間。為了減輕 PC 負擔，同時讓系統的反應加快，在控制卡裡面有提供對 PC 發出事件 (event) 的方法。程式裡要事先設定要觸發事件的條件，然後就可以讓程式進入等待狀態。在等待的過程中，該程式 (或線程，thread) 是不會占用 PC 的 CPU 資源，系統於事件發生後就會馬上讓等待的程式醒來處理。

觸發事件的條件目前有位置的比較、單一 DI 變化、多個 DI 變化，以及運動狀態檢查。最多可以有 32 個觸發事件可以設定。基本上，觸發條件成立後，會自動禁用 (disabled)。如果還要再次使用該事件觸發，則必須在處理的程式裡再次設定為啟用 (enabled)。

ECAT\_SetTimer() 其實是一種時間事件觸發，但是與一般事件觸發用法有些不同: Timer 事件一旦啟用就會持續週期觸發，不用再重設一次啟用。而這裡的事件每次都必須再被設定啟用才能連續使用。

程式裡可以使用一個 event，當該事件成立後去設定或啟用另一個性質不同的 event。這樣可以讓系統執行一連串複雜的動作。

## 7.12.1. ECAT\_EvEnableEvent

### 說明:

啟用事件。

注:所有事件被觸發後會回到禁用狀態。

### 格式:

```
int32_t ECAT_EvEnableEvent(uint16_t DeviceNo, uint16_t EventID)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
{
    printf("Failed to enable event:%d\n",ret);
}
else
{
    printf("Enable event successfully!\n");
}
```

---

## 7.12.2. ECAT\_EvDisableEvent

### 說明:

禁用事件。

### 格式:

```
int32_t ECAT_EvDisableEvent(uint16_t DeviceNo, uint16_t EventID)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
ret = ECAT_EvDisableEvent(DeviceNo, EventID);
if(ret < 0)
{
    printf("Failed to disable event:%d\n",ret);
}
else
{
    printf("Disable event successfully!\n");
}
```

---



### 7.12.3. ECAT\_WaitforEvent

**說明:**

等待事件觸發，為一阻塞函式。

**格式:**

```
int32_t ECAT_WaitforEvent(uint16_t DeviceNo, uint32_t TimeOut, uint32_t
*TriggeredEvent)
```

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
TimeOut	uint32_t	IN	超時，單位:ms
TriggeredEvent	uint32_t*	IN	觸發的事件 注:可能有多個事件同時觸發

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID0 = 0;
uint16_t EventID1 = 1;
uint32_t Value= 0;
ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID0))) & (0x01)) == 1) //EventID 0 triggered
    {
        // do something...
    }
    if(((Value>>(int(EventID1))) & (0x01)) == 1) //EventID 1 triggered
    {
        // do something...
    }
}
}
```

## 7.12.4. ECAT\_AbortWaitforEvent

### 說明:

使用此函式可以強制 ECAT\_WaitforEvent 直接返回 0。

### 格式:

```
int32_t ECAT_AbortWaitforEvent(uint16_t DeviceNo)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

### [C/C++]

```
int32_t ret;  
uint16_t DeviceNo = 0;  
ret = ECAT_AbortWaitforEvent(DeviceNo);  
if(ret != 0)  
{  
    printf("Failed to abort wait event:%d\n",ret);  
}
```

---

### 7.12.5. ECAT\_EvSetComparePositionParameters

**說明:**

設定事件，實際位置比較觸發事件。

**格式:**

int32\_t ECAT\_EvSetComparePositionParameters(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t AxisNo, uint16\_t Operator, double ComparePosition)

**參數:**

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
AxisNo	uint16_t	IN	軸號
Operator	uint16_t	IN	運算子編號(如表 7.16 所示)
ComparePosition	double	IN	比較的位置

**回傳值:**

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

表 7.16: 運算子編號定義

標頭檔定義名稱	數值	說明
GREATER_THAN	0	位置大於比較數值時觸發事件
GREATER_THAN_OR_EQUAL_TO	1	位置大於等於比較數值時觸發事件
LESS_THAN	2	位置小於比較數值時觸發事件
LESS_THAN_OR_EQUAL_TO	3	位置小於等於比較數值時觸發事件

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint16_t Operator = GREATER_THAN;
double ComparePosition = 100;
uint32_t Value;

ret = ECAT_EvSetComparePositionParameters(DeviceNo, EventID, AxisNo, Operator, ComparePosition);
if(ret < 0)
{
    printf("Failed to set compare position parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

```

## 7.12.6. ECAT\_EvSetCompareCmdPositionParameters

### 說明:

設定事件，命令位置比較觸發事件。

### 格式:

int32\_t ECAT\_EvSetCompareCmdPositionParameters(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t AxisNo, uint16\_t Operator, double ComparePosition)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
AxisNo	uint16_t	IN	軸號
Operator	uint16_t	IN	運算子編號
ComparePosition	double	IN	比較的位置

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

### 運算子編號定義

標頭檔定義名稱	數值	說明
GREATER_THAN	0	位置大於比較位置時觸發事件
GREATER_THAN_OR_EQUAL_TO	1	位置大於等於比較位置時觸發事件
LESS_THAN	2	位置小於比較位置時觸發事件
LESS_THAN_OR_EQUAL_TO	3	位置小於等於比較位置時觸發事件

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint16_t Operator = GREATER_THAN;
double ComparePosition = 100;
uint32_t Value;

ret = ECAT_EvSetCompareCmdPositionParameters(DeviceNo, EventID, AxisNo, Operator,
ComparePosition);
if(ret < 0)
{
    printf("Failed to set compare command position parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

```



## 7.12.7. ECAT\_EvSetCompareDIBitParameters

### 說明:

設定事件，DI bit 比較觸發事件。

### 格式:

```
int32_t ECAT_EvSetCompareDIBitParameters(uint16_t DeviceNo, uint16_t EventID,
uint16_t SlaveNo, uint16_t BitNo, uint32_t CompareValue)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
SlaveNo	uint16_t	IN	從站編號
BitNo	uint16_t	IN	指定位元
CompareValue	uint32_t	IN	比較的數值 0:當指定的 DI Bit 數值由 1 變為 0 時觸發 1:當指定的 DI Bit 數值由 0 變為 1 時觸發 2:當指定的 DI Bit 數值由 1 變為 0 時或由 0 變為 1 時觸發

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 0;
uint32_t CompareValue = 1;
uint32_t Value;

ret = ECAT_EvSetCompareDIBitParameters(DeviceNo, EventID, SlaveNo, BitNo, CompareValue);
if(ret < 0)
{
    printf("Failed to set compare DI Bit parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}
```

## 7.12.8. ECAT\_EvSetCompareDIParameters

### 說明:

設定事件，DI 比較觸發事件。

### 格式:

int32\_t ECAT\_EvSetCompareDIParameters(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t SlaveNo, uint16\_t OffsetByte, uint32\_t CompareValue, uint32\_t Mask)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
CompareValue	uint32_t	IN	比較的數值 注: 當 DI 數值由非比較的數值變為比較的數值時觸發事件
Mask	uint32_t	IN	遮罩 注: DI 數值&遮罩後再與比較的數值比較, 亦即比較用 DI 數值為 (DI & Mask)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte = 0;
uint32_t CompareValue = 1;
uint32_t Mask = 1;
uint32_t Value;

ret = ECAT_EvSetCompareDIParameters(DeviceNo, EventID, SlaveNo, OffsetByte, CompareValue, Mask);
if(ret < 0)
{
    printf("Failed to set compare DI parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

```

## 7.12.9. ECAT\_EvSetCompareAxisStateParameters

### 說明:

設定事件，軸狀態比較觸發事件。

### 格式:

```
int32_t ECAT_EvSetCompareAxisStateParameters(uint16_t DeviceNo, uint16_t
EventID, uint16_t AxisNo, uint32_t CompareState)
```

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
AxisNo	uint16_t	IN	軸號
CompareState	uint32_t	IN	比較的軸狀態(如表 7.5 所示)

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint32_t CompareState = MC_AS_STANDSTILL;
uint32_t Value;

ret = ECAT_EvSetCompareAxisStateParameters(DeviceNo, EventID, AxisNo, CompareState);
if(ret < 0)
{
    printf("Failed to set compare status parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

```

## 7.12.10. ECAT\_EvSetMotionCompleteParameters

### 說明:

設定事件，軸完成觸發事件。

### 格式:

int32\_t ECAT\_EvSetMotionCompleteParameters(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t AxisNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
AxisNo	uint16_t	IN	軸號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint32_t Value;

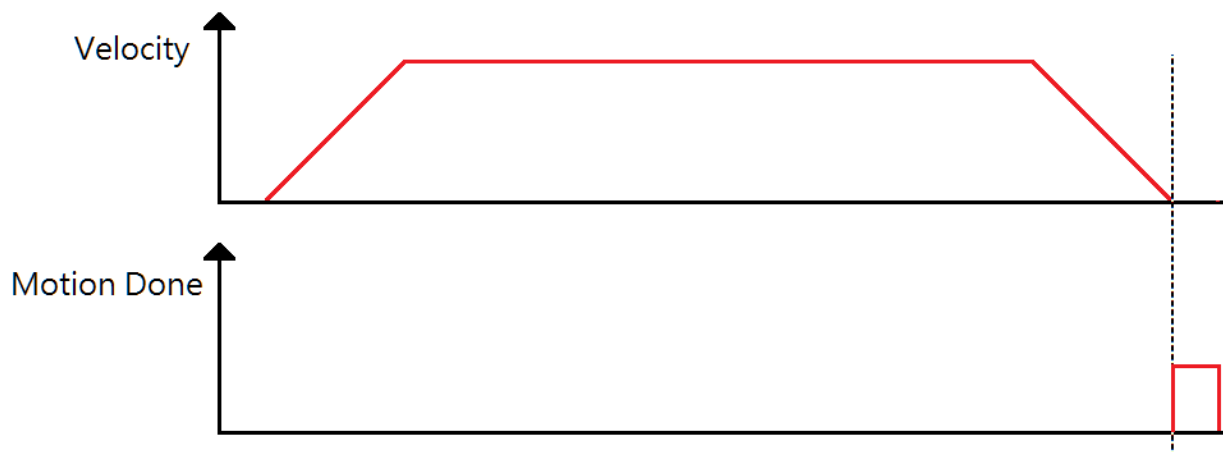
ret = ECAT_EvSetMotionCompleteParameters(DeviceNo, EventID, AxisNo);
if(ret < 0)
{
    printf("Failed to set motion complete parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

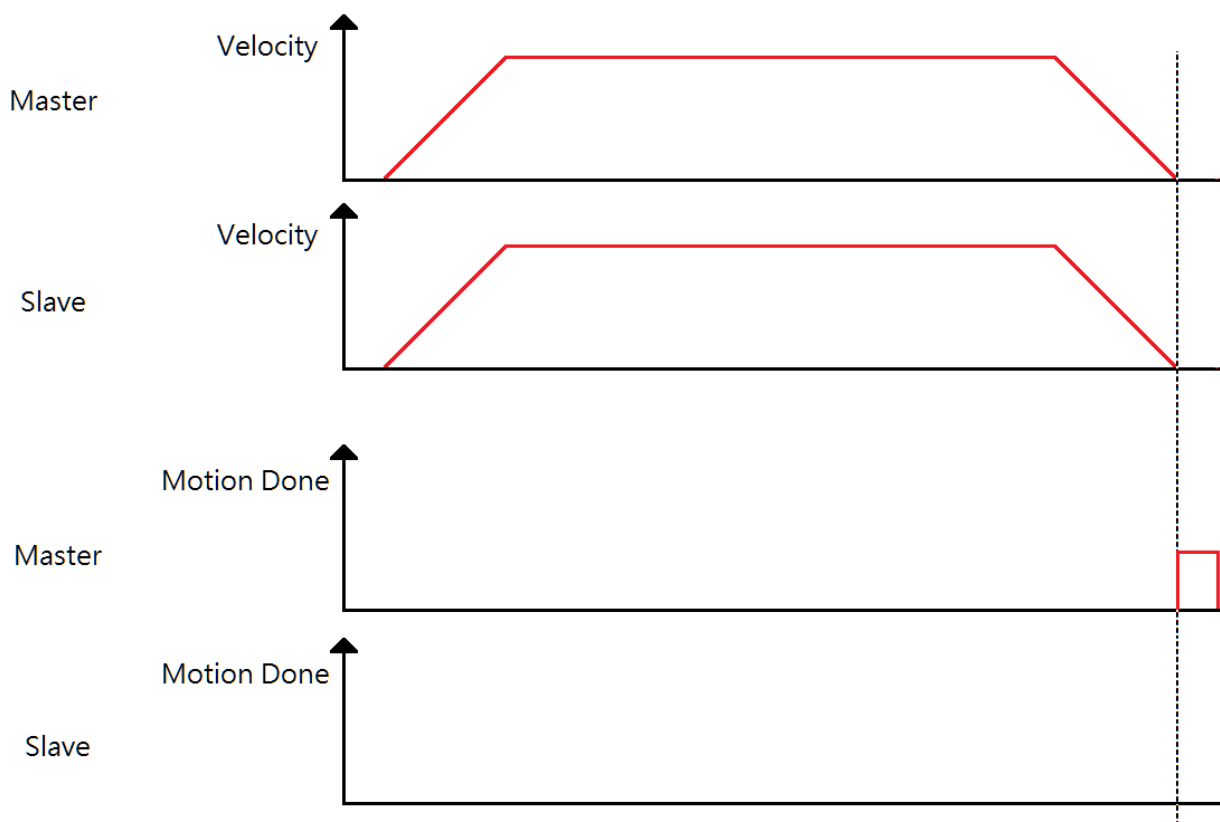
```

## 單軸運動:

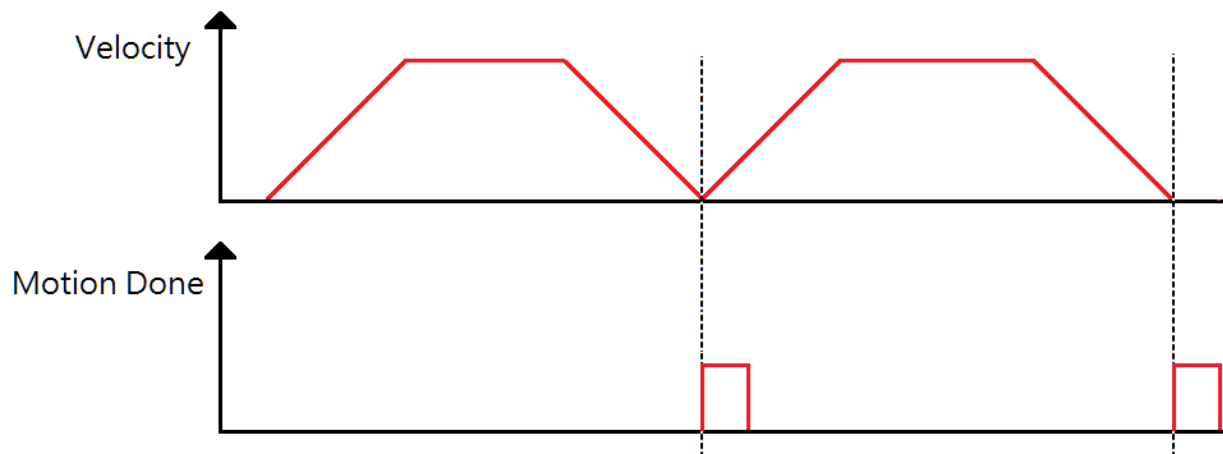




使用 gear/ cam/ gantry 時



單軸運動:Buffer 模式



## 7.12.11. ECAT\_EvSetMotionCompleteParameters\_Grp

### 說明:

設定事件，群組完成觸發事件。

### 格式:

int32\_t ECAT\_EvSetMotionCompleteParameters\_Grp(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t GrpNo)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
GrpNo	uint16_t	IN	群組號

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

## 使用範例:

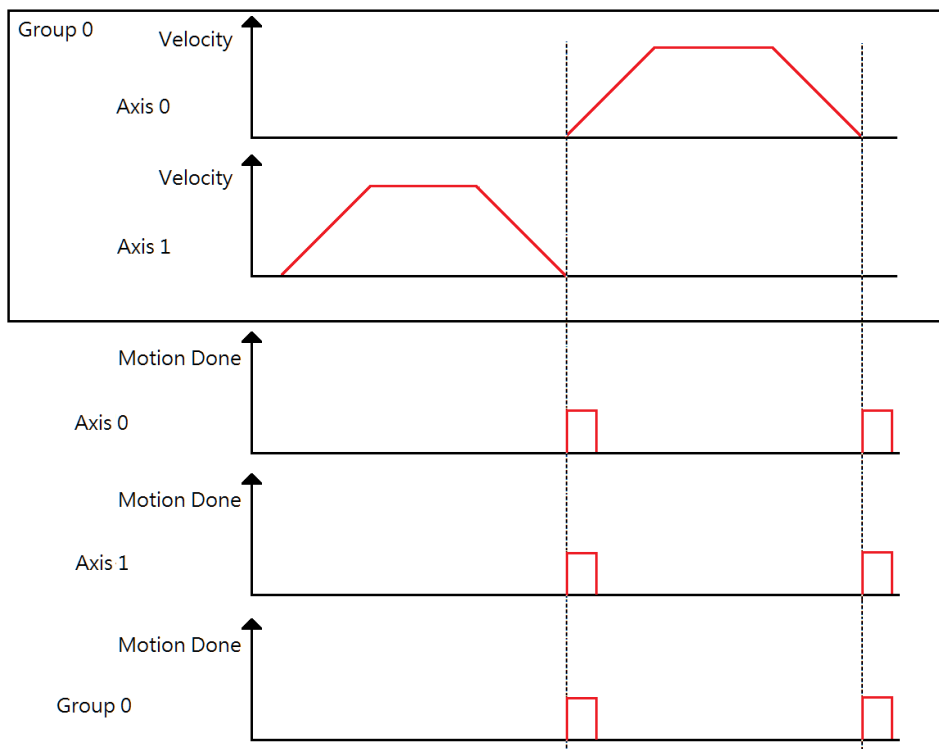
## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t GroupNo = 0;
uint32_t Value;

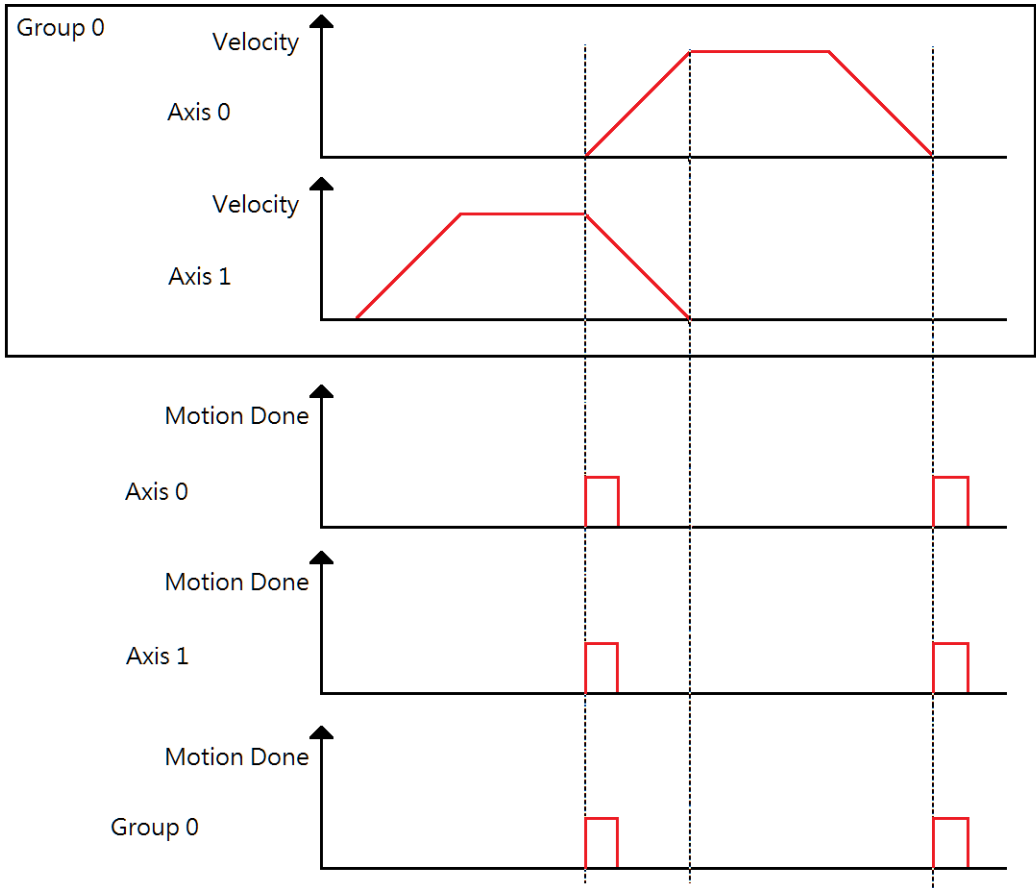
ret = ECAT_EvSetMotionCompleteParameters_Grp(DeviceNo, EventID, GroupNo);
if(ret < 0)
{
    printf("Failed to set motion complete parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}
```

群組運動:Buffer 模式



### 群組運動:Blending 模式



## 7.12.12. ECAT\_EvSetCompareAxisVelStateParameters

### 說明:

設定事件，軸速度狀態比較觸發事件。

### 格式:

int32\_t ECAT\_EvSetCompareAxisVelStateParameters(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t AxisNo, uint32\_t CompareState)

### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
AxisNo	uint16_t	IN	軸號
CompareState	uint32_t	IN	比較的軸速度狀態

### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

### 軸速度狀態定義

標頭檔定義名稱	數值	說明
MC_AS_CONSTANT_VEL	0	等速段
MC_AS_ACC	1	加速段
MC_AS_DEC	2	減速段

## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint32_t CompareState = MC_AS_CONSTANT_VEL;
uint32_t Value;

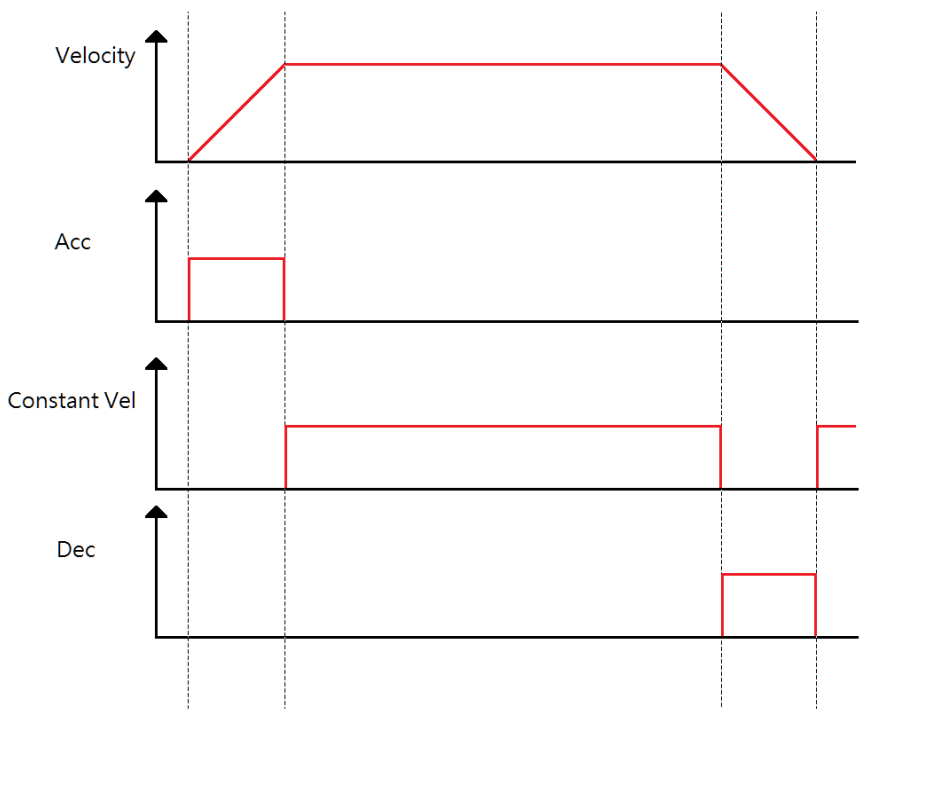
ret = ECAT_EvSetCompareAxisVelStateParameters(DeviceNo, EventID, AxisNo, CompareState);
if(ret < 0)
{
    printf("Failed to set compare velocity status parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

```

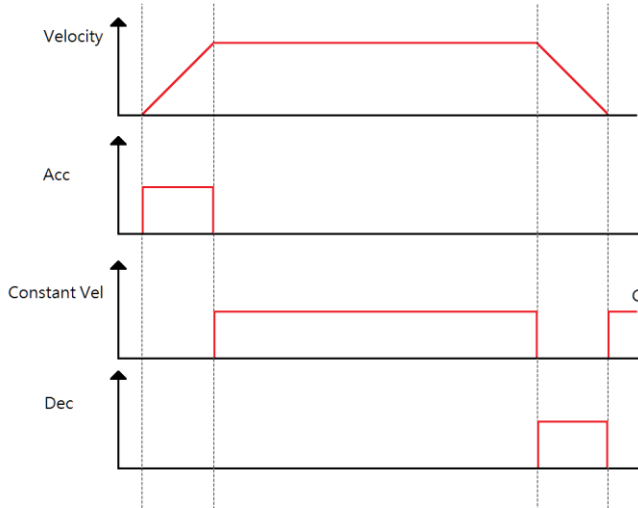
## 單軸:



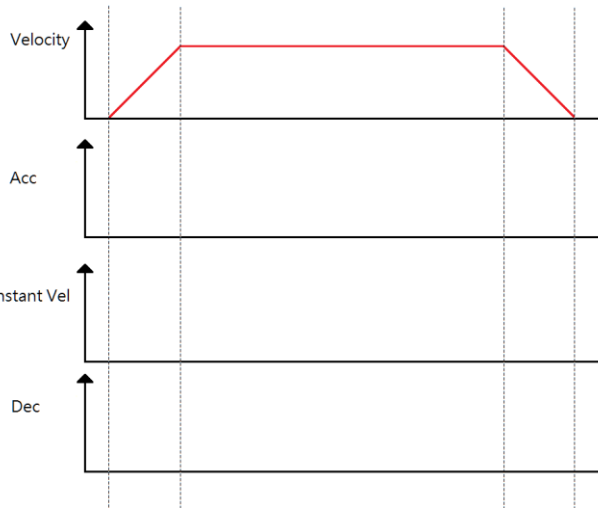


單軸: gear/ cam/ gantry

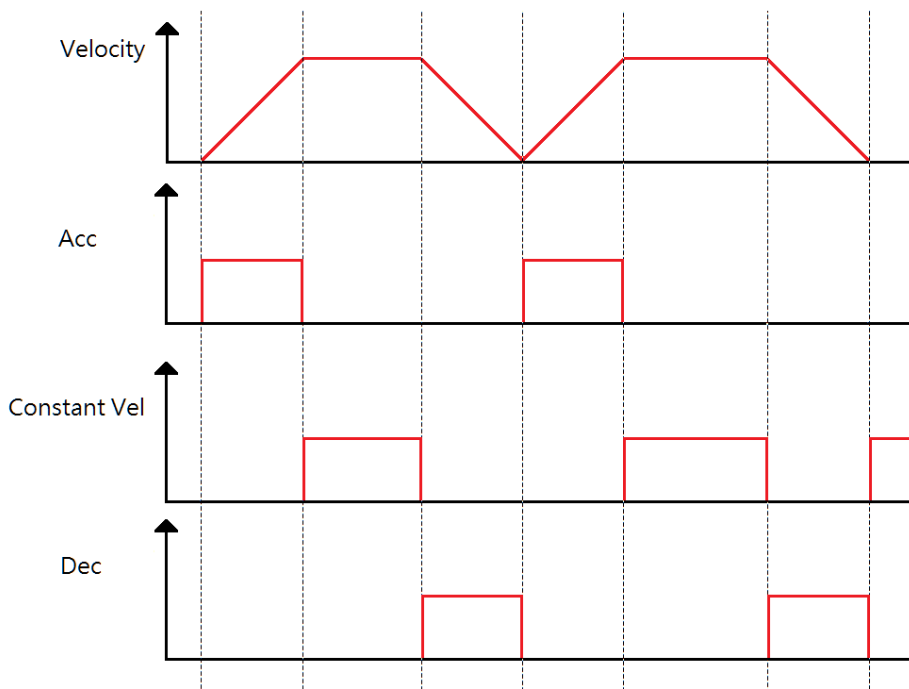
Master



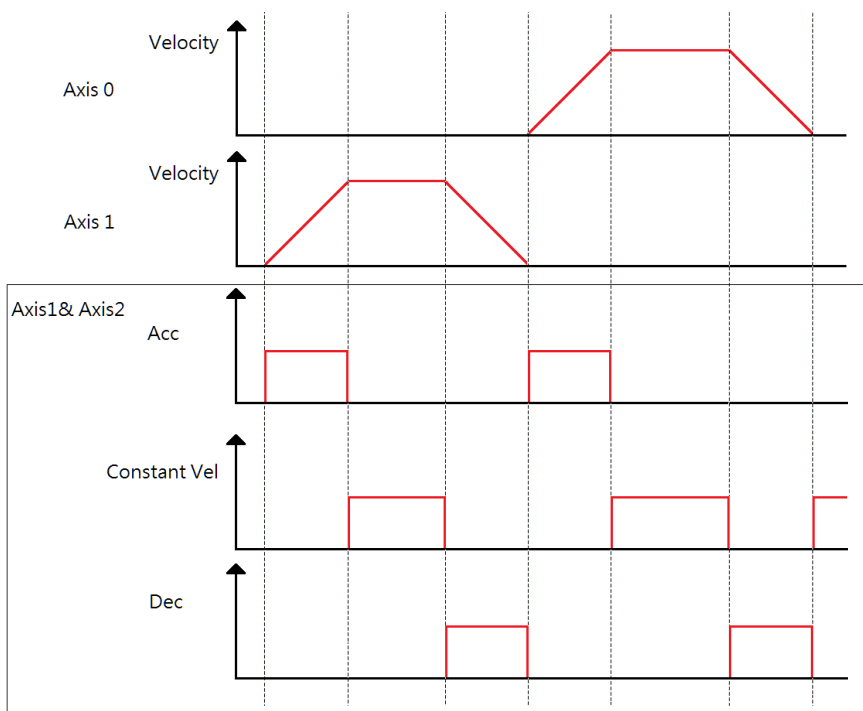
Slave



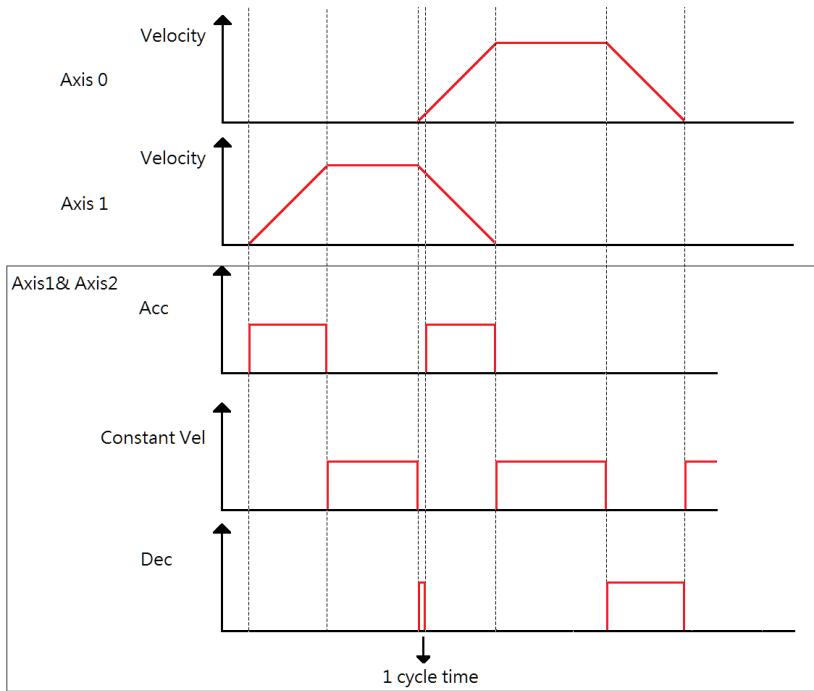
單軸: buffer



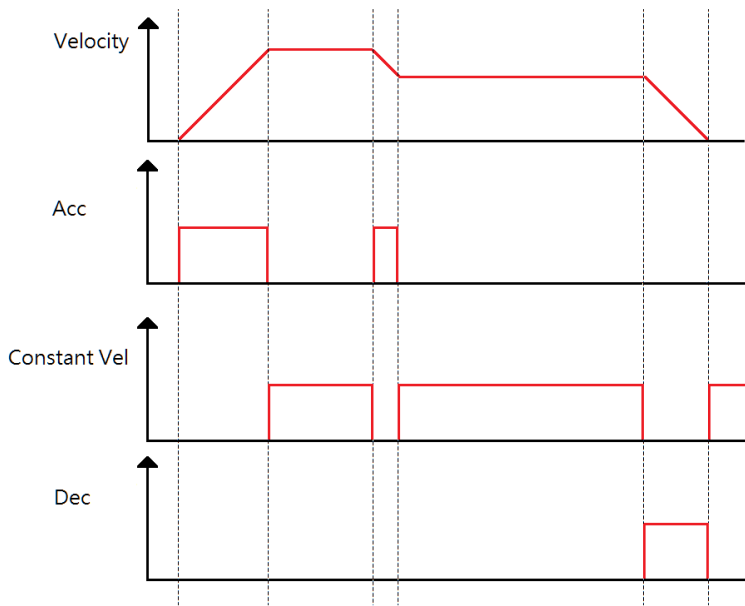
群組: buffer



群組: Blending



群組: Abort



### 7.12.13. ECAT\_EvSetCompareAiParameters

#### 說明:

設定事件，類比輸入比較觸發事件。

使用 OffsetByte 及 DataSize 讀出數值，再經過轉換( $Value = \text{int}(\text{讀出的數值}) * \text{ScaleGain} + \text{ScaleOffset}$ )之後 Value 再與 CompareValue 比較。

#### 格式:

int32\_t ECAT\_EvSetCompareAiParameters(uint16\_t DeviceNo, uint16\_t EventID, uint16\_t Operator, uint16\_t SlaveNo, uint16\_t OffsetByte, uint16\_t DataSize, double ScaleGain, double ScaleOffset, double CompareValue)

#### 參數:

名稱	類型	IN or OUT	說明
DeviceNo	uint16_t	IN	裝置編號 (卡號)
EventID	uint16_t	IN	事件編號
Operator	uint16_t	IN	運算子編號(如表 7.16 所示)
SlaveNo	uint16_t	IN	從站編號
OffsetByte	uint16_t	IN	設定相對於從站模組 TxPdo Process Image 資料開始位置偏移量
DataSize	uint16_t	IN	取得的資料大小 (最大可設定資料定義於標頭檔名稱 RW_PDO_DATA_SIZE_MAX)
ScaleGain	double	IN	輸入增益
ScaleOffset	double	IN	輸入偏移量
CompareValue	double	IN	比較的數值

#### 回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



## 使用範例:

## [C/C++]

```

int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t Operator = GREATER_THAN;
uint16_t SlaveNo = 0;
uint16_t OffsetByte = 0;
uint16_t DataSize = 2; //2 bytes
double ScaleGain = 10.0 / 0x7FFF;
double ScaleOffset = 0;
double CompareValue = 3.3;
uint32_t Value;
ret = ECAT_EvSetCompareAiParameters(DeviceNo, EventID, Operator, SlaveNo, OffsetByte, DataSize,
ScaleGain, ScaleOffset, CompareValue);
if(ret < 0)
{
    printf("Failed to set compare Ai parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something...
    }
}

```



## 8. 附錄

### 8.1. 函式錯誤回傳代碼

標頭檔定義名稱	錯誤碼	說明
ECAT_ERR_REQUEST_MASTER	-1001	核心內部請求 Master 資料失敗 請稍後再試
ECAT_ERR_ETHERNET_LINK_DOWN	-1002	乙太網路孔連接異常 請接上模組後再試
ECAT_ERR_SLAVES_STATE	-1003	從站模組 EtherCAT 狀態非 Operation 請切入 OP 後再試
ECAT_ERR_WORKING_COUNTER	-1004	EtehrCAT 工作計數器數值異常 請稍後再試 請參考”EtherCAT 診斷”
ECAT_ERR_SLAVE_CNT_EXCEEDED	-1005	連接的從站數量超出可支援數量的上限
ECAT_ERR_CREATE_DOMAIN	-1006	核心內部產生 Domain 資料失敗
ECAT_ERR_ALLOCATE_SLAVE_DATA	-1007	核心內部分配 SubDevice 資料失敗
ECAT_ERR_CONFIG_SLAVE	-1008	核心內部配置 SubDevice 資料錯誤 請稍後再試或重新編輯網絡架構
ECAT_ERR_NETWORK_MISMATCH	-1009	當前網絡連接的從站資訊和配置檔從站資訊不匹配 請編輯網絡架構後再試
ECAT_ERR_MASTER_ACTIVATE	-1010	核心內部啟用 Master 失敗



ECAT_ERR_GET_PROCESS_DATA	-1011	核心內部取得 Process Data 失敗
ECAT_ERR_CONFIG_CYCLIC_TASK	-1012	核心內部設定循環任務參數 失敗
ECAT_ERR_RUN_CYCLIC_TASK	-1013	核心內部啟用循環任務失敗
ECAT_ERR_INVALID_SLAVE_TYPE	-1014	指定錯誤種類從站模組號碼 指定的模組不支援此功能
ECAT_ERR_SAME_SLAVE_NO	-1015	重覆指定從站編號模組
ECAT_ERR_INVALID_SLAVE_NO	-1016	調用函式時指定了超出網絡 從站模組數量的編號
ECAT_ERR_INVALID_PARAM	-1017	調用函式時輸入了無效的參 數
ECAT_ERR_INVALID_DATA_SIZE	-1018	調用函式時輸入超出系統預 設的資料大小的數值
ECAT_ERR_SDO_REQUEST_BUSY	-1019	請求 SDO 命令時目前忙碌中 請稍後再試
ECAT_ERR_SDO_REQUEST_ERROR	-1020	請求 SDO 命令時目前發生錯 誤 請確認 SDO 錯誤代碼
ECAT_ERR_ALLOCATE_PDO_QUEUE	-1021	核心分配 PDO 佇列資料失敗
ECAT_ERR_INVALID_OFFSET	-1022	調用函式時輸入了無效的 Offset 參數
ECAT_ERR_INIT_MOTION	-1023	核心內部初始化運動控制所 需資料失敗
ECAT_ERR_GET_SLAVE_INFO	-1024	核心內部讀取 SubDevice 相 關資訊失敗 請稍後再試
ECAT_ERR_OPEN_FILE	-1025	核心內部開啟檔案失敗
ECAT_ERR_WRITE_FILE	-1026	核心內部寫入檔案失敗
ECAT_ERR_READ_FILE	-1027	核心內部讀取檔案失敗
ECAT_ERR_FUNC_NOT_SUPPORT	-1028	功能函式不支援

ECAT_ERR_INVALID_CHANNEL	-1029	無效的 Channel 參數
ECAT_ERR_EMG_HAPPENED	-1030	緊急停止信號觸發 解除停止信號後可 Reset
ECAT_ERR_INVALID_PID_NO	-1031	無效的 PID 編號
ECAT_ERR_TIMER_NOT_ACTIVATED	-1032	TIMER 尚未啟動 請先啟用 TIMER
ECAT_ERR_ALL_EVENT_CREATE	-1033	所有事件已被創建
ECAT_ERR_EVENT_NOT_CREATE	-1034	事件未被創建
ECAT_ERR_INVALID_EVENTID	-1035	無效的事件編號
ECAT_ERR_INVALID_FILTER_TYPE	-1036	無效的濾波器類型
ECAT_ERR_SLAVES_ALIAS	-1037	重複的別名 / 別名為零 請重新設定別名
ECAT_ERR_SLAVES_ALIAS_NOT_EXIST	-1038	不存在的別名 請重新設定別名
ECAT_ERR_OPTASK	-1039	從站模組 EtherCAT 狀態在 Operation 請退回 PreOP 後再試
ECAT_ERR_ALL_BUFFER_USED	-1042	無緩衝區可用
ECAT_ERR_BUFFER_NOT_ENABLE	-1043	緩衝區未啟用
ECAT_ERR_DEACTIVATE_SLAVES	-1046	正在從 OP 轉換為 PreOP
ECAT_ERR_MASTER_BUSY_SCANNING	-1047	掃描模組中 請稍後再試
ECAT_ERR_PDO_BUFFER_OVERFLOW	-1050	PDO 緩衝區溢位
ECAT_ERR_PDO_BUFFER_NOT_ENABLE	-1051	緩衝區未啟用
ECAT_ERR_PDO_CREATE_LOG_DATA	-1052	產生緩衝區失敗
ECAT_ERR_MC_NOT_ENABLE_DC	-1100	初始化運動控制時未設定啟 用 DC 功能 請編輯網絡架構後再試
ECAT_ERR_MC_TIME_OUT	-1101	調用運動控制函式時，與伺服 驅動器進行通訊時通訊逾時 請確認模組狀態是否正常，是

		否有 ALM 或 Error 發生，排除後再試
ECAT_ERR_MC_AXIS_CNT_EXCEEDED	-1102	運動控制初始化時，輸入超出可支援的總軸數上限
ECAT_ERR_MC_NOT_INITIALIZED	-1103	調用運動控制函式時，尚未調用初始化運動控制 請先執行 ECAT_McInIt
ECAT_ERR_MC_INVALID_AXIS_NO	-1104	調用運動控制函式時，輸入了超出初始化軸數之軸號
ECAT_ERR_MC_NOT_AXIS_SERVO_ON	-1105	調用運動控制函式時，指定軸號尚未 Servo ON
ECAT_ERR_MC_INVALID_AXIS_STATE	-1106	調用運動控制函式時，指定軸號目前狀態非核心可接受軸狀態 請確認當前的軸狀態
ECAT_ERR_MC_DRIVE_FAULT	-1107	指定軸號目前伺服驅動器產生錯誤異常 請排除錯誤後可 Reset
ECAT_ERR_MC_DRIVE_WARNING	-1108	指定軸號目前伺服驅動器產生警告異常 請排除錯誤後可 Reset
ECAT_ERR_MC_INVALID_PARAM	-1109	調用運動控制函式時，輸入了無效的參數
ECAT_ERR_MC_HOMING	-1110	指定軸號原點復歸時發生錯誤 請稍後再試
ECAT_ERR_MC_LIMIT_ACTIVE	-1111	指定軸號運動控制時碰觸到極限開關 Reset 後可往另一方向運動
ECAT_ERR_MC_INVALID_ACC_TIME	-1112	設定單軸或群組加減速時間超出核心預設的大小數值

ECAT_ERR_MC_INVALID_GROUP_NO	-1113	調用運動控制函式時，輸入了超出核心可支援的最大群組數量
ECAT_ERR_MC_INVALID_GROUP_STATE	-1114	調用運動控制函式時，指定群組號碼目前狀態非核心可接受軸狀態 請確認當前群組狀態
ECAT_ERR_MC_AXIS_WAS_IN_GROUP	-1115	指定軸號已在此群組內
ECAT_ERR_MC_AXIS_IN_OTHER_GROUP	-1116	指定軸號目前在其他群組內 請將該軸移出群組後再試
ECAT_ERR_MC_GROUP_CMD_ALLOCATE	-1117	調用運動控制函式時，產生該運動命令失敗
ECAT_ERR_MC_GROUP_CMD_BUFFER_OVERFLOW	-1118	調用運動控制函式時，運動命令緩衝區已滿
ECAT_ERR_MC_INVALID_AXIS_SYNC_MODE	-1119	調用不同模式之同步運動控制
ECAT_ERR_MC_INVALID_PROFILE_NO	-1120	調用運動控制函式時，輸入超出核心可支援最大 Profile 編號
ECAT_ERR_MC_INVALID_GROUP_MOVE_CMD	-1121	調用不相匹配之群組運動控制函式
ECAT_ERR_MC_GROUP_CMD_MODE_NOT_SUPPORT	-1122	調用的運動控制函式不支援目前的群組命令模式
ECAT_ERR_MC_INVALID_ACC_DEC_TYPE	-1123	無效的加減速模式
ECAT_ERR_MC_INVALID_VEL	-1124	無效速度參數
ECAT_ERR_MC_INVALID_ANGLE	-1125	無效旋轉角度參數
ECAT_ERR_MC_INVALID_RADIUS	-1126	無效半徑參數
ECAT_ERR_MC_INVALID_END_POS	-1127	無效終點位置參數
ECAT_ERR_MC_INVALID_ECAM_TABLE_NO	-1128	無效 E-CAM 表格編號參數
ECAT_ERR_MC_INVALID_NORMAL_VECTOR	-1129	無效法向量參數
ECAT_ERR_MC_NOT_SETUP	-1130	尚未設定

ECAT_ERR_MC_GREATER_THAN_MAX_RODLENGTH	-1131	計算出的數值超過最大伸長量
ECAT_ERR_MC_LESS_THAN_RODLENGTH	-1132	計算出的數值小於最小伸長量
ECAT_ERR_MC_GREATER_THAN_RECORD_COUNT	-1133	超出最大紀錄筆數
ECAT_ERR_MC_SOFTWARE_LIMIT_ACTIVATE	-1134	指定軸號運動控制時碰觸到軟體極限 Reset 後可向另一方向運動
ECAT_ERR_MC_GANTRY_POS_EXCESSIVE_DEVIATION	-1135	位置誤差大於設定值 請 Reset 後再試
ECAT_ERR_MC_GROUP_NO_NOT_SUPPORT	-1136	不支援的群組
ECAT_ERR_MC_INVALID_MOVE_CMD	-1137	無效的運動命令 請稍後再試
ECAT_ERR_MC_QUEUE_IS_FULL	-1138	緩衝區已滿
ECAT_ERR_MC_COORDINATE_TRANS_ON	-1139	座標轉換為開啟狀態
ECAT_ERR_MC_HAVE_NOT_BEEN_SET	-1140	尚未設定
ECAT_ERR_MC_HOMIE_NOT_DONE	-1141	未完成歸原點
ECAT_ERR_MC_INHIBITED_FUNCTION	-1142	受限制的函式
ECAT_ERR_MC_LACK_PDOS	-1143	缺少必要的 PDO
ECAT_ERR_MC_SAFETY_STOP	-1144	安全停止
ECAT_ERR_MC_GANTRY_SYNC_ERR	-1145	安全停止 當主軸或從軸發生以下情形 錯誤發生/歸原點/Servo off
ECAT_ERR_MC_DIFFERENT_SETTINGS	-1146	與先前的設定不同
ECAT_ERR_DIFFERENT_SETTINGS	-1149	與先前的設定不同
ECAT_ERR_IPC_INVALID_DEVICE_NO	-1201	輸入無效的裝置編號
ECAT_ERR_IPC_DEVICE_IS_OPEN	-1202	目前裝置為開啟狀態
ECAT_ERR_IPC_DEVICE_NOT_OPEN	-1203	目前裝置當未開啟 請執行 ECAT_OpenDevice
ECAT_ERR_IPC_CREATE_HANDLE	-1204	產生與核心內部通訊元件失敗

ECAT_ERR_IPC_BUSY	-1205	與核心內部進行通訊時核心內部忙碌中 請稍後再試
ECAT_ERR_IPC_TIME_OUT	-1206	與核心內部進行通訊時通訊逾時 請稍後再試 若一直發生該問題，請確認EcatUtility 應用程式是否可以使用，若無法開啟，關機後再開機可排除此問題
ECAT_ERR_IPC_INVALID_CMD	-1207	與核心內部進行通訊產生無效的通訊命令
ECAT_ERR_IPC_WRITE_SHM	-1208	與核心內部進行通訊時寫入共享資料失敗
ECAT_ERR_IPC_READ_SHM	-1209	與核心內部進行通訊時讀取共享資料失敗
ECAT_ERR_IPC_RUN_DOWN_UP_LOAD	-1210	與核心內部進行通訊時，處理上下載資料失敗
ECAT_ERR_IPC_INVALID_SHM	-1211	無效的共享資料參數 請勿進入休眠狀態 Windows 系統請關閉”快速啟動”功能 重新開機後再試
ECAT_ERR_IPC_DEVICE_NOT_READY	-1212	與核心內部進行通訊時裝置尚未就緒
ECAT_ERR_IPC_NOT_SUPPROT	-1213	功能不支援
ECAT_ERR_DRV_GET_INFO	-1301	讀取驅動程式相關資訊失敗
ECAT_ERR_DRV_CREATE_HANDLE	-1302	產生驅動程式通訊元件失敗
ECAT_ERR_DRV_IOCTL	-1303	驅動程式調用 I/O 控制函式失敗
ECAT_ERR_DRV_DEVICE_NOT_FOUND	-1304	驅動程式無法找到相匹配之

	<p>裝置</p> <p>請確認裝置管理員是否找到裝置</p> <p>請確認是否有另一支程式已開啟裝置</p> <p>請重新開機後再試</p> <p>若以上問題皆排除仍有錯誤發生，請重新安裝驅動程式</p>
--	---

## 8.2. SDO 操作錯誤終止代碼

錯誤碼	說明
0x05030000	分段傳輸時 toggle bit 無變化
0x05040000	SDO 傳輸逾時
0x05040001	命令碼無效或未知
0x05040005	記憶體不足
0x06010000	不支援存取的物件(object)
0x06010001	嘗試讀取一個唯寫的物件(object)
0x06010002	嘗試寫入一個唯讀的物件(object)
0x06020000	在物件字典(object directory)此物件(object)不存在
0x06040041	此物件(object)不能映射到 PDO 中
0x06040042	要映射的物件(object)數量和長度超過 PDO 數據長度
0x06040043	一般參數不相容
0x06040047	裝置一般內部不相容
0x06060000	由於硬體錯誤導致存取失敗
0x06070010	資料大小不匹配，服務參數長度不匹配
0x06070012	資料大小不匹配，服務參數長度過長
0x06070013	資料大小不匹配，服務參數長度過短
0x06090011	子索引不存在
0x06090030	寫入數據超出範圍
0x06090031	寫入數據太大
0x06090032	寫入數據太小
0x06090036	最大值小於最小值
0x08000000	一般性錯誤
0x08000020	數據無法傳輸或儲存到應用程序端
0x08000021	由於本地控制原因，數據無法傳輸或儲存到應用程序端
0x08000022	由於當前裝置狀態原因，數據無法傳輸或儲存到應用程序端
0x08000023	動態產生物件字典(object directory)失敗，或沒有找到物件字典(object directory)
0x05030000	分段傳輸時 toggle bit 無變化



0x05040000	SDO 傳輸逾時
0x05040001	命令碼無效或未知
0x05040005	記憶體不足
0x06010000	不支援存取的物件(object)
0x06010001	嘗試讀取一個唯寫的物件(object)
0x06010002	嘗試寫入一個唯讀的物件(object)
0x06020000	在物件字典(object directory)此物件(object)不存在
0x06040041	此物件(object)不能映射到 PDO 中
0x06040042	要映射的物件(object)數量和長度超過 PDO 數據長度
0x06040043	一般參數不相容
0x06040047	裝置一般內部不相容
0x06060000	由於硬體錯誤導致存取失敗
0x06070010	資料大小不匹配，服務參數長度不匹配
0x06070012	資料大小不匹配，服務參數長度過長
0x06070013	資料大小不匹配，服務參數長度過短
0x06090011	子索引不存在
0x06090030	寫入數據超出範圍
0x06090031	寫入數據太大
0x06090032	寫入數據太小
0x06090036	最大值小於最小值
0x08000000	一般性錯誤
0x08000020	數據無法傳輸或儲存到應用程序端
0x08000021	由於本地控制原因，數據無法傳輸或儲存到應用程序端
0x08000022	由於當前裝置狀態原因，數據無法傳輸或儲存到應用程序端
0x08000023	動態產生物件字典(object directory)失敗，或沒有找到物件字典(object directory)

## 8.3. 手冊修訂記錄

本章提供此使用手冊的修訂記錄。

下表提供此文件每次修訂的日期與說明。

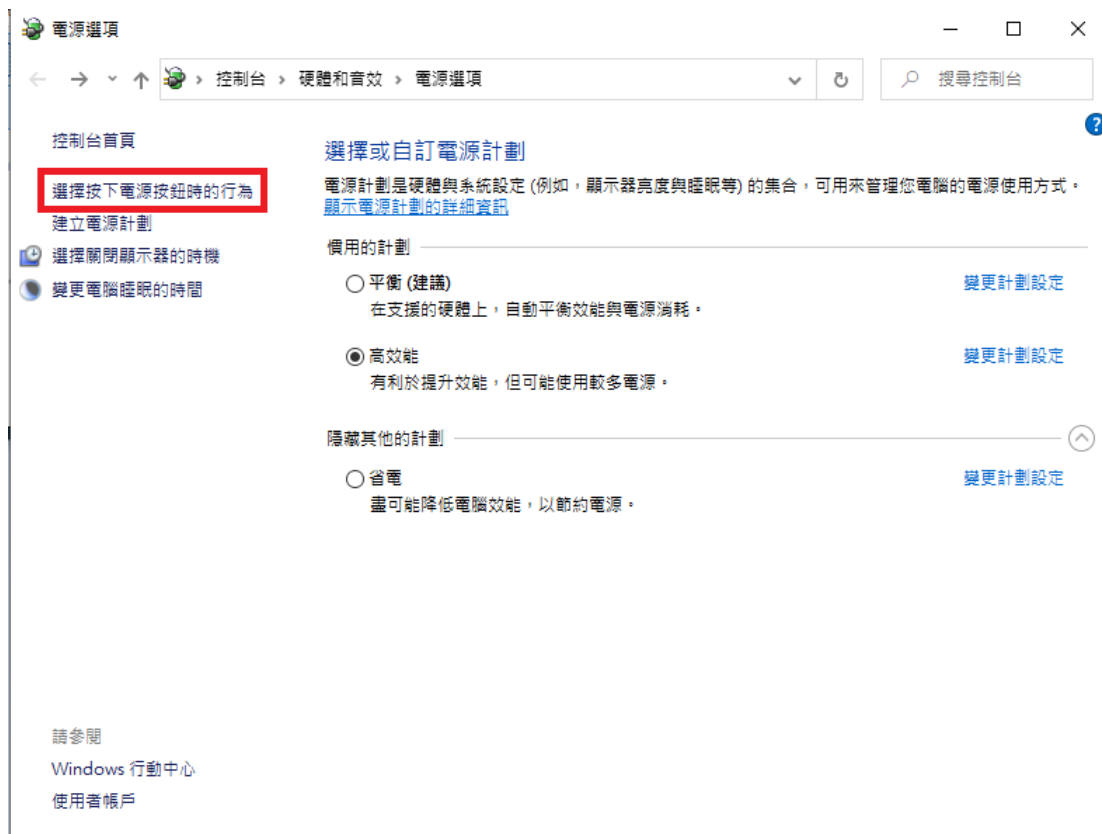
版本	發行日	說明
1.0.24	2021 年	請看 1.1 版本更新資訊

## 8.4. 如何關閉 Windows 10/11 中的快速啟動功能

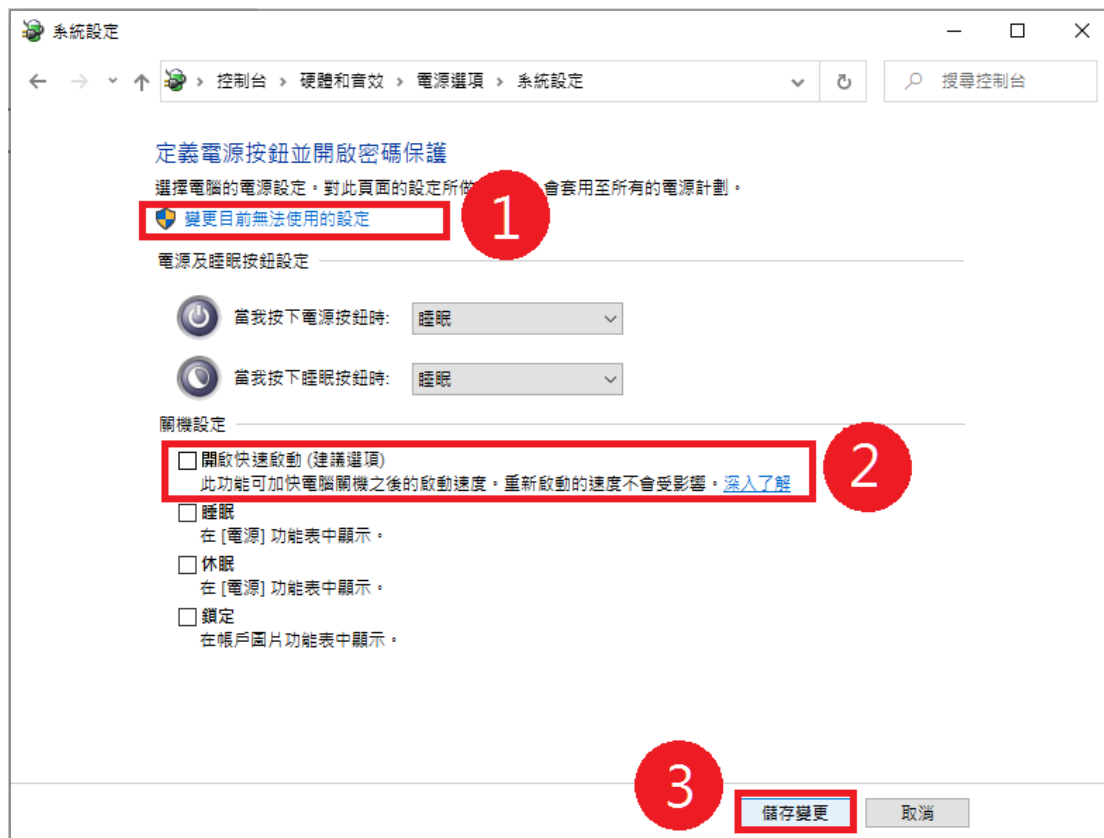
1. 在 Windows 搜尋欄輸入”選擇電源計畫”並開啟。



## 2. 點擊”選擇按下電源按鈕時的行為”。



3. 點擊“變更目前無法使用的設定”。
4. 取消勾選“快速開機啟動”。
5. 點擊“儲存變更”，設定完成。

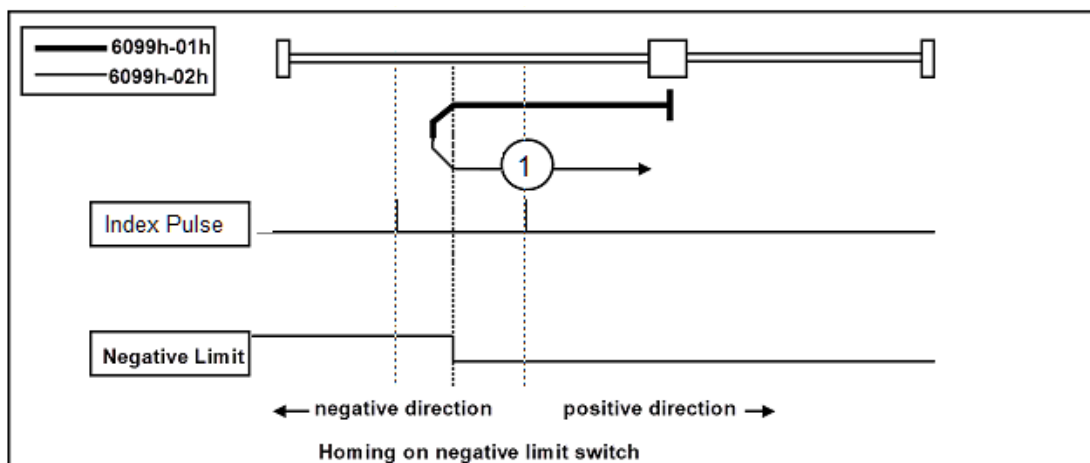


## 8.5. CiA402 原點復歸模式(Homing Mode/ hm mode)

僅供參考，實際復歸模式請參閱驅動器手冊。

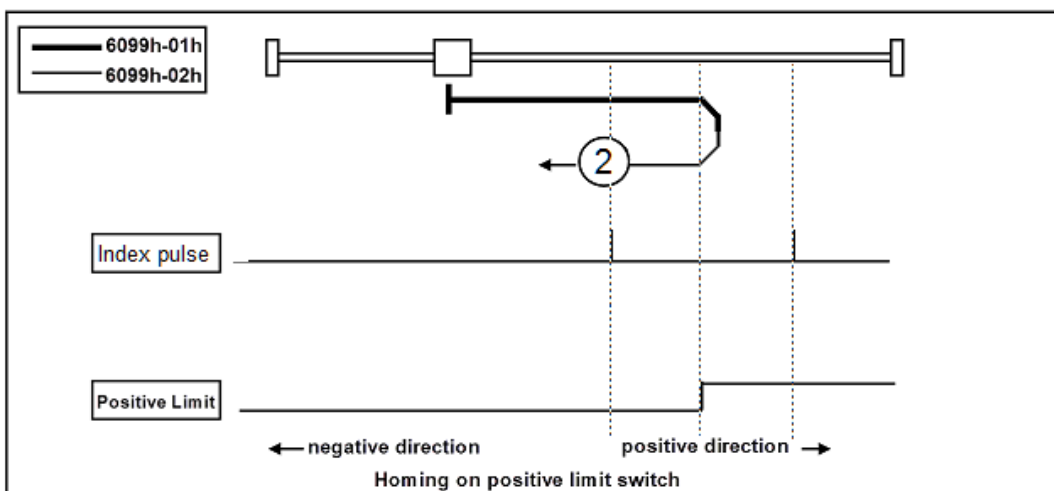
### 8.5.1. Method 1

- 負極限如果在動作開始時未激活，初始動作方向為負方向。
- 負極限如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是負極限的狀態變化後在正方向側位置的最初的 Index pulse 檢出位置。



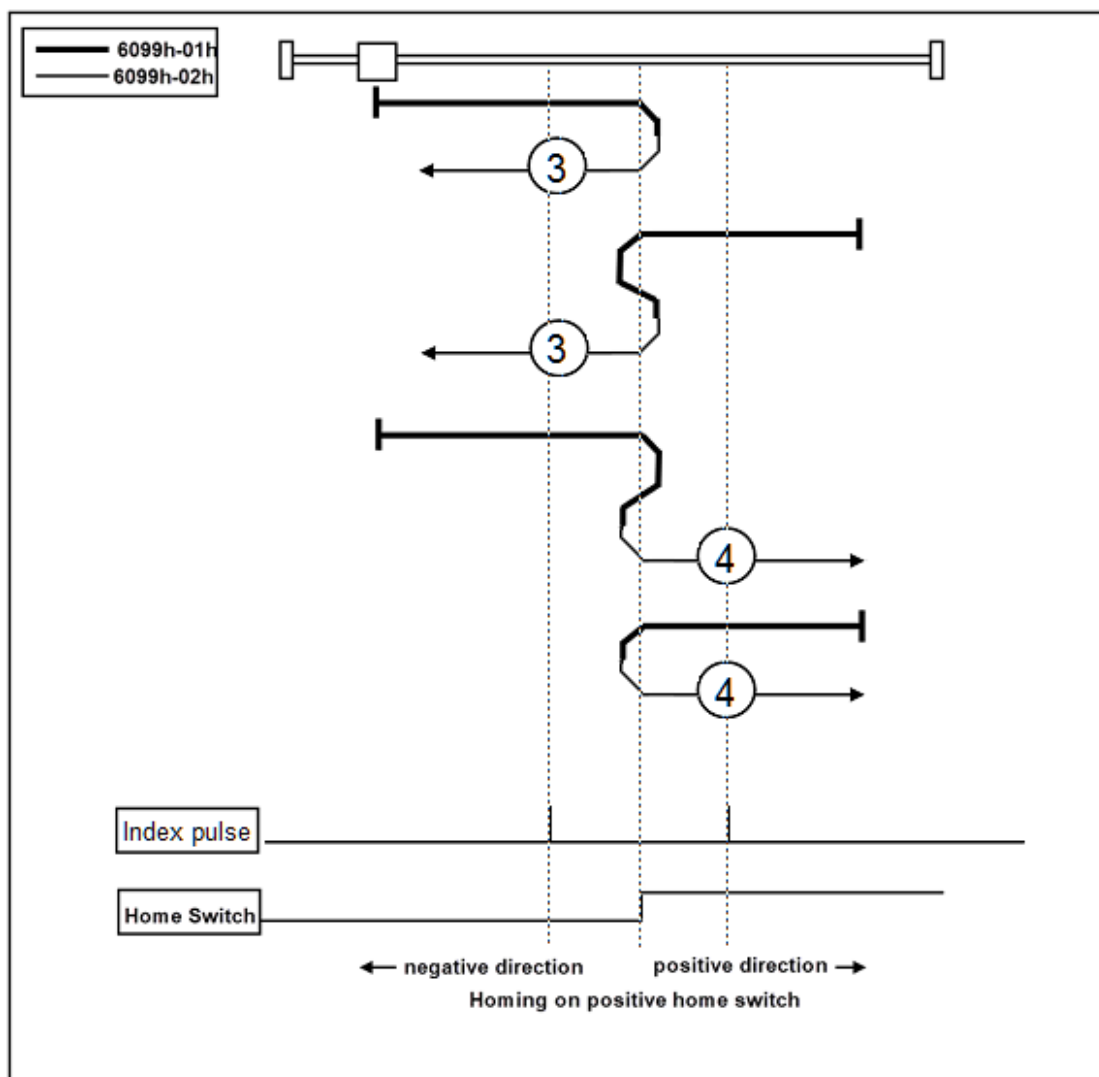
## 8.5.2. Method 2

- 正極限如果在動作開始時未激活，初始動作方向為正方向。
- 正極限如果在動作開始時已經激活，初始動作方向為負方向。
- 原點檢出位置是正極限的狀態變化後在負方向側位置的最初的 Index pulse 檢出位置。



### 8.5.3. Method 3, 4

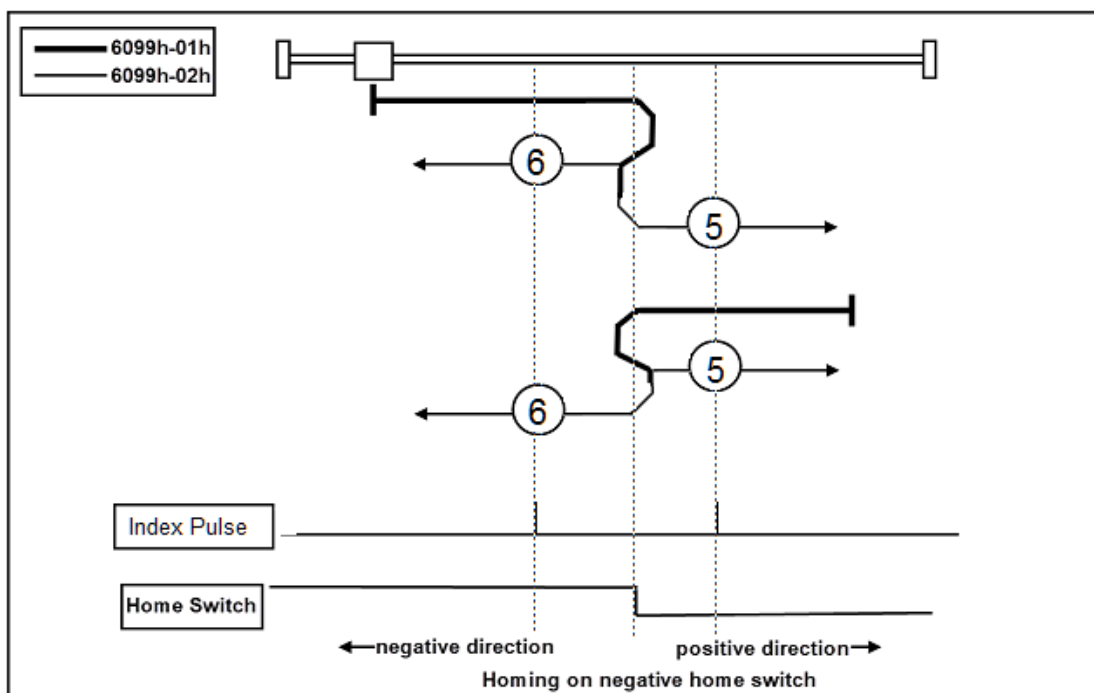
- 原點開關如果在動作開始時未激活，初始動作方向為正方向。
- 原點開關如果在動作開始時已經激活，初始動作方向為負方向。
- 原點檢出位置是原點開關的狀態變化後在正方向側或者負方向側的最初的 Index pulse 檢出位置。





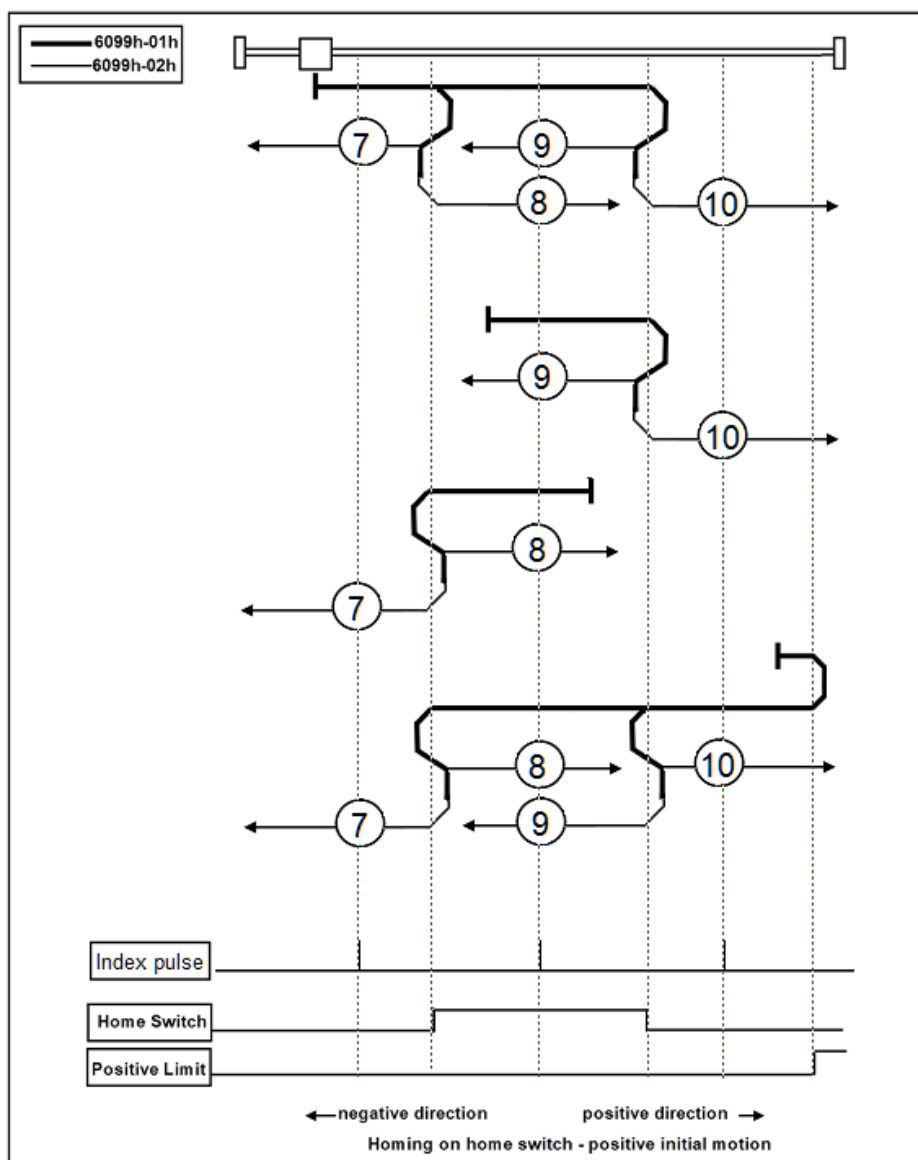
### 8.5.4. Method 5, 6

- 原點開關如果在動作開始時未激活，初始動作方向為負方向。
- 原點開關如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是原點開關的狀態變化後在正方向側或者負方向側的最初的 Index pulse 檢出位置。



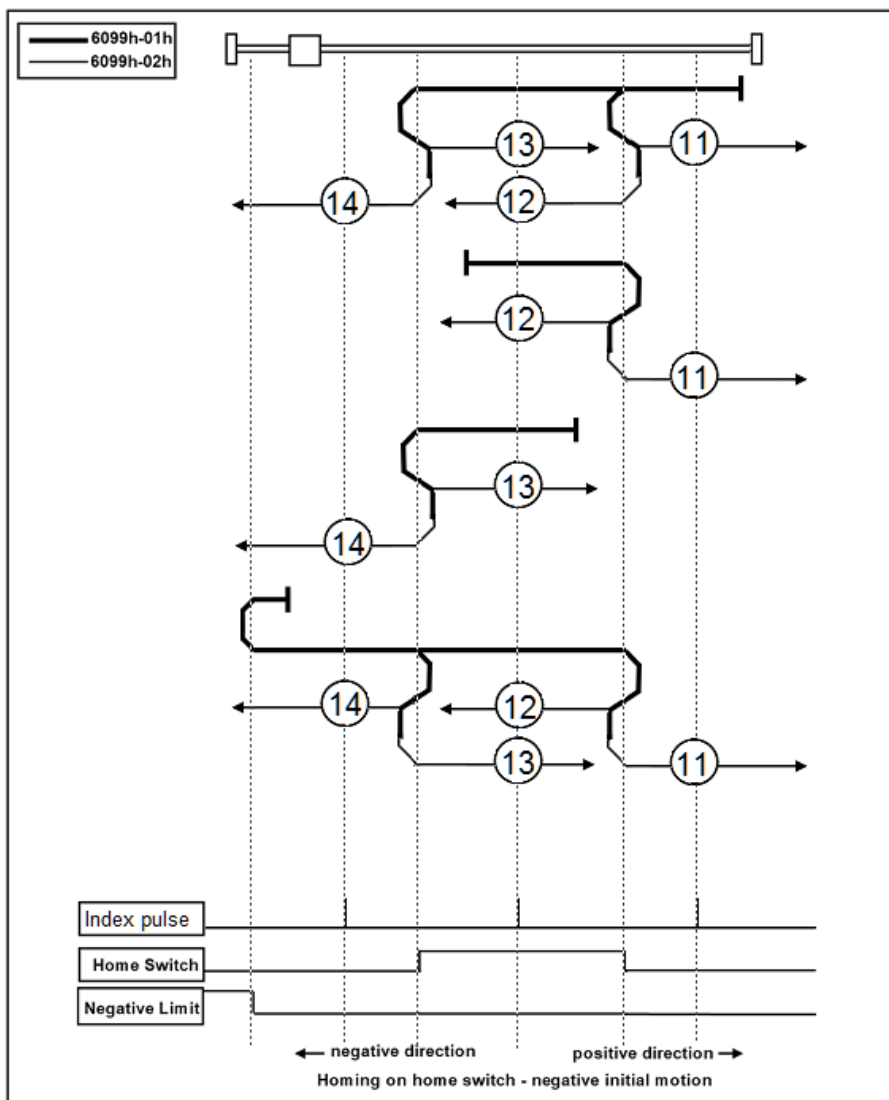
### 8.5.5. Method 7, 8, 9, 10

- 方法 7, 8 的原點開關如果在動作開始時已經激活，初始動作方向為負方向。
- 方法 9, 10 的原點開關如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是原點開關的狀態變化後在正方向側或者負方向側的最初的 Index pulse 檢出位置。



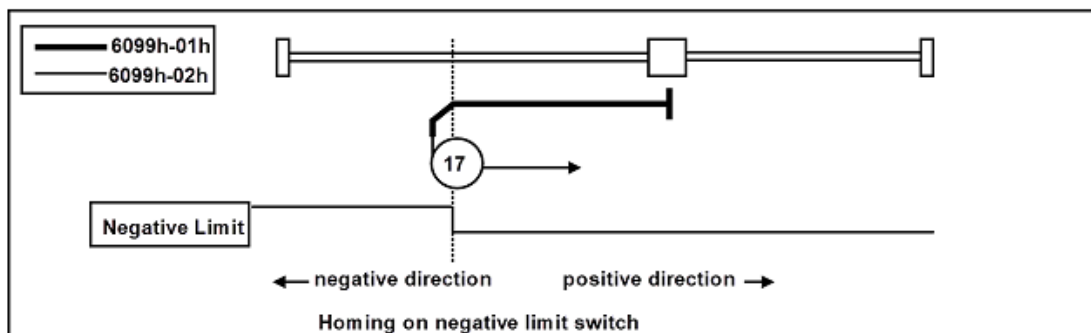
### 8.5.6. Method 11, 12, 13, 14

- 方法 13, 14 的原點開關如果在動作開始時已經激活，初始動作方向為負方向。
- 方法 11, 12 的原點開關如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是原點開關的狀態變化後在正方向側或者負方向側的最初的 Index pulse 檢出位置。



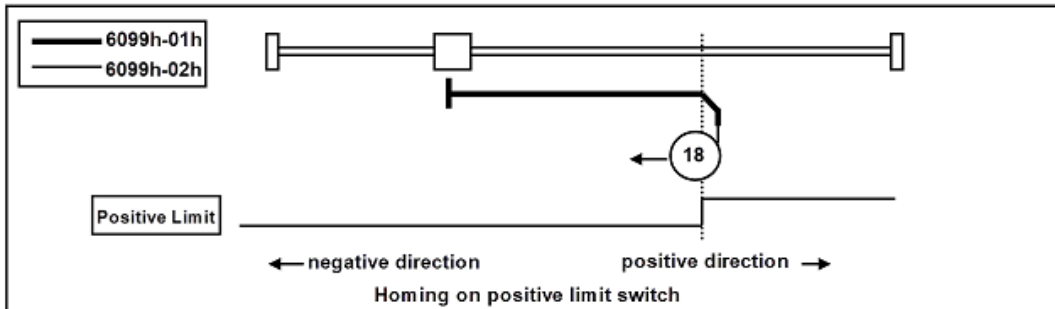
### 8.5.7. Method 17

- 負極限如果在動作開始時未激活，初始動作方向為負方向。
- 負極限如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是負極限的狀態變化時的位置。



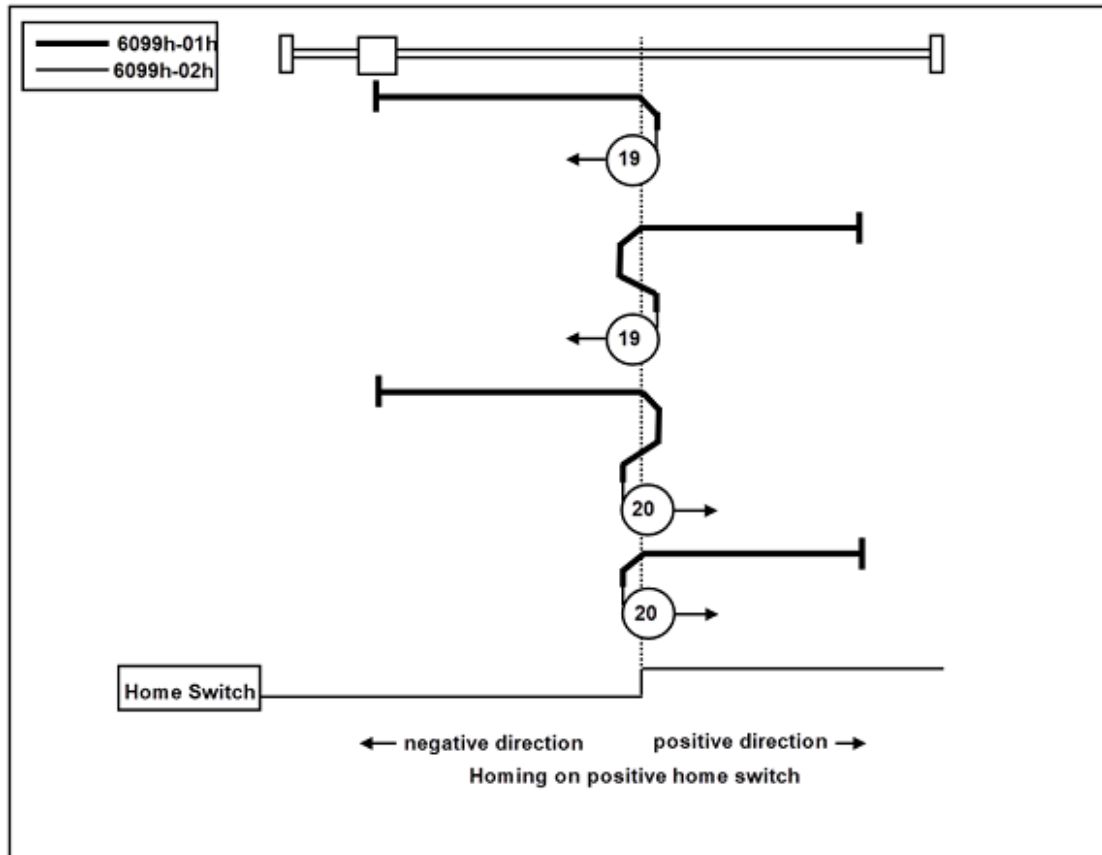
### 8.5.8. Method 18

- 正極限如果在動作開始時未激活，初始動作方向為正方向。
- 正極限如果在動作開始時已經激活，初始動作方向為負方向。
- 原點檢出位置是正極限的狀態變化時的位置。



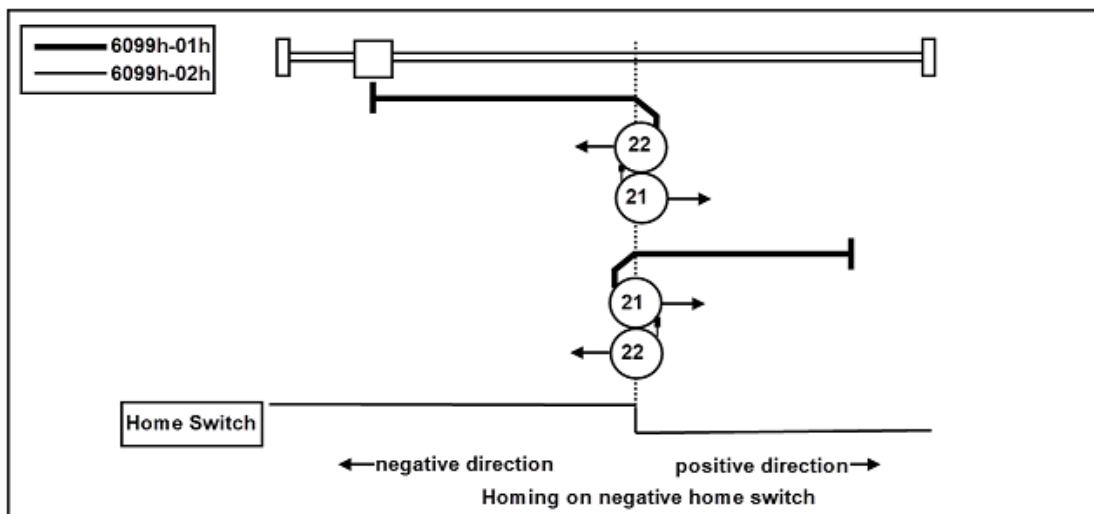
### 8.5.9. Method 19, 20

- 原點開關如果在動作開始時未激活，初始動作方向為正方向。
- 原點開關如果在動作開始時已經激活，初始動作方向為負方向。
- 原點檢出位置是原點開關的狀態變化時的位置。



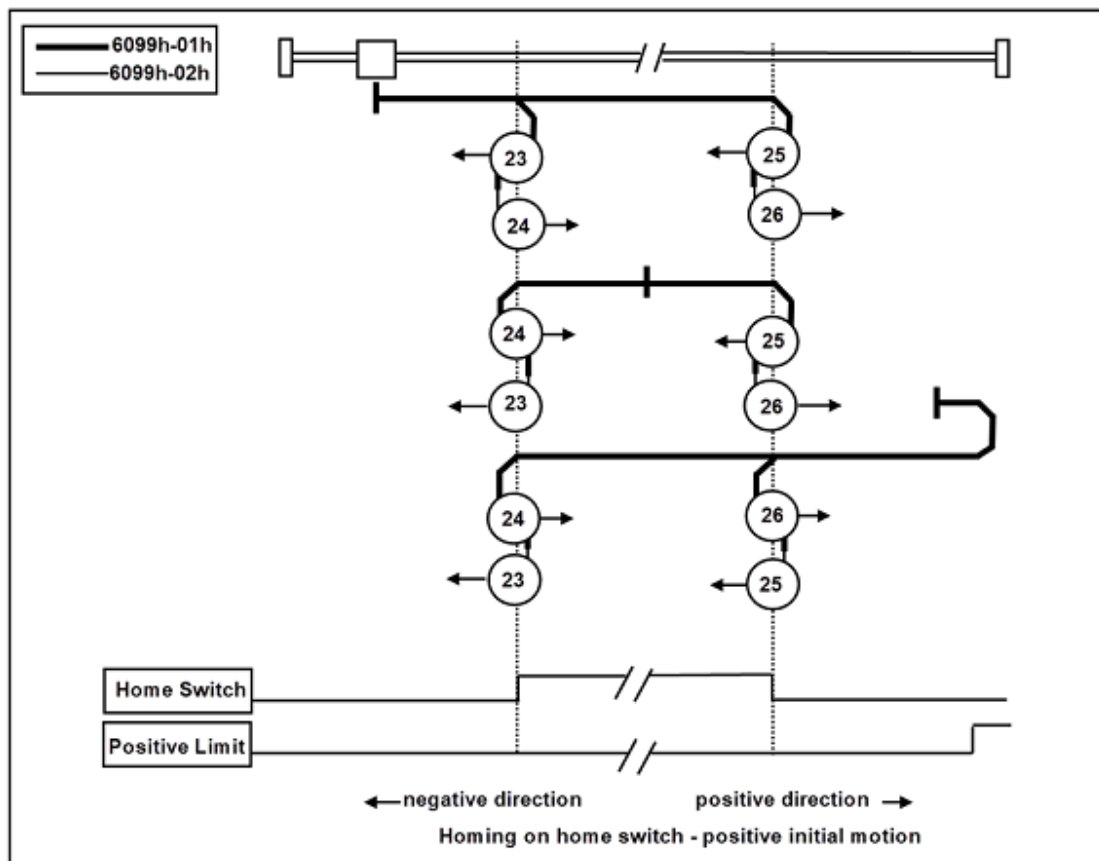
### 8.5.10. Method 21, 22

- 原點開關如果在動作開始時未激活，初始動作方向為負方向。
- 原點開關如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是原點開關的狀態變化時的位置。



### 8.5.11. Method 23, 24, 25, 26

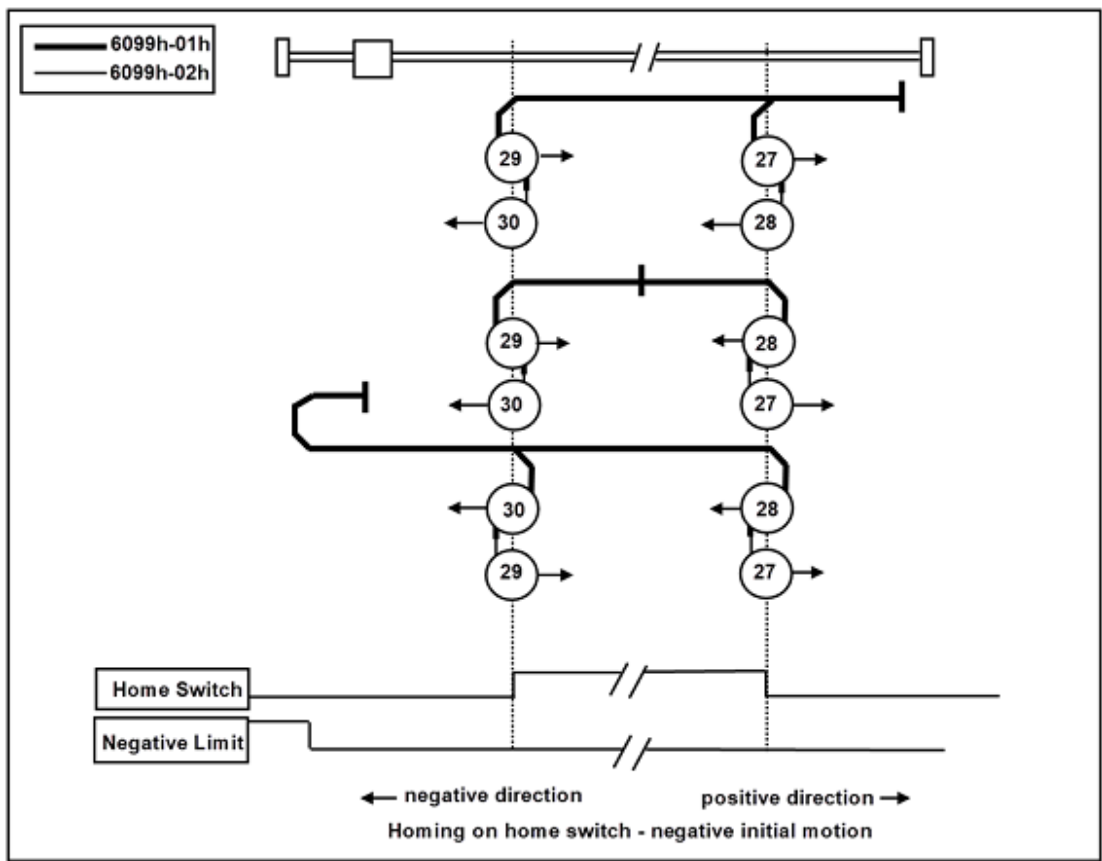
- 方法 23, 24 的原點開關如果在動作開始時已經激活，初始動作方向為負方向。
- 方法 25, 26 的原點開關如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是原點開關的狀態變化時的位置。





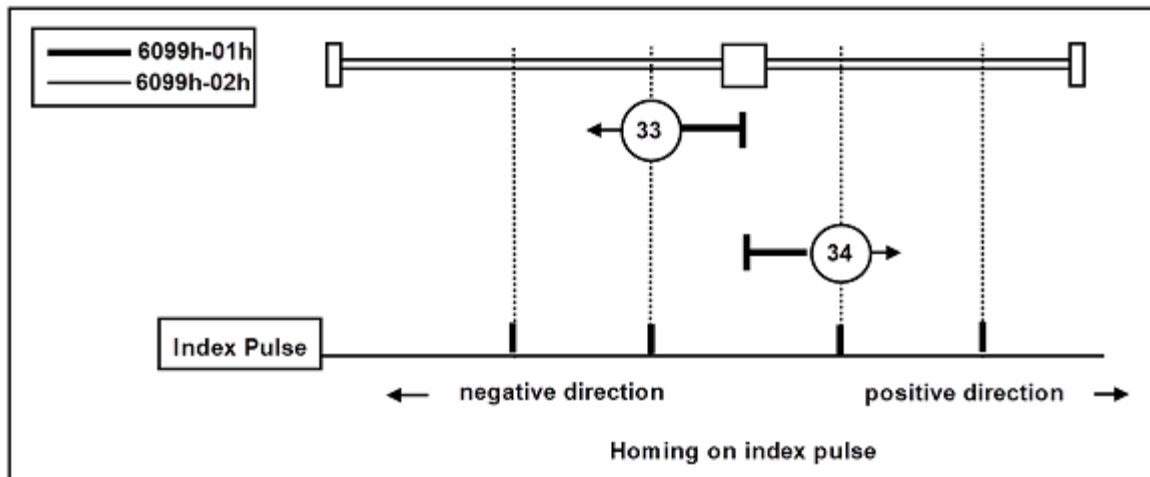
### 8.5.12. Method 27, 28, 29, 30

- 方法 29, 30 的原點開關如果在動作開始時已經激活，初始動作方向為負方向。
- 方法 27, 28 的原點開關如果在動作開始時已經激活，初始動作方向為正方向。
- 原點檢出位置是原點開關的狀態變化時的位置。



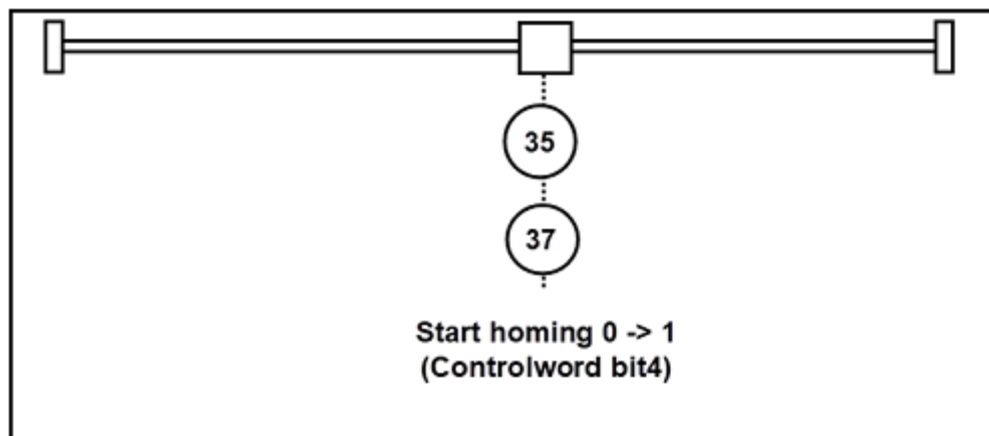
### 8.5.13. Method 33, 34

- 在圖中所示方向動作後檢出 Index pulse 作為原點檢出位置。



## 8.5.14. Method 35, 37

- 原點檢出位置是當下的位置。



## 8.6. CiA402 馬達每轉之解析度與電子齒輪比

### 8.6.1. 驅動器內部參數

注:以下物件非所有驅動器皆支援

注:以下參數需存入 EEPROM 且重啟電源後生效，請參閱 8.8 CiA402 驅動器 EEPROM 存檔

SDO Index	Sub-Index	Data Size	說明
0x608F	0x01	4 byte	Description
0x6091	0x01	4 byte	電子齒輪比分子
0x6091	0x02	4 byte	電子齒輪比分母

例 1: 編碼器分辨率為 8388608 pulse，欲設定為馬達每轉解析度為 100000 pulse

$$\frac{\text{電子齒輪比分子}}{\text{電子齒輪比分母}} = \frac{8388608}{100000}$$

例 2: 編碼器分辨率為 8388608 pulse，滾珠螺桿導程為 5mm，欲設定每個 pulse 的移動量為 1μm

軸旋轉一圈的移動量 = 5mm = 5000μm

$$\frac{\text{電子齒輪比分子}}{\text{電子齒輪比分母}} = \frac{8388608}{5000}$$

## 8.6.2. EtherCAT 主站參數

請參閱 7.2.4 ECAT\_McSetAxisPPU



## 8.7. CiA402 驅動器馬達運轉方向設定

注:此物件非所有驅動器皆支援

注:此參數需存入 EEPROM 且重啟電源後生效，請參閱 8.8 CiA402 驅動器 EEPROM 存檔

SDO Index	Sub-Index	Data Size	說明
0x607E	0x00	1 byte	極性
	bit7: 位置極性 0: 無反轉 1: 有反轉 bit6: 速度極性 0: 無反轉 1: 有反轉 bit5: 轉矩極性 0: 無反轉 1: 有反轉 bit0~4: reserve  必須將位置、速度與轉矩極性設至完全相同為 0(bit5~7 皆為 0)或者設定為 224(bit5~7 皆為 1)		

## 8.8. CiA402 驅動器 EEPROM 存檔

注:此物件非所有驅動器皆支援

SDO Index	Sub-Index	Data Size	說明
0x1010	0x01	4 byte	參數存檔
	寫入 Hex: 65766173h (Dec: 1702257011) , 將參數存檔		



## 8.9. CiA402 驅動器重要參數

### 8.9.1. 減速度

Sdo Index	Sub Index	Name	用途
0x6084	0x00	Profile deceleration	EMG 時用此減速度停止馬達 請設為-1(0xFFFFFFFF)
0x6085	0x00	Quick stop deceleration	碰到極限時用此減速度停止馬達 請設為-1(0xFFFFFFFF)

## 8.10. ECAT-2091S/ ECAT-2094S 使用注意事項

### 8.10.1. 六線式馬達

馬達 COM 點不需要接，不能把 2 個 COM 接在一起，會造成短路，需要把兩個 COM 點分別空接。

## 8.10.2. 重要參數

以下的 Sdo Index 的 n

對於 ECAT-2091S n = 0

對於 ECAT-2094S n = 0 ~ 3(共 4 軸)

Sdo Index	Sub Index	Name	用途
0x8n10	0x01	Maximun run current	最大運行電流，單位: mA 當高速運轉會失步時，請加大此值  當激磁後且運轉時，馬達使用此電流
0x8n10	0x03	Maximun hold current	馬達靜止電流，單位: mA  當激磁後且未運轉時，馬達使用此電流
0x8n10	0x08	Power on motor current	當馬達未激磁時，使用此電流，單位: mA
0x8n10	0x07	Micro Steps	微步進分割數， 預設為 8(256 微步)， 當使用 1 轉為 200 步的馬達時， 實際馬達 1 轉為 $200 * 256$ 微步 = 51200 微步 當速度不夠快時，請減少分割數 建議高速軸設此值為 5(32 微步) 實際馬達 1 轉為 $200 * 32 = 6400$ 微步
0x8n12	0x30 0x31	Invert Digital input	RL/LL 訊號反向與否

0x8n12	0x32 0x36	Function for input	是否啟用硬體停止功能 預設為否
0x8n20	0x02	Velocity Max	最大速度，單位：微步/秒 當速度不夠快，可能是已達 到此限制 請加大此值
0x8n30	0x01	GCONF	若馬達有異音問題 請設此值為 4 否則請保持 0
0x8n30	0x07	TPWMTHRS	若馬達有異音問題 請設此值為 6000 否則請保持 0