EPSON

Epson RC+ 8.0選配 RC+ API 8.0

翻譯版

© Seiko Epson Corporation 2024

Rev.2 TCM24ZS6922B

目錄

1. 前言	
1.1 前言	
1.2 商標	
1.3 本手冊中的商標符號	
1.4 聲明	
1.5 製造商	
1.6 諮詢服務	
1.7 閱讀本手冊之前	
2. 簡介	21
2.1 功能	
3. 安裝	
3.1 逐步說明	
3.2 安裝項目	
4. 開始使用	
4.1 開始使用	
4.2 開始使用Visual Basic	
4.3 開始使用Visual C#	
4.4 開始使用Visual C++	
5. 環境	
5.1 開發環境	
5.1.1 著手開發	
5.1.2 Spel類別執行個體初始化	
5.1.3 Spel類別執行個體終止	
5.1.4 開發週期	
5.2 於生產設備	
5.2.1 在運行時刻開啟Epson RC+ 8.0	
5.2.2 使用Epson RC+ 8.0對話方塊及視窗	
5.2.3 在目標系統上安裝	

D	2
Rev.	/
	-

6. 執行方法、程式、任務	
6.1 執行方法	
6.1.1 使用多執行緒	
6.2 執行SPEL+程式	
6.3 執行SPEL+程式	
6.4 終止所有任務	
7.事件	
7.1 概述	
7.2 系統事件	
7.3 來自SPEL+的使用者事件	
8. 錯誤處理	
8.1 Spel方法的錯誤	
9. 處理暫停和繼續	
9.1 暫停狀態	
9.2 攔截Pause事件	
9.3 執行暫停	
9.4 暫停後繼續	
9.5 暫停後終止	
10. 處理緊急停止	
10.1 使用系統EStop事件	
11. EpsonRC+8.0視窗及對話方塊	
11.1 視窗	53
11.2 對話方塊	
12. 顯示視訊	
12.1 顯示視訊	
12.2 使用多視訊顯示	
13. 使用AsyncMode	
13.1 使用AsyncMode	
14. SPELCom_Event	
14.1 SPELCom_Event	

15. RCAPINet參考	
15.1 Spel類別	
15.2 Spel類別屬性	
15.2.1 AsyncMode屬性.Spel類別	
15.2.2 AvoidSingularity屬性,Spel類別	
15.2.3 CommandInCycle屬性·Spel類別	
15.2.4 CommandTask屬性·Spel類別	
15.2.5 DisableMsgDispatch屬性,Spel類別	
15.2.6 ErrorCode屬性·Spel類別	
15.2.7 ErrorOn屬性 · Spel類別	
15.2.8 EStopOn屬性·Spel類別	
15.2.9 MotorsOn屬性·Spel類別	
15.2.10 NoProjectSync屬性·Spel類別	
15.2.11 OperationMode屬性 · Spel類別	
15.2.12 ParentWindowHandle屬性,Spel類別	
15.2.13 PauseOn屬性 · Spel類別	
15.2.14 PowerHigh屬性·Spel類別	
15.2.15 Project屬性,Spel類別	
15.2.16 ProjectBuildComplete屬性·Spel類別	
15.2.17 ProjectOverwriteWarningEnabled屬性, Spel 類別	
15.2.18 ResetAbortEnabled屬性 · Spel類別	
15.2.19 Robot屬性 · Spel類別	
15.2.20 RobotModel屬性·Spel類別	
15.2.21 RobotType屬性·Spel類別	
15.2.22 SafetyOn屬性·Spel類別	
15.2.23 ServerInstance屬性·Spel類別	
15.2.24 SPELVideoControl屬性 · Spel類別	
15.2.25 Version屬性·Spel類別	
15.2.26 WarningCode屬性.Spel類別	
15.2.27 WarningOn屬性·Spel類別	
15.3 Spel類別方法	
15.3.1 Accel方法,Spel類別	
15.3.2 AccelR方法,Spel類別	
15.3.3 AccelS方法,Spel類別	
15.3.4 Agl方法 · Spel類別	

15.3.5 AIO_In 方法, Spel類別	
15.3.6 AIO_InW 方法, Spel類別	
15.3.7 AIO_Out 方法, Spel類別	
15.3.8 AIO_OutW 方法, Spel類別	102
15.3.9 Arc方法 · Spel類別	103
15.3.10 Arc3方法 · Spel類別	104
15.3.11 Arch方法 · Spel類別	105
15.3.12 Arm方法 · Spel類別	
15.3.13 ArmClr方法 · Spel類別	
15.3.14 ArmDef方法 · Spel類別	
15.3.15 ArmSet方法 · Spel 類別	109
15.3.16 Atan方法 · Spel類別	
15.3.17 Atan2方法 · Spel類別	111
15.3.18 ATCLR 方法 · Spel類別	112
15.3.19 AtHome方法 · Spel類別	113
15.3.20 ATRQ 方法 · Spel類別	114
15.3.21 AvgSpeed 方法 · Spel類別	115
15.3.22 AvgSpeedClear 方法 · Spel類別	
15.3.23 AxisLocked方法 · Spel類別	117
15.3.24 Base方法 · Spel類別	
15.3.25 BGo方法 · Spel類別	
15.3.26 BMove方法 · Spel類別	121
15.3.27 Box方法 · Spel類別	
15.3.28 BoxClr方法 · Spel類別	
15.3.29 BoxDef方法 · Spel類別	
15.3.30 Brake 方法 · Spel類別	
15.3.31 BTst方法 · Spel類別	
15.3.32 BuildProject方法 · Spel類別	
15.3.33 Call方法 · Spel類別	
15.3.34 CalPls 方法·Spel類別	
15.3.35 ClearPoints方法 · Spel類別	
15.3.36 Connect方法 · Spel類別	
15.3.37 Continue方法, Spel類別	
15.3.38 Ctr方法 · Spel 類別	
15.3.39 CtReset方法,Spel類別	

15.3.40 Curve方法 · Spel類別	
15.3.41 CVMove方法 · Spel類別	
15.3.42 CX、CY、CZ、CU、CV、CW、CR、CS、CT方法.Spel類別	
15.3.43 Delay方法 · Spel類別	
15.3.44 DegToRad方法 · Spel類別	
15.3.45 Disconnect方法 · Spel類別	
15.3.46 ECP方法 · Spel類別	
15.3.47 ECPCIr方法 · Spel類別	
15.3.48 ECPDef方法 · Spel類別	
15.3.49 ECPSet方法 · Spel類別	
15.3.50 EnableEvent方法 · Spel類別	
15.3.51 ExecuteCommand方法 · Spel類別	
15.3.52 FBusIO_GetBusStatus 方法 · Spel類別	
15.3.53 FBusIO_GetDeviceStatus 方法 · Spel類別	
15.3.54 FBusIO_SendMsg 方法 · Spel類別	
15.3.55 FGGet 方法 · Spel類別	
15.3.56 FGRun 方法,Spel類別	
15.3.57 Find方法 · Spel類別	
15.3.58 Fine方法 · Spel類別	
15.3.59 GetAccel方法 · Spel類別	
15.3.60 GetArm方法 · Spel類別	
15.3.61 GetConnectionInfo方法 · Spel類別	
15.3.62 GetControllerInfo方法 · Spel類別	
15.3.63 GetCurrentConnectionInfo 方法 · Spel類別	
15.3.64 GetCurrentUser方法 · Spel類別	
15.3.65 GetECP方法 · Spel類別	
15.3.66 GetErrorMessage方法 · Spel類別	
15.3.67 GetIODef方法 · Spel類別	
15.3.68 GetJRange 方法 · Spel類別	
15.3.69 GetLimitTorque方法 · Spel類別	170
15.3.70 GetLimZ方法 · Spel類別	171
15.3.71 GetPoint方法, Spel類別	
15.3.72 GetRealTorque方法 · Spel類別	173
15.3.73 GetRobotInfo 方法,Spel類別	174
15.3.74 GetRobotPos方法, Spel類別	175

15.3.75 GetSpeed方法 · Spel類別	
15.3.76 GetTaskInfo 方法·Spel類別	
15.3.77 GetTool方法 · Spel類別	
15.3.78 GetVar方法 · Spel類別	
15.3.79 Go方法 · Spel類別	
15.3.80 Halt方法 · Spel類別	
15.3.81 Here方法 · Spel類別	183
15.3.82 HideWindow方法·Spel類別	
15.3.83 Home方法 · Spel類別	
15.3.84 Hofs 方法 · Spel類別	186
15.3.85 HomeSet方法 · Spel類別	
15.3.86 Hordr方法 · Spel類別	
15.3.87 Hour方法 · Spel類別	189
15.3.88 ImportPoints方法 · Spel類別	
15.3.89 In方法 · Spel類別	
15.3.90 InBCD方法 · Spel類別	
15.3.91 Inertia方法 · Spel類別	193
15.3.92 Initialize方法 · Spel類別	
15.3.93 InReal 方法 · Spel類別	195
15.3.94 InsideBox方法 · Spel類別	
15.3.95 InsidePlane方法·Spel類別	
15.3.96 InW方法 · Spel類別	
15.3.97 IsOptionActive方法·Spel類別	
15.3.98 JRange方法 · Spel類別	
15.3.99 JS方法 · Spel類別	
15.3.100 JTran方法 · Spel類別	
15.3.101 Jump方法 · Spel類別	
15.3.102 Jump3方法·Spel類別	
15.3.103 Jump3CP方法 · Spel類別	
15.3.104 LimitTorque方法 · Spel類別	
15.3.105 LimZ方法 · Spel類別	
15.3.106 LoadPoints方法·Spel類別	
15.3.107 Local方法 · Spel類別	
15.3.108 LocalClr方法, Spel類別	
15.3.109 LocalDef方法 · Spel類別	

1531101 ogin方法,Spel	215
15.3.110 LOGIII7.7次 Spei7条/	215
15.3.112 MCalComplete方法,Spel類別	210
153113 Mcordr方法·Spel類別	217
15.3.114 MemIn方法,Snel類別	210
15.3.114 MemInW方法,Spel類別	219
15.3.116 MemOff方法,Spel類別	220
15.3.117 MemOn方注,Spel短则	221
15.3.117 Wellion为公,Spel频加	222
15.3.119 MemOutW方法,Spel類別	223
15.3.120 MemSw方法,Snel類別	224
15.3.120 Wellisw为公、Spel展示	225
15.3.121 WOVE分位,Spel展加	220
15.3.122 OII/1/公·SpEi规则	220
15.3.123 OLIVATE 方法,Spel预加	229
15.3.124 OID 方法,Spel 规则	230
15.3.125 OppCD/1/公·Spet规则	251 222
15.3.120 Op0107/2 · Spei来加	
15.3.127 Outping,Sper展加	233
15.3.128 Outreal 万公,Sper规则	234 235
15.3.129 Outwy/元,Sper规则	235
15.3.130 PAGI万法,Spel)规则	230 227
15.2.122 Pace古注,Spel)類別	237
15.3.132 Fass/元、Spel规则	238
15.3.135 Pause万法,Spel)规则	239
15.3.134 PDel方法, Spel规则	
15.5.155 PDel/方法, Spel/短周	
15.3.136 PeakSpeed 万法·Spel 规则	
15.3.137 Peakspeed cleal 万法,spel _舆 別	
15.3.138 PF_Abort 万法·Spel)则	
15.3.139 PF_Backlight 方法·Spel 契闭	
15.3.140 PF_BacklightBrightness 方法,Spel 類別	246
15.3.141 PF_Name万法·Spel 類別	
15.3.142 PF_Number方法, Spel類別	
15.3.143 PF_Start方法, Spel類別	
15.3.144 PF_Stop方法, Spel類別	

15.3.145 PLabel 方法 · Spel類別	
15.3.146 Plane方法 · Spel類別	
15.3.147 PlaneClr方法 · Spel類別	
15.3.148 PlaneDef方法 · Spel 類別	
15.3.149 Pls方法 · Spel類別	
15.3.150 PTCLR方法 · Spel類別	
15.3.151 PTPBoost方法 · Spel類別	
15.3.152 PTPBoostOK方法 · Spel類別	
15.3.153 PTran方法 · Spel類別	
15.3.154 PTRQ方法·Spel類別	
15.3.155 Pulse方法·Spel類別	
15.3.156 Quit方法 · Spel類別	
15.3.157 RadToDeg方法 · Spel類別	
15.3.158 RebootController方法 · Spel類別	
15.3.159 RebuildProject方法·Spel類別	265
15.3.160 Recover方法 · Spel類別	
15.3.161 Reset方法 · Spel類別	
15.3.162 ResetAbort方法 · Spel類別	
15.3.163 Resume方法·Spel類別	270
15.3.164 RunDialog方法 · Spel類別	
15.3.165 SavePoints方法 · Spel類別	
15.3.166 Sense方法·Spel類別	
15.3.167 SetIODef方法 · SpeI類別	
15.3.168 SetPoint方法 · Spel類別	
15.3.169 SetVar方法·Spel類別	278
15.3.170 SFree方法 · Spel類別	
15.3.171 ShowWindow方法 · Spel類別	
15.3.172 SimGet 方法·Spel類別	
15.3.173 SimResetCollision方法 · Spel類別	
15.3.174 SimSet 方法 · Spel類別	
15.3.175 SimSetParent 方法 · Spel類別	284
15.3.176 SimSetPick 方法 · Spel類別	
15.3.177 SimSetPlace 方法,Spel類別	
15.3.178 Shutdown方法, Spel類別	
15.3.179 SLock方法,Spel類別	

15.3.180 Speed方法 · Spel類別	
15.3.181 SpeedR方法 · Spel類別	
15.3.182 SpeedS方法 · Spel類別	
15.3.183 Start方法 · Spel類別	
15.3.184 StartBGTask方法,Spel類別	
15.3.185 Stat方法 · Spel類別	
15.3.186 Stop方法 · Spel類別	
15.3.187 Sw方法 · Spel類別	
15.3.188 TargetOK方法 · Spel類別	
15.3.189 TasksExecuting方法 · Spel類別	
15.3.190 TaskState方法 · Spel類別	
15.3.191 TeachPoint方法·Spel類別	
15.3.192 Till方法 · Spel類別	
15.3.193 TillOn方法 · Spel類別	
15.3.194 TGo方法 · Spel類別	
15.3.195 TLCIr方法 · Spel類別	
15.3.196 TLDef方法 · Spel類別	
15.3.197 TLSet方法 · Spel類別	
15.3.198 TMove方法 · Spel類別	
15.3.199 Tool方法 · Spel類別	
15.3.200 TrapStop方法 · Spel類別	
15.3.201 TW方法 · Spel類別	
15.3.202 UserHasRight方法 · Spel類別	
15.3.203 VCal方法 · Spel類別	
15.3.204 VCalPoints方法 · Spel類別	
15.3.205 VCls方法 · Spel類別	
15.3.206 VCreateCalibration方法 · Spel類別	
15.3.207 VCreateObject方法 · Spel類別	
15.3.208 VCreateSequence方法 · Spel類別	
15.3.209 VDefArm方法 · Spel類別	
15.3.210 VDefGetMotionRange方法·Spel類別	
15.3.211 VDefLocal方法,Spel類別	
15.3.212 VDefSetMotionRange方法,Spel類別	
15.3.213 VDefTool方法 · Spel類別	
15.3.214 VDefToolXYZ 方法, Spel 類別	

15.3.215 VDefToolXYZUVW 方法, Spel 類別	
15.3.216 VDeleteCalibration方法 · Spel類別	
15.3.217 VDeleteObject方法 · Spel類別	
15.3.218 VDeleteSequence方法 · Spel類別	
15.3.219 VEditWindow 方法, Spel類別	
15.3.220 VGet方法 · Spel類別	
15.3.221 VGetCameraXYU方法 · Spel類別	
15.3.222 VGetEdgeCameraXYU方法 · Spel類別	
15.3.223 VGetEdgePixelXYU方法 · Spel類別	
15.3.224 VGetEdgeRobotXYU方法 · Spel類別	
15.3.225 VGetExtrema方法 · Spel類別	
15.3.226 VGetModelWin方法 · Spel類別	
15.3.227 VGetPixelToCamera 方法, Spel 類別	
15.3.228 VGetPixelToRobot 方法, Spel 類別	
15.3.229 VGetPixelXYU方法·Spel類別	
15.3.230 VGetRobotPlacePos方法 · Spel類別	
15.3.231 VGetRobotPlaceTargetPos方法 · Spel類別	
15.3.232 VGetRobotXYU方法 · Spel類別	
15.3.233 VGetRobotToolXYU方法 · Spel類別	
15.3.234 VGetSearchWIn方法 · Spel類別	
15.3.235 VGoCenter方法 · Spel類別	
15.3.236 VLoad方法 · Spel類別	
15.3.237 VLoadModel方法·Spel類別	
15.3.238 VRun方法 · Spel類別	
15.3.239 VSave方法 · Spel類別	
15.3.240 VSaveImage方法 · SpeI類別	
15.3.241 VSaveModel方法·Spel類別	
15.3.242 VSet方法 · Spel類別	
15.3.243 VSetModelWin方法 · Spel類別	
15.3.244 VSetRobotPlaceTargetPos方法 · Spel類別	
15.3.245 VSetSearchWin方法 · Spel類別	
15.3.246 VShowModel方法,Spel類別	
15.3.247 VShowSequence方法, Spel類別	
15.3.248 VStatsReset方法,Spel類別	
15.3.249 VStatsResetAll方法 · Spel類別	

15.3.250 VStatsSave方法 · Spel類別	
15.3.251 VStatsShow方法 · Spel類別	
15.3.252 VTeach方法 · Spel類別	
15.3.253 VTrain方法 · Spel類別	
15.3.254 WaitCommandComplete方法 · Spel類別	
15.3.255 WaitMem方法 · Spel類別	
15.3.256 WaitSw方法 · Spel類別	
15.3.257 WaitTaskDone方法 · Spel類別	
15.3.258 Weight方法 · Spel類別	
15.3.259 Xqt方法·Spel類別	
15.3.260 XYLim方法 · Spel類別	
15.3.261 XYLimClr方法 · Spel類別	
15.3.262 XYLimDef方法 · Spel類別	
15.4 Spel類別事件	
15.4.1 EventReceived事件 · Spel類別	
15.5 SPELVideo控制項	
15.6 SPELVideo控制項屬性	
15.6.1 Camera屬性·SPELVideo控制項	
15.6.2 GraphicsEnabled屬性·SPELVideo控制項	
15.6.3 VideoEnabled屬性 · SPELVideo控制項	
15.7 SPELVideo控制項方法	
15.7.1 LoadImage方法 · SPELVideo控制項	
15.8 SPELVideo控制項事件	
15.9 SpelConnectionInfo類別	
15.10 SpelControllerInfo類別	
15.11 SpelException類別	
15.11.1 SpelException屬性	
15.11.2 SpelException方法	
15.12 SpelOptionInfo類別	
15.13 SpelPoint類別	
15.13.1 SpelPoint屬性	
15.13.2 SpelPoint方法	
15.14 SpelRobotInfo類別	
15.15 SpelTaskInfo類別	

15.16 列舉	
15.16.1 SpelArmDefMode列舉	
15.16.2 SpelArmDefType列舉	
15.16.3 SpelAxis列舉	
15.16.4 SpelBaseAlignment列舉	
15.16.5 SpelCalPlateType列舉	
15.16.6 SpelConnectionType列舉	
15.16.7 SpelDialogs列舉	
15.16.8 SpelElbow列舉	
15.16.9 SpelEvents列舉	
15.16.10 SpelForceCompareType列舉	
15.16.11 SpelForceProps列舉	
15.16.12 SpelHand列舉	
15.16.13 SpellOLabelTypes列舉	
15.16.14 SpelLocalDefType列舉	
15.16.15 SpelOperationMode列舉	
15.16.16 SpelOptions列舉	
15.16.17 SpelOptionStatus列舉	
15.16.18 SpelRobotPosType列舉	
15.16.19 SpelRobotType列舉	
15.16.20 SpelShutdownMode列舉	
15.16.21 SpelSimObjectType列舉	
15.16.22 SpelSimProps列舉	
15.16.23 SpelStopType列舉	
15.16.24 SpelTaskState列舉	
15.16.25 SpelTaskType列舉	
15.16.26 SpelToolDefType列舉	
15.16.27 SpelToolDefType3D列舉	
15.16.28 SpelUserRights列舉	
15.16.29 SpelVDefShowWarning列舉	
15.16.30 SpelVisionImageSize列舉	
15.16.31 SpelVisionObjectTypes列舉	
15.16.32 SpelVisionProps列舉	
15.16.33 SpelWindows列舉	
15.16.34 SpelWrist列舉	

oson RC+ 8.0選配 RC+ API 8.0	Rev.2
15.17 Spel錯誤編號及訊息	
16.32位元與64位元應用程式	
17. 使用LabVIEW VI程式庫	
17.1 概述	
17.2 安裝	
17.3 工具面板	
17.4 開始使用	
17.5 使用Spel+專案	
17.6 顯示視訊	
17.7 VI參考	
17.7.1 Accel VI	
17.7.2 AccelS VI	
17.7.3 Arc VI	
17.7.4 Arc3 VI	
17.7.5 Arch VI	
17.7.6 Arm VI	
17.7.7 Armset VI	
17.7.8 AtHome VI	
17.7.9 AvoidSing VI	
17.7.10 BGo VI	
17.7.11 BMove VI	
17.7.12 Box VI	
17.7.13 Continue VI	
17.7.14 Delay VI	
17.7.15 ECP VI	
17.7.16 ECPset VI	
17.7.17 EStopOn VI	

17.7.17 EStopOn VI	449
17.7.18 Find VI	450
17.7.19 Fine VI	451
17.7.20 GetArm VI	452
17.7.21 GetAvoidSing VI	453
17.7.22 GetECP VI	454
17.7.23 GetLimZ VI	455
17.7.24 GetMotor VI	456

17.7.25 GetOprMode VI	
17.7.26 GetPoint VI	
17.7.27 GetPower VI	
17.7.28 GetRobot VI	
17.7.29 GetTool VI	
17.7.30 GetVar VI	
17.7.31 Go VI	
17.7.32 Halt VI	
17.7.33 Here VI	
17.7.34 HideWindow VI	
17.7.35 In VI	
17.7.36 InBCD VI	
17.7.37 InsideBox VI	
17.7.38 InsidePlane VI	
17.7.39 InW VI	
17.7.40 Inertia VI	
17.7.41 Initialize VI	
17.7.42 JRange VI	
17.7.43 JS VI	
17.7.44 JTran VI	
17.7.45 Jump VI	
17.7.46 Jump3 VI	
17.7.47 Jump3CP VI	
17.7.48 LimZ VI	480
17.7.49 LoadPoints VI	
17.7.50 MemIn VI	
17.7.51 MemInW VI	
17.7.52 MemOut VI	
17.7.53 MemOff VI	
17.7.54 MemOn VI	
17.7.55 MemOut VI	
17.7.56 MemOutW VI	
17.7.57 MemSw VI	
17.7.58 MotorOff VI	
17.7.59 MotorOn VI	

D	1
Rev	/
	-

17.7.60 Move VI	
17.7.61 Off VI	
17.7.62 On VI	
17.7.63 OPort VI	
17.7.64 OprMode VI	
17.7.65 Out VI	
17.7.66 OutW VI	
17.7.67 Pause VI	
17.7.68 Plane VI	
17.7.69 PowerHigh VI	
17.7.70 PowerLow VI	
17.7.71 Quit VI	503
17.7.72 Reset VI	
17.7.73 Resume VI	
17.7.74 Robot VI	
17.7.75 RunDialog VI	
17.7.76 SafetyOn VI	
17.7.77 SavePoints VI	
17.7.78 Sense VI	
17.7.79 SetPoint VI	
17.7.80 SetVar VI	
17.7.81 SFree VI	
17.7.82 ShowWindow VI	
17.7.83 Shutdown VI	515
17.7.84 SLock VI	
17.7.85 Speed VI	517
17.7.86 SpeedS VI	
17.7.87 Start VI	
17.7.88 Stop VI	521
17.7.89 Sw VI	522
17.7.90 TargetOK VI	
17.7.91 TGo VI	
17.7.92 Till VI	
17.7.93 TillOn VI	
17.7.94 TLSet VI	

531
532
535

Rev.2

1. 前言

1.1 前言

感謝您選購本公司的機器人產品。 本手冊包含正確使用機器人所需的資訊。 在安裝機器人系統之前,請務必詳閱本手冊及其他相關手冊。 請將本手冊放在方便隨時取用的地方。

所有機器人系統與其選配部件經嚴格的品質控管、測試與檢驗,以確保其符合我們的高效能標準,始能出貨給貴客 戶。請注意,若未依本手冊說明的使用條件與產品規格使用本機器人系統,將無法發揮產品的基本性能。

本手冊說明我們可預測的可能危險及後果。務必遵守本手冊的安全注意事項,確保安全及正確地使用機器人系統。

1.2 商標

Microsoft、Windows及Windows標誌是Microsoft Corporation在美國及/或其他國家的註冊商標或商標。其他品牌 及產品名稱均為其各自持有人的商標或註冊商標。

1.3 本手冊中的商標符號

MicrosoftR WindowsR 10 operating system

MicrosoftR WindowsR 11 operating system

本手冊中的Windows 10和Windows 11分別指上述作業系統。在某些情況下, Windows通常是指Windows 10和 Windows 11。

1.4 聲明

未經授權,不得翻印或重製本手冊的任何內容。 本手冊內容如有變更,恕不另行通知。 如發現本手冊有任何錯誤或有對內容任何意見,歡迎與我們聯繫。

1.5 製造商

SEIKO EPSON CORPORATION

1.6 諮詢服務

如需詳細資訊,請參閱下列手冊的「供應商」。 「安全手冊」

1.7 閱讀本手冊之前

本節說明在閱讀本手冊前您應先瞭解的資訊。

關於Epson RC+ 8.0安裝資料夾

Epson RC+ 8.0 允許您將安裝資料夾路徑更改為任何位置。本手冊假定Epson RC+ 8.0 安裝在C:¥EpsonRC80上進行 說明。

2. 簡介

Epson RC+ 8.0選購件RC+ API可讓您使用支援.NET技術的Microsoft Visual Basic或其他語言來執行您的機器應用 程式。您可以創建複雜的用戶介面並使用資料庫。也可以使用.NET Framework或.NET 6,它們在設計時考慮了 .NET Framework。

也包含LabVIEW程式庫。

.NET6僅在「Visual Basic」或Visual C¥中可用。

.NET6僅在Visual Studio 2022中可用。

2.1 功能

RC+ API套件支援以下功能:

- .NET程式庫與LabVIEW程式庫。
- 支援32位元與64位元應用程式。
- 從多個控制器控制多個機器人、I/O及任務的屬性與方法。
- 執行視覺及力覺感測*命令的方法。
 * 力覺感測及力覺感測器並不相同。API手冊中所述之力覺感測的方法與屬性不適用於力覺感測器。若要使用力覺 感測器的命令,請使用Xpt方法執行SPEL函數。API不支援Epson RC+選購件Force Guide。
- 支援透過多執行緒並行執行非同步命令。
- 您可透過.NET應用程式使用多個Epson RC+ 8.0視窗及對話方塊,包括:
 - 機器人管理器
 - IO監視器
 - 任務管理器
 - 模擬器
 - 控制器工具對話方塊

在開發期間, Epson RC+ 8.0可以和Visual Basic一起執行。在生產設備中, Epson RC+ 8.0可隱藏在背景中執行。

▶ 提示

RC+ API 支援".NET Framework Library"與"LabVIEW VI Library"。不支援".NET Core"與".NET"5版本。 以Epson RC+ 8.0或更高版本製作的RC+ API應用程式,僅可在 Epson RC+ 8.0 使用。

以Epson RC+ 7.0 製作的專案及 RC+ 應用程式,可與 Epson RC+ 8.0 相容。想使用之前版本的專案時,需要在 Epson RC+ 8.0 上重建專案。

下圖顯示使用RC+ API之系統的基本結構。



Epson RC+ 8.0是RCAPINet程式庫的跨處理序伺服器。 RCAPINet Spel類別的每個執行個體會啟動一個Epson RC+ 8.0執行個體。

3. 安裝

請依照本章的說明,確保正確安裝RC+ API軟體。 在開始之前,請確認已關閉所有Windows應用程式。

3.1 逐步說明

- 安裝下述任一軟體。
 Visual Studio 2015, 2017, 2019, 2022
 (包括Enterprise、Professional、Community或Express Edition)
 LabVIEW 2009或更新版本
- 2. 安裝Epson RC+ 8.0。
- 3. 如欲使用LabVIEW, 應安裝LabVIEW VI程式庫。
- 4. 確認已在您要使用的控制器中啟用RC+ API軟體金鑰。 有關如何在控制器中啟用選購件的詳細資訊,請參閱Epson RC+ 8.0使用指南。

RC+API安裝程序到此結束。

3.2 安裝項目

下表顯示的目錄及檔案會在安裝過程中安裝至您的電腦。

目錄與檔案	描述
(Data folder)¥API¥VS20xx¥VB¥DEMOS	Visual Basic .NET展示
(Data folder)¥API¥VS20xx¥VCS¥DEMOS	Visual C# .NET展示
(Data folder)¥API¥VS20xx¥VC¥DEMOS	Visual C++ .NET展示
(Program folder)¥API¥LabVIEW	LabVIEW VI程式庫安裝程式
(Data folder)¥PROJECTS¥API_Demos	展示用的Epson RC+ 8.0專案
(Program folder)¥API¥Assembly¥net462¥RCAPINt2.dll	適用於.NET Framework的RCAPINet程式庫(32位 或64位)
(Program folder)¥API¥Assembly¥net6.0- windows¥RCAPINt2.dll	適用於.NET 6的RCAPINet程式庫(32位或64位)

4. 開始使用

4.1 開始使用

本章包含下列開發環境的開始使用資訊。

- Visual Basic .NET
- Visual C# .NET
- Visual C++ .NET

展示程式隨附於RC+ API。建議您透過展示程式來更加熟悉本產品。 有關目錄位置,請參閱以下內容。 安裝項目

若要使用LabVIEW,請參閱以下內容。 開始使用

第一次在Visual C++ 2017上建置展示程式時,程式建置可能失敗。若創建失敗,請參閱以下內容。 開始使用Visual C++

如果您使用.NET 應用程式,則.NET Framework的版本需要是.NET Framework v4.6.2 或更高版本,或者請.NET 6。

以下內容基於Visual Studio 2022中的操作。 如果您使用的是Visual Studio 2022以外的版本,請替換成相應操作。

4.2 開始使用Visual Basic

若要在Visual Basic .NET專案中使用RCAPINet,請宣告Spel類別執行個體,如以下範例所示。您可在專案中使用 g_spel。

- 1. 在Visual Studio .NET中, 選擇[File]-[New]-[Project]。
- 2. 請選擇以下種類的Visual Basic專案。
- .NET Framework : Windows表單應用程式 (.NET Framework)
- .NET6: Windows表單應用程式
- 3. 選擇功能表-[Project]-[Add Reference]。
- 4. 在[瀏覽]標籤上, 選擇「¥API」目錄下的「RCAPINt2.dll」檔。
- .NET Framework : ¥Assembly¥net462
- .NET6 : ¥Assembly¥net6.0-windows
- 5. 從[專案]功能表中, 建立新的模組並添加以下程式碼。

```
Module Module1
Public WithEvents g_spel As RCAPINet.Spel
Public Sub InitApp()
g_spel = New RCAPINet.Spel
With g_spel
.Initialize
.Project = "c:\EpsonRC80\projects\API_Demos\Demo1 \demo1.sprj"
End With
End Sub
Public Sub EventReceived(
    ByVal sender As Object,
    ByVal e As RCAPINet.SpelEventArgs)
Handles g_spel.EventReceived
MsgBox("received event " & e.Event)
```

End Sub End Module

▶ 提示

當應用程式結束時,您必須針對每個Spel類別執行個體執行Dispose。您可在主要表單的FormClosed事件中完成 此操作。若沒有執行Dispose,應用程式將無法正確關閉。

g_spel.Dispose()

4.3 開始使用Visual C#

1. 在Visual Studio .NET中, 選擇功能表-[File]-[New]-[Project]。

- 2. 請選擇以下種類的Visual C#專案。
- .NET Framework : Windows表單應用程式 (.NET Framework)
- .NET6: Windows表單應用程式
- 3. 選擇功能表-[Project]-[Add Reference]。
- 4. 在[瀏覽]標籤上,選擇「¥API」目錄下的「RCAPINt2.dll」檔。
- .NET Framework : ¥Assembly¥net462
- .NET6 : ¥Assembly¥net6.0-windows
- 5. 在Form1類別中宣告Spel類別變數,如下所示。

private RCAPINet.Spel m_spel;

6. 在Form_Load事件中添加初始化程式碼,如下所示。

```
private void Form1_Load(object sender, EventArgs e)
{
    m_spel = new RCAPINet.Spel();
    m_spel.Initialize();
    m_spel.Project =
    "c:\\EpsonRC80\\projects\\API_Demos\\Demo1\\demo1.sprj";

    m_spel.EventReceived += new
    RCAPINet.Spel.EventReceivedEventHandler(m spel EventReceived);
```

7. 添加事件處理常式,如下所示。

✔ 提示 當應用程式結束時,您必須針對每個Spel類別執行個體執行Dispose。您可在主要表單的FormClosed事件中完成 此操作。若沒有執行Dispose,應用程式將無法正確關閉。

m_spel.Dispose();

4.4 開始使用Visual C++

- 1. 在Visual Studio .NET中, 選擇[Create a new project]。
- 2. 選擇 [C++]-[Windows]-[CLR Empty Project (.NET Framework)]。

Create a new project	Search for templates (Alt+S)	- م	<u>C</u> lear all
<u>R</u> ecent project templates	C++ •	All <u>p</u> latforms • All pr	oject <u>t</u> ypes •
List of your recently accessed templates will be lisplayed here.	Dynamic-Link Library (DL Build a .dll that can be shi C++ Library Wir ClR Empty Project (.NET C C++ roject that has no s interoperability between . C++ Library Wir	L) ared between multiple running Windows ap ndows Core) tarting files and targets the .NET Core. Prov .NET and C++ code. ndows	ops. ides
	CLR Empty Project (.NET F C++ project that has no s interoperability between . Console C++ W	Framework) tarting files and targets the .NET framework .NET and C++ code. indows	ι. Provides
	CLR Console App (.NET Fr Run C++ code in a Windd interoperability between .	amework) ows terminal targeting the .NET framework. NET and C++ code. Indows	Provides
	Build a .lib that can be pa	ckaged inside other Windows executables.	
		<u>B</u> ack	Next

3. 選擇功能表-[Project]-[Add Reference]。

- 4. 在[瀏覽]標籤上, 選擇「¥API」目錄下的「RCAPINt2.dll」檔。
 - .NET Framework : ¥Assembly¥net462
 - .NET6 : ¥Assembly¥net6.0-windows
- 5. 選擇功能表-[Project]-[Add New Item]-[UI]-[Windows Form]。
- 6. 打開添加的表單的cpp檔案(例: Form1.cpp), 然後添加以下原始碼。

```
#include "Form1.h"
using namespace SampleProject; // 建立的專案名稱
void main() {
   Application EnableVisualStyles();
   Application SetCompatibleTextRenderingDefault(false);
   Form1 frm; // 新增的表單名稱
```

}

7. 選擇功能表- [Project]-[Project Properties]。

Application Run(% frm);

8. 在屬性頁選擇 [Configuration Properties]-[Linker]-[System], 然後從子系統中選擇「Windows (/SUBSYSTEM:WINDOWS)」。

SampleProject Property Pages		? ×
Configuration: Active(Debug)	✓ Platform:	Win32 ~ Configuration Manager
 ▲ Configuration Properties ▲ General Advanced Debugging VC++ Directories ▷ C/C++ ▲ Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Metadata Advanced All Options Command Line ▷ Manifest Tool ▷ XML Document Genera 	SubSystem Minimum Required Version Heap Reserve Size Heap Commit Size Stack Reserve Size Stack Commit Size Enable Large Addresses Terminal Server Swap Run From CD Swap Run From Network Driver	Windows (/SUBSYSTEM:WINDOWS) Not Set Console (/SUBSYSTEM:CONSOLE) Windows (/SUBSYSTEM:WINDOWS) Native (/SUBSYSTEM:NATIVE) EFI Application (/SUBSYSTEM:EFI_APPLICATION) EFI Boot Service Driver (/SUBSYSTEM:EFI_BOOT_SERVICE_DRIVER) EFI ROM (/SUBSYSTEM:EFI_ROM) EFI Runtime (/SUBSYSTEM:EFI_ROM) EFI Runtime (/SUBSYSTEM:EFI_ROM) EFI Runtime (/SUBSYSTEM:EFI_ROM) efi rem (rom parent or project defaults>
Build Events	SubSystem The /SUBSYSTEM option tells the affects the entry point symbol (o	e operating system how to run the .exe file.The choice of subsystem or entry point function) that the linker will choose.
		OK キャンセル 適用(A)

9. 在屬性頁選擇[Configuration Properties]-[Linker]-[Advanced], 然後在「Entry Point」中輸入步驟(6)添加的函 數名稱。在此範例中, 鍵入 "main"。

SampleProject Pro	perty Pages						?	×
<u>C</u> onfiguration:	Active(Debug)	✓ <u>P</u> latform:	Win32		~	C <u>o</u> nfiguratio	n Manag	er
 ✓ Configuration General Advanced Debuggir VC++ Dir ▷ C/C++ ▲ Linker General Input Manif Debug System Optim Embe Winde Advar All Op Comm ▷ Manifest ▷ XML Docc ▷ Browse Ir ▷ Build Eve 	n Properties	Entry Point No Entry Point Set Checksum Base Address Randomized Base Address Fixed Base Address Data Execution Prevention (Turn Off Assembly Generation Unload delay loaded DLL Nobind delay loaded DLL Import Library Merge Sections Target Machine Profile CLR Thread Attribute CLR Image Type Key File Key Container Delav Sion	DEP) on	main No No Yes (/DYNAMICBASE) No (/FIXED:NO) Yes (/NXCOMPAT) No MachineX86 (/MACHINE No Default image type function as the starting action	:X86)	r an .exe file or	· DLL.	
				OK	+	ャンセル	適用(<u>A</u>)

10. 按一下[OK]按鈕。



11. 在Form1類別中宣告Spel變數,如下所示。

```
private:RCAPINet::Spel^ m_spel;
```

12. 在Form_Load事件中添加初始化程式碼,如下所示。

```
private:System::Void Form1_Load(
   System::Object^ sender, System::EventArgs^ e)
{
   m_spel = gcnew RCAPINet::Spel();
   m_spel->Initialize();
   m_spel->Project =
        "c:\\EpsonRC80\\projects\\ API_Demos\\Demo1\\demo1.sprj";
   m_spel->EventReceived += gcnew
        RCAPINet::Spel::EventReceivedEventHandler(
        this, &Form1::m_spel_EventReceived);
}
```

13. 添加事件處理常式,如下所示。

```
private:System::Void m_spel_EventReceived(
   System::Object^ sender, RCAPINet::SpelEventArgs^ e)
{
   MessageBox::Show(e->Message);
}
```

♪ 提示

當應用程式結束時·若Spel類別執行個體是在堆積上分配(使用gcnew)·則您必須刪除每個Spel類別執行個體。您可在主要表單的FormClosed事件中完成此操作。若沒有刪除Spel類別執行個體,應用程式將無法正確關閉。

delete m_spel;

5. 環境

5.1 開發環境

5.1.1 著手開發

一般而言, 您會執行下列步驟開始進行開發:

- 1. 在.NET專案的模組中宣告Spel類別變數。
- 2. 啟動Epson RC+。
- 3. 開啟所需的Epson RC+專案,或建立新的Epson RC+專案。
- 4. 建置Epson RC+ 專案。
- 5. 針對Spel類別執行個體,添加初始化程式碼。
- 6. 執行並偵錯.NET專案。

5.1.2 Spel類別執行個體初始化

建立Spel類別的新建執行個體後,必須予以初始化。執行初始化時,會載入並初始化基本Epson RC+ 8.0模組。在第一個方法調用或屬性存取的情況下,初始化為隱含狀態。您可透過調用Initialize方法將類別初始化。

m_spel.Initialize()

5.1.3 Spel類別執行個體終止

當應用程式結束時,您必須針對每個Spel類別執行個體執行Dispose。您可在主要表單的FormClosed事件中完成此操作。若沒有執行Dispose,應用程式將無法正確關閉。 對於Visual Basic與Visual C#,請使用Dispose方法:

m_spel.Dispose()

對於Visual C++,若Spel類別執行個體是在堆積上建立(使用gcnew),則使用delete:

delete m_spel;

5.1.4 開發週期

請依照下列基本步驟編輯及執行.NET程式碼:

- 1. 停止.NET專案。
- 2. 編輯.NET專案。
- 3. 開啟Epson RC+ 8.0。
- 4. 變更Epson RC+ 8.0專案。
- 5. 建置Epson RC+ 8.0專案。
- 6. 關閉Epson RC+ 8.0。
- 7. 切換至Visual Studio。
- 8. 執行.NET專案。

5.2 於生產設備

5.2.1 在運行時刻開啟Epson RC+ 8.0

決定是否要從您的應用程式開啟Epson RC+ 8.0環境。此在偵錯時特別實用。將OperationMode屬性設至Program, 讓Epson RC+ 8.0進入Program模式,並開啟Epson RC+ 8.0 GUI。

5.2.2 使用Epson RC+ 8.0 對話方塊及視窗

您可從.NET應用程式開啟及隱藏特定的Epson RC+8.0視窗。您也可以執行特定的Epson RC+8.0對話方塊。有關詳細資訊,請參閱以下內容。

EpsonRC+8.0視窗及對話方塊

5.2.3 在目標系統上安裝

您應使用Visual Studio安裝專案對.NET專案進行程式安裝,然後依下列步驟設定.NET應用程式的目標系統:

- 1. 安裝Epson RC+ 8.0。
- 2. 安裝Epson RC+ 8.0專案。
- 3. 安裝.NET應用程式。

6. 執行方法、程式、任務
6.1 執行方法

Spel類別具有多種方法。有關的可用方法,請參閱以下內容。

Spel類別方法

執行方法時, 關聯的內部函數會在Epson RC+伺服器程序中調用, 接著與控制器進行通信以執行關聯的函數。方法 有以下兩種:立即和非同步。使用立即方法時, 內部函數會在控制器中執行並且會立即傳回回覆。立即命令包含所有 I/O命令。使用非同步方法時, 關聯的函數會在控制器中啟動, 然後Spel類別執行個體會等待來自Epson RC+伺服器 程序的事件指出函數已完成。非同步方法包含所有機器人動作命令。當等待命令完成時, Spel類別執行個體會發送 Windows事件, 讓使用者GUI保持回應。例如:當調用Go方法時, 機器人會移至指定點, 使用者可能會想按一下按 鈕來停止移動。您可將DisableMsgDispatch設為True, 即可在執行非同步方法時停用Windows事件發送。您也可以 將AsyncMode設為True, 等待非同步方法在程式中完成。

6.1.1 使用多執行緒

您可在應用程式中以多執行緒執行Spel方法。下列章節說明各種案例。

一個Spel類別執行個體用於多執行緒

您可在多執行緒中使用相同的Spel類別執行個體來執行方法,但一次只能執行一個非同步命令。如果您嘗試在一個執 行緒中執行非同步命令,但同時已經有另一個非同步命令在其他執行緒中執行,將會發生「命令循環中」錯誤。當在 其他執行緒中執行非同步命令時,您可執行立即命令。

獨立Spel類別執行個體用於每個執行緒

對於每個控制器連線,您可以有一或多個Spel類別執行個體。每個控制器的第一個執行個體會初始化Epson RC+ 8.0 伺服器程序,並連接至指定的控制器。若要在其他執行緒使用一或多個其他執行個體和相同的控制器進行通信,您必須將ServerInstance屬性指定為相同的值。在使用其他Spel類別執行個體前,您需要為第一個執行個體調用Initialize。

VB 例:

```
' 初始化執行緒1的Spel類別執行個體
m_spel_1 = New Spel
m_spel_1.ServerInstance = 1
m_spel_1.Initialize()
m_spel_1.Project = "c:\EpsonRC80\Projects\MyProject\MyProject.sprj"
m_spel_1.Connect(1)
```

· 初始化執行緒2的spe1類別執行個體

```
' 此執行個體使用與m spel 1相同的控制器
```

```
m_spel_2 = New Spel
m_spel_2.ServerInstance = 1
```

執行緒1

```
' 將m_spel_1執行個體用於動作
m_spel_1.Robot = 1
Do
m_spel_1.Go(1)
m_spel_1.Go(2)
Loop Until m_stop
執行緒2
' 將m_spel_2執行個體用於I/O
Do
m_spel_2.On(1)
m_spel_2.Oelay(500)
m_spel_2.Off(1)
m_spel_2.Delay(500)
Loop Until m_stop
```

C# 例:

```
// 初始化執行緒1的spe1類別執行個體
RCAPINet.Spel m spel 1 = new RCAPINet.Spel();
m spel 1.ServerInstance = 1;
m spel 1.Initialize();
m spel 1.Project = @"c:\EpsonRC80\Projects\MyProject\MyProject.sprj";
m spel 1.Connect(1);
// 初始化執行緒2的spel類別執行個體
// 此執行個體使用與m spel 1相同的控制器
RCAPINet.Spel m_spel_2 = new RCAPINet.Spel();
m spel 2.ServerInstance = 1;
執行緒1
// 將m spel 1執行個體用於動作
m spel 1.Robot = 1;
do {
m spel 1.Go(1);
m spel 1.Go(2);
}while(\!m stop);
執行緒2
// 將m spel 2執行個體用於I/O
do {
m spel 2.On(1);
m_spel_2.Delay(500);
m spel 2.0ff(1);
m_spel_2.Delay(500);
}while(\!m stop);
```

在控制器中使用API執行緒

根據預設,控制器僅支援一個API執行緒。在此範例中,即使控制多個機器人,在控制器中一次只執行一個非同步方法。對於使用一個機器人或以SPEL+任務執行機器人動作的大多數應用,此設定便以足夠,但您可將系統設為最多 在控制器中使用10個API任務,用以平行處理您的.NET執行緒,例如從相同控制器控制多個機器人時。

若要在控制器中使用多個API任務,必須執行兩個基本步驟。

- 在Epson RC+ GUI中,連接至控制器,然後開啟[Setup]-[System Configuration]-[Controller]-[Preferences]。 將「Reserved tasks for API」設為所需的API任務數量。請注意,您保留的API任務越多,SPEL+程式可使用的 任務就越少。例如:如果您保留5個API任務,則SPEL+可使用27個任務(32-5)。
- 2. 在應用程式中,設定CommandTask屬性,以指定要執行方法的API任務。

在下方的簡易範例中,相同控制器的每個機器人都有一個執行緒。由於在每個執行緒中使用不同的CommandTask, 且兩個Spel執行個體的ServerInstance皆設為1,因此機器人動作命令將會並行執行。

VB 例:

```
' 初始化執行緒1的Spel類別執行個體
m_spel_1 = New Spel
m_spel_1.ServerInstance = 1
m_spel_1.CommandTask = 1
m_spel_1.Initialize()
m_spel_1.Project = "c:\EpsonRC80\Projects\MyProject\MyProject.sprj"
m_spel_1.Connect(1)
```

· 初始化執行緒2的Spe1類別執行個體

- ' 此執行個體使用與m_spel_1相同的控制器,
- ' 並在控制器中使用第二個CommandTask。

```
m_spel_2 = New Spel
m_spel_2.ServerInstance = 1
m spel 2.CommandTask = 2
執行緒1
、將m spel 1執行個體用於Robot 1動作
m spel 1.Robot = 1
Do
m spel 1.Go(1)
m spel_1.Go(2)
Loop Until m_stop
執行緒2
、將m spel 2執行個體用於Robot 2動作
m spel 2.Robot = 2
Do
m spel 2.Go(1)
m spel 2.Go(2)
Loop Until m stop
```

C# 例:

```
// 初始化執行緒1的Spel類別執行個體
RCAPINet.Spel m_spel_1 = new RCAPINet.Spel();
m_spel_1.ServerInstance = 1;
m_spel_1.CommandTask = 1;
m_spel_1.Initialize();
m_spel_1.Project = @"c:\EpsonRC80\Projects\MyProject\MyProject.sprj";
m_spel_1.Connect(1);
```

// 初始化執行緒2的spe1類別執行個體

// 此執行個體使用與m spel 1相同的控制器,

// 並在控制器中使用第二個CommandTask。
RCAPINet.Spel m_spel_2 = new RCAPINet.Spel();
m_spel_2.ServerInstance = 1;
m_spel_2.CommandTask = 2;

執行緒1

```
// 將m_spel_1執行個體用於Robot 1動作
m_spel_1.Robot = 1;
do{
m_spel_1.Go(1);
m_spel_1.Go(2);
}while(\!m_stop);
```

執行緒2

// 將m_spel_2執行個體用於Robot 2動作
m_spel_2.Robot = 2;
do{
m_spel_2.Go(1);
m_spel_2.Go(2);
}while(\!m_stop);

6.2 執行SPEL+程式

SPEL+程式包含一或多個函數,並透過啟動其main函數來執行程式。您可透過使用Spel類別的Start方法,在目前控制器專案中執行64種內建main函數的任何一種。您啟動的main函數必須在您的SPEL+程式碼中定義。當您啟動main

函數時,所有全域變數和模組變數會恢復為預設值。 下表顯示SPEL+專案中的程式編號及其對應函數名稱。

程式編號	SPEL+函數名稱
0	main
1	main1
2	main2
3	main3
63	main63

以下是啟動「main」函數的範例: VB例:

C# 例:

```
void btnStart_Click(object sender, EventArgs e)
{
    m_spel.Start(0); //啟動 main 函數
    btnStart.Enabled = false;
    btnStop.Enabled = true;
}
```

6.3 執行SPEL+程式

您可使用Xqt方法在SPEL+程式中執行函數,如同一般任務一樣。當您執行任務時,全域變數不會恢復為您使用Start 方法時的預設值。

若要暫停及繼續任務,請使用Halt及Resume方法。 若要結束任務,請使用Quit方法。 您也可以使用StartBGTask方法啟動控制器背景任務。

6.4 終止所有任務

正在執行任務時,若您想一次終止所有任務,您可使用Spel類別的Stop方法。Stop方法具有選用參數,可讓您另外停止所有背景任務。

VB 例:

C# 例:

```
void btnStop_Click(object sender, EventArgs e)
{
    m_spel.Stop();
    btnStop.Enabled = false;
    btnStart.Enabled = true;
}
```

7.事件

7.1 概述

Spel類別支援兩種事件:系統事件和使用者事件。系統事件是系統狀態的通知。使用者定義事件會從任何SPEL+任務傳送至.NET應用程式。

7.2 系統事件

有幾種系統事件會傳送至.NET應用程式。事件包括Pause、Continue、EStopOn等。每個系統事件都代表狀態的改變。

有關系統事件的詳細資訊,請參閱以下內容。

Spel類別事件

請使用Spel類別EnableEvents方法控制要傳送的系統事件。

7.3 來自SPEL+的使用者事件

您可從SPEL+程式讓事件在.NET應用程式中發生。例如,您可以將連續循環通知給.NET應用程序。這種方法比在 控制器中從.NET使用變數值輪詢要來得好。若要從SPEL+將事件引發至Visual Basic,請在SPEL+程式陳述式中使 用「SPELCom_Event」命令。

範例: (Visual Basic)

SPELCom_Event 1000, cycNum, lot\$, cycTime

「SPELCom_Event」命令類似於「Print」命令。您可指定一或多個要傳送至.NET應用程式的資料。關於「SPELCom_Event」的詳細資訊,請參閱以下內容。

SPELCom_Event

在接收事件之前,您必須使用WithEvents子句宣告Spel類別變數。

Public WithEvents m spel As RCAPINet.Spel

針對Spel類別執行個體攔截EventReceived常式中的事件。若要編輯此常式,請在宣告Spel類別的模組中,從類別名稱清單選擇「m_spel」,並從程序清單選擇「EventReceived」。

以下是發生事件時更新部分標籤的EventReceived常式中之程式碼範例。

VB 例:

```
Sub m_spel_EventReceived (ByVal sender As Object, _
ByVal e As RCAPINet.SpelEventArgs) _
Handles m_spel.EventReceived
Dim tokens() As String
Select Case e.Event
Case 2000
tokens = e.Message.Split(New [Char]() {" "c}, _
System.StringSplitOptions.RemoveEmptyEntries)
lblCycCount.Text = tokens(0)
lblLotNumber.Text = tokens(1)
lblCycTime.Text = tokens(2)
End Select
End Sub
```

C# 例:

```
void m_spel_EventReceived(object sender, SpelEventArgs e)
```

```
string[] tokens = new string[3];
switch(e.Event)
{
    case 2000:
        tokens = e.Message.Split(' ');
        lblCycCount.Text = tokens(0);
        lblLotNumber.Text = tokens(1);
        lblCycTime.Text = tokens(2);
        break;
    default:
        break;
    }
}
```

8. 錯誤處理

8.1 Spel方法的錯誤

當您執行Spel類別方法時,會在發生錯誤時擲回例外。 發生錯誤時,Spel類別執行個體會將其擲回調用常式。您應使用應用程式中的錯誤處理常式來攔截此錯誤。在某些情 況下,您會只想顯示錯誤訊息。

VB 例:

```
Sub btnStart_Click(
	ByVal sender As System.Object, _
	ByVal e As System.EventArgs) _
	Handles btnStart.Click
Try
	m_spel.Start(0)
Catch ex As RCAPINet.SpelException
	MsgBox(ex.Message)
End Try
End Sub
```

您可使用SpelException的ErrorNumber屬性,檢視有關例外的錯誤編號。

```
Try
  m_spel.Start(0)
Catch ex As RCAPINet.SpelException
  MsgBox("SPEL Error: " + ex.ErrorNumber.ToString())
End Try
```

C# 例:

```
void btnStart_Click(object sender, EventArgs e)
{
  try{
    m_spel.Start(0);
    }
    catch(SpelException ex){
    MessageBox.Show(ex.Message);
    }
}
```

您可使用SpelException的ErrorNumber屬性,檢視有關例外的錯誤編號。

```
try {
  m_spel.Start(0);
}
catch(SpelException ex) {
  MessageBox.Show(("SPEL Error: " + ex.ErrorNumber.ToString());
}
```

9. 處理暫停和繼續

9.1 暫停狀態

發生暫停時,控制器與SPEL+任務會處於暫停狀態。 在任務執行期間發生下列狀況時,控制器會處於暫停狀態: 執行了Spel類別Pause方法 SPEL+任務執行了Pause。 安全防護打開。

9.2 攔截Pause事件

Spel類別會向您的.NET應用程式通知有暫停狀況發生。 您可在Spel類別的EventReceived事件中攔截Pause事件。

VB 例:

```
Sub m_spel_EventReceived (ByVal sender As Object, ByVal e As RCAPINet.
SpelEventArgs) Handles m_spel.EventReceived
Select Case e.Event
Case RCAPINet.SpelEvents.Pause
btnPause.Enabled = False
btnContinue.Enabled = True
End Select
End Sub
```

C# 例:

```
void m_spel_EventReceived(object sender, SpelEventArgs e)
{
    switch(e.Event)
        {
            case SpelEvents.Pause:
                btnPause.Enabled = false;
                btnContinue.Enabled = true;
                break;
                default:
                     break;
                }
}
```

9.3 執行暫停

以下常式說明如何使用Pause方法從Visual Basic發出PAUSE。

VB 例:

```
Sub btnPause_Click(
	ByVal sender As System.Object,
	ByVal e As System.EventArgs) _
	Handles btnPause.Click
m_spel.Pause()
	btnPause.Enabled = False
	btnContinue.Enabled = True
End Sub
```

C# 例:

```
void btnPause_Click(object sender, EventArgs e)
{
    m_spel.Pause();
    btnPause.Enabled = false;
    btnContinue.Enabled = true;
}
```

9.4 暫停後繼續

若要在發生暫停後繼續,請使用Continue方法。

VB 例:

```
Sub btnContinue_Click(
	ByVal sender As System.Object, _
	ByVal e As System.EventArgs) _
	Handles btnContinue.Click
m_spel.Continue()
	btnContinue.Enabled = False
	btnPause.Enabled = True
End Sub
```

C# 例:

```
void btnContinue_Click(object sender, EventArgs e)
{
    m_spel.Continue();
    btnContinue.Enabled = false;
    btnPause.Enabled = true;
}
```

9.5 暫停後終止

若不想在暫停後繼續,您也可以執行Stop方法。

VB 例:

```
Sub btnStop_Click(
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles btnStop.Click
m_spel.Stop()
    btnContinue.Enabled = False
    btnPause.Enabled = False
End Sub
```

C# 例:

```
void btnStop_Click(object sender, EventArgs e)
{
    m_spel.Stop();
    btnContinue.Enabled = true;
    btnPause.Enabled = false;
}
```

10. 處理緊急停止

發生緊急停止時,您可能會想在程式中執行某些特定操作,例如顯示對話方塊或訊息方塊。 Spel類別會針對緊急停止狀態發出兩種標準事件:EStopOn和EStopOff。

10.1 使用系統EStop事件

您可在Visual Basic應用程式的EventReceived處理常式中攔截系統EStop事件。

```
Imports RCAPINet.Spel
Private Sub m_spel_EventReceived(ByVal sender As Object, ByVal e As SpelEventArgs)
Handles m_spel.EventReceived
Select Case e.Event
Case RCAPINet.SpelEvens.EstopOn
MsgBox "E-Stop detected"
gEStop = True
lbleStop.BackColor = Color.Red
lbleStop.Text = "EStop ON"
Case RCAPINet.SpelEvents.EstopOff
gEStop = False
lbleStop.BackColor = Color.Green
lblEStop.Text = "EStop OFF"
End Select
Find Sub
```

End Sub

在C#應用程序中,您可以將系統EStop事件捕獲到「EventReceived」中。

```
private void m_spel_EventReceived(object sender, SpelEventArgs e)
{
    switch(e.Event)
    {
        case SpelEvents.EstopOn:
            MessageBox.Show("E-Stop detected");
        gEStop = true;
        lblEStop.BackColor = Color.Red;
        lblEStop.Text = "EStop ON";
        case SpelEvents.EstopOff:
        gEStop = false;
        lblEStop.BackColor = Color.Green;
        lblEStop.Text = "EStop OFF";
        }
}
```

11. EpsonRC+8.0視窗及對話方塊

您可以使用Spel類別的ShowWindow和RunDialog方法,從.NET應用程式開啟特定的Epson RC+ 8.0視窗及對話方 塊。

11.1 視窗

視窗屬於非強制回應,也就是可在使用Visual Basic GUI的其他元素時保持開啟。您可從Visual Basic程式顯示及隱藏 Epson RC+ 8.0視窗。

例:若要開啟及關閉I/O監視器視窗

```
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, Me)
m_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor)
```

使用C #GUI的其他元素時,也可以顯示Epson RC+視窗。

```
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, this);
m spel.HideWindow(RCAPINet.SpelWindows.IOMonitor);
```

視窗ID	視窗
RCAPINet.SpelWindows.IOMonitor	IO監視器
RCAPINet.SpelWindows.TaskManager	Task Manager
RCAPINet.SpelWindows.ForceMonitor	Force Monitor
RCAPINet.SpelWindows.Simulator	Simulator

🜼 I/O Monitor - Virtual I/O Mode 🗸 🗖 🗙					
_					+
All Inp	outs		All Ou	tputs ×	
All In	puts	\checkmark	All O	utputs	\checkmark
Bit	Byte	Word Q	Bit	Bvte	Word Q
				2,12	
Bit	Status	Label	Bit	Status	Label
0	0	Start	0	0	Ready
1	0	SelProg1	1	0	Running
2	0	SelProg2	2	0	Paused
3	0	SelProg4	3	0	Error
4	0	Stop	4	0	
5	0	Pause	5	0	SafeguardOn
6	0	Continue	6	0	SError
7	0	Reset	7	0	Warning
8	0		8	۲	EStopOff
9	0		9	0	
	Edit Show only registered			dit	Show only registered
ŀ	lexadecir	nal Values	H	lexadecin	nal Values
		Read Write			Read Write

I/O Monitor視窗

11.2 對話方塊

顯示[機器人管理器]的對話框時,也可使用 .NET GUI 的其他元素。

例:[機器人管理器]對話框的開啟方法

m spel.RunDialog(RCAPINet.SpelDialogs.RobotManager)

顯示[機器人管理器]以外的對話框時,必須先關閉對話框才能使用 .NET GUI 的其他元素。(模態對話框)

對話方塊一旦開啟, 就必須由操作員關閉。您無法從程式中關閉對話方塊。此為安全上的考量。 下表顯示可以開啟的對話方塊。

對話方塊ID	對話方塊
RCAPINet.SpelDialogs.RobotManager	Robot Manager
RCAPINet.SpelDialogs.ControllerTools	Controller Tools
RCAPINet.SpelDialogs.VisionGuide	Vision Guide
RCAPINet.SpelDialogs.ForceGuide	Force Guide

對話方塊ID	對話方塊
RCAPINet.SpelDialogs.PartFeeding	Part Feeding

12. 顯示視訊

您可藉由使用SPELVideo控制項,在應用程式的表單上顯示視訊。執行視覺序列時,圖形也可顯示在視窗上。

12.1 顯示視訊

請執行下列步驟來建立視訊顯示:

1. 將SPELVideo控制項添加至專案。

在.NET Framework與.NET6中,對於SPELVideo控制項的添加方法,有不同的設定方法。

若為.NET Framework

開啟[Tools]-[Choose Toolbox Items]。選擇瀏覽標籤,瀏覽至¥EpsonRC80¥Exe目錄,並選擇RCAPINt2.dll檔案。SPELVideo控制項圖示將會添加至工具箱。

若為.NET6

(1) 將SPELVideo添加至專案。開啟[Tools]-[NuGet Package Manager]-[Package Manager Settings]。



(2) 從樹狀目錄中,選擇套件來源。按一下右上方的[+]按鈕,建立套件來源。 文字方塊的[Name]:隨意,[Source]:輸入已安裝Epson RC+的路徑(若無指定路徑而安裝者,則輸入 C:¥EpsonRC80¥API),然後按下[OK]按鈕。

Options				?	×
Search Options (Ctrl+E)	P Pack	age sources:		+ ×	
 F# Tools GitHub Graphics Diagnostics IntelliCode Live Share NuGet Package Manager General Configuration Files Package Sources Package Source Mapping Test Test Adapter for Google Test 		nuget.org https://api.nuget.org/v3/index.json Package source https://packagesource Microsoft Visual Studio Offline Packages C:¥Program Files (x86)¥Microsoft SDKs¥NuGetP RCAPI C:¥EpsonRC80¥API	ackages¥		
 Test Adapter for Google Test Text Templating vcpkg Package Manager Viasfora Web Forms Designer Web Performance Test Tools 	<u>N</u> ame Source	e: <u>RCAPI</u> ce: <u>C:¥EpsonRC80¥API</u>		Update	
			ОК	Cancel	

(3) 開啟[Tools]-[NuGet Package Manager]-[Manage NuGet Packages for Solution]。

e	Tool	ls	Extensions	Window	Help	₽ Searc	h -		WinFormsApp1
inF		Get	Tools and Fea	atures				-[]-0	• <u>II</u> * <u>T</u> [2] @ \$\$ \$\$ \$ \$
		Mar	nage Preview	Features			- 1		
		And	droid				•		
		iOS							
		Arc	hive Manager				_		
	٣ð	Con	nect to Datab	ase					
	Ë	Con	nect to Serve	r					
		Cod	le Snippets M	anager	Ctr	I+K, CtrI+B			
		Cho	ose Toolbox I	tems					
		Nu	Get Package N	/lanager		I		-	Package Manager Console
		Crea	ate GUID				Ċ	0	Manage NuGet Packages for Solution
		Erro	or Lookup					3	Package Manager Settings
		Spy	++						
		Exte	ernal Tools					ľ	
		The	me				·		
		Con	nmand Line				·		
		Imp	ort and Expor	t Settings					
		Cus	tomize						
	£33	Opt	tions						

(4) 按一下[瀏覽]標籤,	搜尋並選擇[RCAPINt2]。	選擇想導入的專案,	並按一下[Install]。	安裝完成後,
SPELVideo控制項圖示	将會添加至工具箱。			

RCAPINt2 by Seiko Epson Corporation NuGet Package for RC+ API.	1.0.0	Versions - 1		
		 Project WinFormsApp1 	Version 1.0.0	Installed 1.0.0
	Ŧ	Installed: 1.0.0 Version: Latest stable 1.0.0	[Uninstall

- 2. 將SPELVideo控制項放置在您要顯示視訊的表單上。控制項尺寸最大可改變至全尺寸。
- 3. 將VideoEnabled屬性設為True。
- 4. 若要顯示視覺圖形,請將GraphicsEnabled屬性設為True。此外,您必須使用Spel類別SpelVideoControl屬性,將 SPELVideo控制項附加至Spel類別執行個體。

SPELVideo	E Form1		
	Video Enabled	Graphics Enabled	

放置在表單上的SPELVideo控制項

當GraphicsEnabled屬性為True且控制項附加至Spel類別執行個體時,每當在連接至Spel類別執行個體的控制器上執行VRun方法,都會顯示視覺圖形。

以下範例顯示如何從使用Spel類別執行個體並放置SPELVideo控制項的Visual Basic表單上啟用視訊與圖形:

```
Private Sub Form_Load(sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
    m_spel = New Spel
    m_spel.Initialize()
    m_spel.Project = "c:\EpsonRC80\projects\test.sprj"
    SpelVideo1.VideoEnabled = True
    SpelVideo1.GraphicsEnabled = True
    m_spel.SpelVideoControl = SPELVideo1
End Sub
```

如何在SPELVideo控制項所在的C#表單上, 啓用視訊與圖形

12.2 使用多視訊顯示

Epson RC+ 8.0可在應用程式中使用多視訊顯示。對於各顯示,您可選擇要顯示的攝影機視訊。 若要使用多個顯示,您必須為每個顯示設定SpelVideoControl屬性。 以下範例顯示包含兩個視訊顯示的初始化。

VB 例:

```
Private Sub Form_Load(sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
    m_spel = New Spel
    m_spel.Initialize()
    m_spel.Project = "c:\EpsonRC80\projects\test\test.sprj"
    SpelVideo1.VideoEnabled = True
    SpelVideo1.GraphicsEnabled = True
    SpelVideo2.VideoEnabled = True
    SpelVideo2.VideoEnabled = True
    SpelVideo2.GraphicsEnabled = True
    SpelVideo2.Camera = 2
    m_spel.SpelVideoControl = SPELVideo1
    m_spel.SpelVideoControl = SPELVideo2
End Sub
```

```
C# 例:
```

```
private void Form_Load)object sender, EventArgs e)
{
    RCAPINet.Spel m_spel = new RCAPINet.Spel();
    m_spel.Initialize();
    m_spel.Project = @"c:\\EpsonRC80\\project\\test\\test.sprj";
    SpelVideo1.VideoEnabled = true;
    SpelVideo1.GraphicsEnabled = true;
    SpelVideo1.Camera = 1;
    SpelVideo2.VideoEnabled = true;
    SpelVideo2.GraphicsEnabled = true;
    SpelVideo3.GraphicsEnabled = tr
```

13. 使用AsyncMode

AsyncMode可讓您在執行其他方法期間同時執行Spel方法。只有下列的Spel類別方法可以非同步執行:

13.1 使用AsyncMode

Arc	Jump3
Arc3	Jump3CP
BGo	MCal
BMove	Move
CVMove	Pass
Go	PTran
Home	Pulse
JTran	TGo
Jump	TMove

若要非同步執行某方法,請將AsyncMode屬性設為True,然後執行該方法。當AsyncMode屬性為true並執行一非同步方法時,該方法會隨即啟動,且控制項會立即返回.NET應用程式,以作進一步處理。若您在上一個方法正在執行時執行其他非同步方法,SPEL將會等待第一個方法完成,再啟動下一個方法,然後返回.NET。

若要等待非同步方法完成,您可執行下列其中一項操作:

- 執行WaitCommandComplete方法。
- 將AsyncMode屬性設為False。

如果因為錯誤(例如指定點不存在)而無法啟動非同步命令,將會立即出現例外。不過,如果在非同步執行命令期間發 生錯誤,則會在執行下一個非同步命令、執行WaitCommandComplete或AsyncMode設為False時出現錯誤例外。如 果在執行下一個命令時出現例外,您會不知道是哪個陳述式導致錯誤(上一個或目前陳述式)。如果您需要在執行其他 命令前檢查非同步命令是否成功完成,則在執行下一個命令前調用WaitCommandComplete。如果在上一個非同步命 令期間發生錯誤,SpelException例外將會與錯誤編號及訊息一起顯示。請參閱以下範例。

```
VB 例:
```

```
Try
    m_spel.AsyncMode = True
    m_spel.Go(1)
    f 在動作期間於此處執行其他操作
    u果Go(2)執行時出現錯誤, Go(1)在執行期間出現異常,
    l則我們無法得知錯誤發生於Go(1)或Go(2)
    m_spel.Go(2)
    m_spel.Go(3)
    f 在動作期間於此處執行其他操作
    k檢査Go(3)是否成功
    m_spel.WaitCommandComplete() ' 如果Go(3)發生錯誤, 則會出現例外
    m_spel.Go(4)
Catch ex As SpelException
    k 處理錯誤例外
```

End Try

C# 例:

```
try {
```

}

}

```
m spel.AyncMode = true;
 m spel.Go(1);
 // 在動作期間於此處執行其他操作
 // 如果Go(2)執行時出現錯誤, Go(1)在執行期間出現異常,
 // 則我們無法得知錯誤發生於Go(1)或Go(2)
 m_spel.Go(2);
 m_spel.Go(3);
 // 在動作期間於此處執行其他操作
 // 檢查Go(3)是否成功
 m spel.WaitCommandComplete();
 // 如果Go(3)發生錯誤, 則會出現例外
 m_spel.Go(4);
catch(RCAPINet.SpelException ex) {
 // 處理錯誤例外
```

14. SPELCom_Event

從Spel類別執行個體產生使用者事件。

14.1 SPELCom_Event

語法

SPELCom_Event eventNumber[, msgArg1, msgArg2, msgArg3,...]

參數

- 事件編號 數值介於1000 - 32767的整數運算式。
- 訊息參數 數字、字串常值或變數名稱。

描述

此指令方便您將即時資訊傳送至控制器中正在執行的Spel任務之某應用程式。例如:您可透過傳送事件來更新工件計數、批號等。

SPELCom_Event範例

在此範例中, SPEL+任務利用RC+ API將週期資料傳送至應用程式。

```
Function RunParts
Integer cycNum
String lot$
Double cycTime
cycNum = 0
Do
TmrReset(0)
...
cycTime = Tmr(0)
cycNum = cycNum + 1
Spelcom_Event 3000, cycNum, lot$, cycTime
Wait 0.01
Loop
Fend
```

15. RCAPINet參考

15.1 Spel類別

描述

此類別可讓您執行命令並接收Epson RC+ 8.0的事件。

檔案名稱

RCAPINt2.dll(64位元與32位元)

15.2 Spel類別屬性

15.2.1 AsyncMode屬性, Spel類別

描述

設定/傳回非同步執行模式。

語法

Property AsyncMode As Boolean

預設值

False

傳回值

非同步模式作用時,布林值為True,未作用時則為False。

另請參閱

使用AsyncMode, WaitCommandComplete

AsyncMode範例

VB 例:

```
With m_spel
   .AsyncMode = True
   .Jump("pick")
   .Delay(500)
   .On(1)
   .WaitCommandComplete()
End With
```

C# 例:

```
m_spel.AsyncMode = true;
m_spel.Jump("pick");
m_spel.Delay(500);
m_spel.On(1);
m_spel.WaitCommandComplete();
```

15.2.2 AvoidSingularity屬性, Spel類別

描述

設定/傳回奇點避開模式。

語法

Property AvoidSingularity As Boolean

預設值

False

傳回值

奇點避開作用時, 布林值為True, 未作用時則為False。

另請參閱

Go, Jump, Move

AvoidSingularity範例 VB 例:

m_spel.AvoidSingularity = True

C# 例:

m_spel.AvoidSingularity = true;

15.2.3 CommandInCycle屬性, Spel類別

描述

傳回顯示某方法是否正在執行。

語法

ReadOnly Property CommandInCycle As Boolean

傳回值

方法正在執行時, 布林值為True, 未執行時則為False。

另請參閱

AsyncMode

CommandInCycle範例

VB 例:

```
If m_spel.CommandInCycle Then
    MsgBox "A SPEL command is executing, operation aborted"
End If
```

C# 例:

```
if (m_spel.CommandInCycle)
    MessageBox.Show("SPEL command is executing, operation aborted");
```

15.2.4 CommandTask屬性, Spel類別

描述

為執行機器人命令指定在控制器使用的保留API任務。

語法

Property CommandTask As Integer

預設值

預設值為0(不使用保留的API任務)。

備註

當您想在控制器中以其他執行緒執行Spel機器人命令時,請使用CommandTask。一般而言,CommandTask用於多 機器人系統。使用CommandTask前,您必須先從Epson RC+功能表-[Setup]-[System Configuration]-[Controller]-[Preferences]保留要在控制器中使用的API任務。您最多可在控制器中保留16個API任務。

另請參閱

ServerInstance

CommandTask範例

VB 例:

```
' 在Robot1執行緒中
m_spel.CommandTask = 1
m_spel.Robot = 1
' 在Robot2執行緒中
m_spel.CommandTask = 2
m_spel.Robot = 2
C# 例:
// 在Robot1執行緒中
m_spel.CommandTask = 1;
m_spel.Robot = 1;
// 在Robot2執行緒中
m_spel.CommandTask = 2;
m_spel.Robot = 2;
```

15.2.5 DisableMsgDispatch屬性, Spel類別

描述

設定/傳回顯示Windows訊息是否應在Spel方法執行期間處理。

語法

DisableMsgDispatch

類型

Boolean

預設值

False

備註

此屬性通常不應使用。主要是用於當執行Spel方法時不想用鍵盤或滑鼠處理的特殊應用。
15.2.6 ErrorCode屬性, Spel類別

描述

傳回目前控制器錯誤代碼。

語法

ReadOnly Property ErrorCode As Integer

傳回值

包含錯誤代碼的整數值。

另請參閱

ErrorOn

ErrorCode範例

VB 例:

```
If m_spel.ErrorOn Then
    lblErrorCode.Text = m_spel.ErrorCode.ToString()
Else
    lblErrorCode.Text = ""
End If
C# 例:
if (m_spel.ErrorOn)
    lblErrorCode.Text = m_spel.ErrorCode.ToString();
else
    lblErrorCode.Text = "";
```

15.2.7 ErrorOn屬性, Spel類別

描述

若控制器發生嚴重錯誤,會傳回True。

語法

ReadOnly Property ErrorOn As Boolean

傳回值

若控制器處於錯誤狀態,會傳回True,否則會傳回False。

備註

當控制器處於錯誤狀態時,ErrorOn屬性會傳回True,且您可使用ErrorCode擷取錯誤代碼。

另請參閱

ErrorCode

ErrorOn範例

VB 例:

```
If m_spel.ErrorOn Then
    m_spel.Reset
End If
C# 例:
if (m_spel.ErrorOn)
    m_spel.Reset();
```

15.2.8 EStopOn屬性, Spel類別

描述

傳回控制器緊急停止的狀態。

語法

ReadOnly Property EStopOn As Boolean

傳回值

若緊急停止生效,會傳回True,否則會傳回False。

EStopOn範例

VB 例:

```
If m_spel.EStopOn Then
    lblEStop.Text = "Emergency stop is active"
Else
    lblEStop.Text = ""
EndIf
```

```
if (m_spel.EStopOn)
    lblEStop.Text = "Emergency stop is active";
else
    lblEStop.Text = "";
```

15.2.9 MotorsOn屬性, Spel類別

描述

設定並傳回目前機器人馬達開啟或關閉的狀態。

語法

Property MotorsOn As Boolean

預設值

False

傳回值

馬達開啟時, 布林值為True, 未開啟時則為False。

另請參閱

PowerHigh, Reset, Robot

MotorsOn範例

VB 例:

```
If Not m_spel.MotorsOn Then
    m_spel.MotorsOn = True
End If
```

```
if (\!m_spel.MotorsOn)
    m_spel.MotorsOn = true;
```

15.2.10 NoProjectSync屬性, Spel類別

描述

設定/傳回是否要將PC中的目前專案與控制器專案進行同步。

語法

NoProjectSync

類型

Boolean

預設值

False

備註

當NoProjectSync設為False(預設)時,Spel類別會確保PC中的專案與控制器中的專案進行同步。 當NoProjectSync設為True時,Spel類別不會檢查PC上的任何專案,也不會將PC專案與控制器進行同步。這可讓您 執行控制器中的程式,而不會執行PC中的任何專案。

此屬性不是持續有效。如果您想將其設為True,必須在建立Spel類別執行個體後進行設定。

另請參閱

Start

NoProjectSync範例

VB 例:

```
m_spel.Initialize()
m_spel.NoProjectSync = True
```

```
m_spel.Initialize();
m spel.NoProjectSync = true;
```

15.2.11 OperationMode屬性, Spel類別

描述

讀取或設定Epson RC+ 8.0操作模式。

語法

Property OperationMode As SpelOperationMode

傳回值

SpelOperationMode值

備註

當OperationMode設為Program時, Spel類別目前執行個體的Epson RC+ 8.0 GUI會開啟, 而控制器操作模式會設為 Program。若使用者關閉GUI, OperationMode會設為Auto。若從Visual Basic將OperationMode設為Auto, 則GUI 會同時關閉。

OperationMode範例

VB 例:

```
Sub btnSpelProgramMode Click
       ByVal sender As System.Object,
       ByVal e As System.EventArgs)
       Handles btnHideIOMonitor.Click
  Try
    m spel.OperationMode =
                  RCAPINet.SpelOperationMode.Program
    · 如果您想等待使用者關閉RC+ GUI,
    ' 您可在此等待OperationMode變更為Auto
    Do
      Application.DoEvents()
      System.Threading.Thread.Sleep(10)
    Loop Until m spel.OperationMode =
                   RCAPINet.SpelOperationMode.Auto
  Catch ex As RCAPINet.SpelException
    MsgBox(ex.Message)
  End Try
End If
```

```
void btnSpelProgramMode_Click(object sender, EventArgs e)
{
    try {
    m_spel.OperationMode = RCAPINet.SpelOperationMode.Auto;
    //如果想等待RC+ GUI關閉,可在此待機
    Do {
        Application.DoEvents();
        Systme.Threading.Thread.Sleep(10);
        } while(\!m_spel.OperationMode = RCAPINet.OperationMode.Auto);
    }
    Catch (SpelException ex) {
        MessageBox.Show(ex.Message);
    }
}
```

15.2.12 ParentWindowHandle屬性, Spel類別

描述

設定/傳回對話方塊及視窗所用父視窗的控制代碼。

語法

Property ParentWindowHandle As Integer

傳回值

包含視窗控制代碼的整數值。

備註

當ServerOutOfProcess設定為True時使用此屬性。從不包含.NET表單的應用程式(例如LabVIEW),使用 ParentWindowHandle來指定父視窗。

另請參閱

ServerOutOfProcess

ParentWindowHandle範例

VB 例:

m_spel.ParentWindowHandle = Me.Handle
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor)

C# 例:

m_spel.ParentWindowHandle = (int)this.Handle; m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor);

15.2.13 PauseOn屬性, Spel類別

描述

傳回控制器暫停狀態的狀態。

語法

ReadOnly Property PauseOn As Boolean

傳回值

若控制器處於暫停狀態,會傳回True,否則會傳回False。

另請參閱

Continue, Pause

PauseOn範例

VB 例:

```
If m_spel.PauseOn Then
    btnPause.Enabled = False
    btnContinue.Enabled = True
End If
```

```
if (m_spel.PauseOn) {
    btnPause.Enabled = false;
    btnContinue.Enabled = true;
}
```

15.2.14 PowerHigh屬性, Spel類別

描述

設定並傳回目前機器人的運行功率狀態。

語法

Property PowerHigh As Boolean

預設值

False

傳回值

若目前機器人運行功率為high, 會傳回True, 否則會傳回False。

另請參閱

MotorsOn

PowerHigh範例

VB 例:

```
If Not m_spel.PowerHigh Then
    m_spel.PowerHigh = True
End If
```

```
if (\!m_spel.PowerHigh)
    m_spel.PowerHigh = true;
```

15.2.15 Project屬性, Spel類別

描述

設定/傳回目前專案。

語法

Project, Property As String

預設值

空字串。

傳回值

包含專案路徑及檔案的字串。

備註

設定Project時,您必須提供Epson RC+ 8.0專案製作檔案的完整路徑與名稱。該製作檔案為包含.SPRJ副檔名的專案 名稱。

Project範例

VB 例:

m_spel.Project = "c:\EpsonRC80\projects\myapp\myapp.sprj"

C# 例:

m_spel.Project = @"c:\EpsonRC80\projects\myapp\myapp.sprj";

15.2.16 ProjectBuildComplete屬性, Spel類別

描述

傳回目前專案建置的狀態。

語法

ReadOnly Property ProjectBuildComplete As Boolean

傳回值

若專案建置完成,會傳回True,否則會傳回False。

另請參閱

BuildProject

ProjectBuildComplete範例

VB 例:

```
if (m_spel.ProjectBuildComplete)
    lblBuild.Text = "Project build is Complete";
else
    lblBuild.Text = "Project build is not Complete";
```

15.2.17 ProjectOverwriteWarningEnabled屬性, Spel 類別

描述

設定/傳回是否要啟用專案覆蓋時的錯誤訊息顯示。

語法

Property ProjectOverwriteWarningEnabled As Boolean

預設值

True

傳回值

若啟用覆蓋時的錯誤訊息,會傳回True,否則會傳回False。

另請參閱

BuildProject

備註

預設情況下,若當前專案與控制器的專案不同,生成程式或向控制器發送訊息時,將顯示專案將被覆蓋的錯誤訊息。 如果您不需要顯示訊息,請將ProjectOverwriteWarningEnabled 設定為 False。當您的應用時需要切換控制器使用的 專案時,可以選擇這個功能。

ProjectOverwriteWarningEnabled範例

VB 例:

```
' Disable the project overwrite warning
m_spel.ProjectOverwriteWarningEnabled = False
m_spel.Project = "c:\EpsonRC80\Projects\Project1\Project1.sprj"
```

```
// Disable the project overwrite warning
m_spel.ProjectOverwriteWarningEnabled = false;
m_spel.Project = @"c:\EpsonRC80\Projects\Project1\Project1.sprj";
```

描述

設定/傳回是否要啟用ResetAbort方法。

語法

Property ResetAbortEnabled As Boolean

預設值

True

傳回值

若ResetAbort已啟用, 會傳回True, 否則會傳回False。

另請參閱

ResetAbort

ResetAbortEnabled範例

VB 例:

' 啟用ResetAbort m_spel.ResetAbortEnabled = True

```
// 啟用ResetAbort
m_spel.ResetAbortEnabled = true;
```

15.2.19 Robot屬性, Spel類別

描述

設定/傳回目前機器人編號。

語法

Property Robot As Integer

預設值

若有一或多個機器人存在, 第一個Spel執行個體的預設值為1, 否則為0。對於其他Spel執行個體, 預設值為0。

傳回值

包含目前機器人編號的整數值。

備註

在使用多個機器人的系統上,使用Robot屬性設定後續機器人相關命令,例如動作命令。

另請參閱

RobotModel, RobotType

Robot範例

VB 例:

```
m_spel.Robot = 2
If Not m_spel.MotorsOn Then
    m_spel.MotorsOn = True
End If
```

```
m_spel.Robot = 2;
if (\!m_spel.MotorsOn)
  m spel.MotorsOn = true;
```

15.2.20 RobotModel屬性, Spel類別

描述

傳回目前機器人的型號名稱。

語法

ReadOnly Property RobotModel As String

傳回值

包含目前機器人型號名稱的字串。

<mark>另請參閱</mark> Robot, RobotType

RobotModel範例

VB 例:

lblRobotModel.Text = m_spel.RobotModel

C# 例:

lblRobotModel.Text = m_spel.RobotModel;

15.2.21 RobotType屬性, Spel類別

描述

傳回目前機器人的類型。

語法

ReadOnly Property RobotType As SpelRobotType

傳回值

SpelRobotType值

另請參閱

Robot, RobotModel

RobotType範例

VB 例:

```
Select Case m_spel.RobotType
Case RCAPINet.SpelRobotType.Scara
    lblRobotType.Text = "Scara"
Case RCAPINet.SpelRobotType.Cartesian
    lblRobotType.Text = "Cartesian"
End Select
```

```
C# 例:
```

```
switch (m_spel.RobotType)
{
    case SpelRobotType.Scara:
        lblRobotType.Text = "Scara";
    break;
    case SpelRobotType.Cartesian:
        lblRobotType.Text = "Cartesian";
    break;
    default:
        break;
}
```

15.2.22 SafetyOn屬性, Spel類別

描述

傳回控制器安全防護輸入的狀態。

語法

ReadOnly Property SafetyOn As Boolean

傳回值

若安全防護打開,會傳回True,否則會傳回False。

備註

請在應用程式啟動時,使用SafetyOn屬性取得安全防護狀態,然後使用SafeguardOpen和SafeguardClose事件來更新 狀態。

SafetyOn範例

VB 例:

```
If m_spel.SafetyOn Then
    lblSafeguard.Text = "Safe guard is active"
Else
    lblSafeguard.Text = ""
End If
```

```
if (m_spel.SafetyOn)
    lblSafeguard.Text = "Safe guard is active";
else
    lblSafeguard.Text = "";
```

15.2.23 ServerInstance屬性, Spel類別

描述

指定Epson RC+伺服器要使用的執行個體。

語法

Property ServerInstance As Integer

預設值

預設值為下一個可用的伺服器執行個體。

備註

API與RC+伺服器進程通信。ServerInstance指定要使用的伺服器。每個伺服器實例對應一個控制器和一個專案。預設情況下,創建新Spel類實例時,ServerInstance將自動設置為"1"。

當您想要對同一控制器 (如應用程式的多線程)使用多個Spel類實例,請為使用同一控制器的每個Spel類實例設置 ServerInstance屬性。

ServerInstance 必須介於1和10之間,並且必須在初始化或執行其他方法之前進行設置。

另請參閱

CommandTask, Initialize

ServerInstance範例

VB 例:

```
' Controller 1
spel1 = New Spel
spel1.ServerInstance = 1
spel1.Initialize()
spel1.Connect(1)
' Controller 2
spel2 = New Spel
spel2.ServerInstance = 2
spel2.Initialize()
spel2.Connect(2)
```

```
// Controller 1
RCAPINet.Spel spel1 = new RCAPINet.Spel();
spel1.ServerInstance = 1;
spel1.Initialize();
spel1.Connect(1);
// Controller 2
RCAPINet.Spel spel2 = new RCAPINet.Spel();
spel2.ServerInstance = 2;
spel2.Initialize();
spel2.Connect(2);
```

15.2.24 SPELVideoControl屬性, Spel類別

描述

用來將SPELVideo控制項連接至Spel類別執行個體,以顯示視訊與圖形。

語法

Property SpelVideoControl As SpelVideo

另請參閱

Graphics Enabled, VideoEnabled, Camera

SpelVideoControl範例

VB 例:

m_spel.SpelVideoControl = SpelVideo1

C# 例:

m_spel.SpelVideoControl = SpelVideo1;

15.2.25 Version屬性, Spel類別

描述

傳回目前Epson RC+ 8.0軟體版本。

語法

ReadOnly Property Version As String

傳回值

包含目前Epson RC+ 8.0軟體版本的字串。

Version範例

VB 例:

取得軟體版本
 curVer = m_spel.Version

C# 例:

// 取得軟體版本 curVer = m_spel.Version;

15.2.26 WarningCode屬性, Spel類別

描述

傳回控制器警告代碼。

語法

ReadOnly Property WarningCode As Integer

傳回值

包含目前控制器警告代碼的整數值。

另請參閱

WarningOn

WarningCode範例

VB 例:

```
If m_spel.WarningOn Then
    lblWarningCode.Text = m_spel.WarningCode.ToString()
Else
    lblWarningCode.Text = ""
End If
```

```
if (m_spel.WarningOn)
    lblWarningCode.Text = m_spel.WarningCode.ToString();
else
    lblWarningCode.Text = "";
```

15.2.27 WarningOn屬性, Spel類別

描述

傳回控制器警告狀態的狀態。

語法

ReadOnly Property WarningOn As Boolean

傳回值

若控制器處於警告狀態,會傳回True,否則會傳回False。

另請參閱

WarningCode

WarningOn範例

VB 例:

```
If m_spel.WarningOn Then
    lblWarningStatus.Text = "ON"
Else
    lblWarningStatus.Text = "OFF"
End If
```

C# 例:

```
if (m_spel.WarningOn)
    lblWarningStatus.Text = "ON";
else
    lblWarningStatus.Text = "OFF";
```

15.3 Spel類別方法

15.3.1 Accel方法 · Spel類別

描述

設定PTP動作命令Go、Jump及Pulse的加速與減速。

語法

Sub Accel (PointToPointAccel As Integer, PointToPointDecel As Integer, _[JumpDepartAccel As Integer], [JumpDepartDecel As Integer], _[JumpApproAccel As Integer], [JumpApproDecel As Integer])

參數

- PointToPointAccel
 1以上整數,代表相對於最大加速度的比例(單位:%)。
- PointToPointDecel
 1以上整數,代表相對於最大減速度的比例(單位:%)。
- JumpDepartAccel
 1以上整數,代表Jump時的起始加速度(可省略)。
- JumpDepartDecel
 1以上整數,代表Jump時的起始減速度(可省略)。
- JumpApproAccel
 1以上整數,代表Jump時的接近加速度(可省略)。

JumpApproDecel
 1以上整數,代表Jump時的接近減速度(可省略)。

另請參閱

Accels, Speed

Accel範例

VB 例:

m_spel.Accel(50, 50)
m_spel.Go ("pick")

C# 例:

m_spel.Accel(50, 50); m_spel.Go ("pick");

15.3.2 AccelR方法, Spel類別

描述

設定工具旋轉動作的加速與減速。

語法

Sub AccelR (Accel As Single, [Decel As Single])

參數

Accel

0.1至5000 deg/sec2的單一運算式,用以定義在動作命令中使用ROT時的工具旋轉加速。若省略Decel,此值會同時用於加速與減速速率。

Decel

從0.1至5000的實數值(單位:deg/sec2),在動作命令中使用ROT時,用於定義工具姿勢變化的減速度。可以省略

另請參閱

Arc, Arc3, BMove, Jump3CP, Power, SpeedR, TMove

AccelR範例

VB 例:

```
Sub MoveToPlace()
    m_spel.AccelR(100)
    m_spel.Move("place ROT")
End Sub
```

```
void MoveToPlace()
{
    m_spel.AccelR(100);
    m_spel.Move("place ROT");
}
```

15.3.3 AccelS方法, Spel類別

描述

設定直綫動作和CP動作(Jump3CP, Move, TMove)的加速與減速。

語法

Sub AccelS (Accel As Single, Decel As Single, [JumpDepartAccel As Single], [JumpDepartDecel As Single], [JumpApproAccel As Single], [JumpApproDecel As Single])

參數

Accel

實數值(單位:deg/sec2),用於指定直線動作或CP動作時的加速度。若省略Decel,加速時與減速時皆會套用此Accel值。

- Decel 實數值(單位:deg/sec2),用於指定直線動作或CP動作時的減速度(可省略)。
 JumpDepartAccel
- JumpDepartAccel
 實數值(單位:deg/sec2),用於在Jump、Jump3CP時的向上動作中指定起始加速度(可省略)。
- JumpDepartDecel 實數值(單位:deg/sec2),用於在Jump、Jump3CP時的向上動作中指定起始減速度(可省略)。
- JumpApproAccel 實數值(單位:deg/sec2),用於在Jump、Jump3CP時的向下動作中指定接近加速度(可省略)。
- JumpApproDecel 實數值(單位:deg/sec2),用於在Jump、Jump3CP時的向下動作中指定接近減速度(可省略)。

另請參閱

Accel, SpeedS, Jump3CP, Move, TMove

AccelS範例

VB 例:

```
Sub MoveToPlace()
    m_spel.Base(P1)
    m_spel.Move(pick)
    m_spel.Accels(500, 300)
    m_spel.Move(place)
End Sub
```

```
void MoveToPlace()
{
    m_spel.AccelS(500);
    m_spel.Move(pick);
    m_spel.AccelS(500, 300);
    m_spel.Move(place);
}
```

15.3.4 Agl方法 · Spel類別

描述

傳回所選旋轉軸的關節角度,或所選線性軸的位置。

語法

Function Agl (JointNumber As Integer) As Single

參數

JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)

傳回值

所選旋轉軸的關節角度,或所選線性軸的位置

另請參閱

Pls, CX - CT

Agl範例

C# 例:

Dim j1Angle As Single
j1Angle = m_spel.Agl(1)

```
float jlAngle;
jlAngle = m_spel.Agl(1);
```

15.3.5 AIO_In 方法, Spel類別

描述

從選配件類比IO的輸入通道讀取類比值。

語法

Function AIO_In (Channel As Integer) As Single

參數

 Channel 指定類比IO的通道編號。

傳回值

傳回由通道指定的類比IO通道的類比輸入值(以實數爲單位)。傳回值得範圍取決於類比IO板的輸入範圍設定。

另請參閱

AIO_InW, AIO_Out, AIO_OutW

AIO_In範例

VB 例:

Dim val As Single
val = m_spel.AIO_In(2)

C# 例:

float val; val = m_spel.AIO_In(2);

15.3.6 AIO_InW 方法, Spel類別

描述

從選配件類比IO的輸入通道讀取類比值。

語法

Function AIO_InW (Channel As Integer) As Integer

參數

 Channel 指定類比IO的通道編號。

傳回值

傳回指定的類比IO通道的輸入狀態(0~65535的整數值)。

根據類比IO板的輸入範圍設定,每個輸入通道的輸入電壓(電流)和返回值之間的對應關係如下所示。

輸入數據		設定輸入範圍							
16進數	10進數	$\pm 10.24(V)$	±5.12(V)	0-5.12(V)	0-10.24(V)	0-24(mA)			
0xFFFF	65535	10.23969	5.11984	5.12000	10.24000	24.00000			
0x8001	32769	0.00031	0.00016	2.56008	5.12016	12.00037			
0x8000	32768	0.00000	0.00000	2.56000	5.12000	12.00000			
0x0000	0	-10.24000	-5.12000	0.00000	0.00000	0.00000			

另請參閱

AIO_In, AIO_Out, AIO_OutW

AIO_InW範例

VB 例:

Dim val As Integer val = m spel.AIO InW(2)

C# 例:

int val; val = m_spel.AIO_InW(2);

15.3.7 AIO_Out 方法, Spel類別

描述

從選配件類比IO的輸出通道,讀取或設定類比值。

語法

Function AIO_Out (Channel As Integer) As Single Sub AIO_Out (Channel As Integer, Value As Single)

參數

- Channel 指定類比IO的通道編號。
- Value

用陳述式或數位,表示要輸出的電壓[V]或電流值[mA]的實數。

傳回值

傳回指定類比IO通道的電壓或電流輸出狀態(以實數為單位)。電壓輸出的單位為 [V], 電流輸出的單位為 [mA]。 Function AIO_Out (Channel As Integer) As Single時:如果在指定通道輸出機器人的速度資訊時,也可以用本方法 取得傳回值。

備註

輸出表示指定電壓[V]或電流[mA]的實數,到通道號碼指定的類比輸出埠。類比輸出埠的電壓輸出範圍和電壓、電流 輸出選擇,由機板上的開關設定。如果指定的值超出類比IO板的輸出範圍,則輸出不超過範圍的邊界值(最大值或最 小值)。

Sub AIO_Out (Channel As Integer, Value As Single) 時:如果在指定通道輸出機器人的速度資訊時,本方法會發生輸出設定錯誤。請在停止輸出速度信息后,執行本方法。

另請參閱

AIO_In, AOI_InW, AIO_OutW

AIO_Out範例

VB 例:

```
Dim val As Single
val = m_spel.AIO_Out(1)
```

```
float val;
val = m spel.AIO Out(1);
```

15.3.8 AIO_OutW 方法, Spel類別

描述

從選配件類比IO的輸出通道,讀取或設定類比值。

語法

Function AIO_OutW (Channel As Integer) As Integer Sub AIO_OutW (Channel As Integer, OutputData As Integer)

參數

- Channel 指定類比IO的通道編號。
- OutputData
 用陳述式或數位,指定輸出數據(從0到65535的整數)。

傳回值

傳回指定類比IO通道的輸出狀態,用0~65535的整數表示。 根據類比IO板的輸出範圍設定,每個輸出通道的輸出電壓(電流)和返回值之間的對應關係如下所示。

輸出數據		設定輸出範圍								
16進數	10進數	±10(V)	$\pm 5(V)$	0-5(V)	0-10(V)	4-20(mA)	0-20(mA)			
0xFFFF	65535	9.99970	4.99985	5.00000	10.00000	20.00000	20.00000			
0x8001	32769	0.00031	0.00015	2.50008	5.00015	12.00024	10.00031			
0x8000	32768	0.00000	0.00000	2.50000	5.00000	12.00000	10.00000			
0x0000	0	-10.00000	-5.00000	0.00000	0.00000	4.00000	0.00000			

如果在指定通道輸出機器人的速度資訊時,也可以執行此方法。

另請參閱

AIO_In, AOI_InW, AIO_Out

AIO_OutW範例

VB 例:

Dim val As Integer
val = m_spel.AIO_OutW(1)

C# 例:

int val; val = m spel.AIO OutW(1);

15.3.9 Arc方法, Spel類別

描述

Arc利用XY平面中的圓形插補將手臂移至指定點。

語法

Sub Arc (MidPoint As Integer, EndPoint As Integer) Sub Arc (MidPoint As SpelPoint, EndPoint As SpelPoint) Sub Arc (MidPoint As String, EndPoint As String)

參數

每個語法都有兩個參數,分別指定圓弧的中點與結束點。

- MidPoint 使用整數、SpelPoint或字串運算式來指定中點。
- EndPoint
 使用代表目標位置的整數值、SpelPoint或字串時,可包含ROT、CP、SYNC、Till、Find及並行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move BGo, BMove, TGo, TMove Find, Till

Arc範例

VB 例:

```
' 使用SpelPoint指定的點
Dim midPoint, endPoint As SpelPoint
midPoint = m_spel.GetPoint("P1")
endPoint = m_spel.GetPoint("P2")
m_spel.Arc(midPoint, endPoint)
```

```
    使用字串運算式指定的點
    m_spel.Arc("P1", "P2")
    m_spel.Arc("P1", "P2 CP")
    使用平行處理
```

```
m_spel.Arc("P1", "P2 \!D50; On 1; D90; Off 1\!")
```

C# 例:

```
// 使用SpelPoint指定的點
SpelPoint midPoint, endPoint;
midPoint = m_spel.GetPoint("P1");
endPoint = m_spel.GetPoint("P2");
m_spel.Arc(midPoint, endPoint);
// 使用字串運算式指定的點
m_spel.Arc("P1", "P2");
m_spel.Arc("P1", "P2 CP");
// 使用平行處理
```

m_spel.Arc("P1", "P2 \!D50; On 1; D90; Off 1\!");

15.3.10 Arc3方法, Spel類別

描述

Arc3在3D中利用圓形插補將手臂移至指定點。

語法

Sub Arc3 (MidPoint As Integer, EndPoint As Integer) Sub Arc3 (MidPoint As SpelPoint, EndPoint As SpelPoint) Sub Arc3(MidPoint As String, EndPoint As String)

參數

每個語法都有兩個參數,分別指定圓弧的中點與結束點。

- MidPoint 使用整數、SpelPoint或字串運算式來指定中點。
- EndPoint
 使用代表目標位置的整數值、SpelPoint或字串時,可包含ROT、CP、SYNC、Till、Find及並行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS Arc, CVMove, Go, Jump, Jump3, Jump3CP, Move BGo, BMove, TGo, TMove Find, Till

Arc3範例

VB 例:

```
' 使用SpelPoint指定的點
Dim midPoint, endPoint As SpelPoint
midPoint = m_spel.GetPoint("P1")
endPoint = m_spel.GetPoint("P2")
m_spel.Arc3(midPoint, endPoint)
```

```
' 使用平行處理
m_spel.Arc3("P1", "P2 \!D50; On 1; D90; Off 1\!")
```

```
// 使用SpelPoint指定的點
SpelPoint midPoint, endPoint;
midPoint = m_spel.GetPoint("P1");
endPoint = m_spel.GetPoint("P2");
m_spel.Arc3(midPoint, endPoint);
// 使用字串運算式指定的點
m_spel.Arc3("P1", "P2");
m_spel.Arc3("P1", "P2 CP");
// 使用平行處理
m_spel.Arc3("P1", "P2 \!D50; On 1; D90; Off 1\!");
```

15.3.11 Arch方法, Spel類別

描述

定義要搭配JUMP指令使用的ARCH參數(開始水平動作之前要移動的Z高度)。

語法

Sub Arch (ArchNumber As Integer, DepartDist As Integer, ApproDist As Integer)

參數

- ArchNumber 0至6的整數,代表拱形編號。
- DepartDist 在Jump命令中代表開始水平動作之前的Z值(起始)(從起點的垂直距離)(單位:mm) 在Jump3、Jump3CP命令中代表跨距動作之前的Z值(起始)(單位:mm)
- ApproDist 在Jump命令中代表水平動作完全結束時的Z值(結束)(從目標點的垂直距離)(單位:mm) 在Jump3、Jump3CP命令中代表跨距動作完全結束時的Z值(結束)(單位:mm)

另請參閱

Jump, Jump3, Jump3CP

Arch範例

VB 例:

```
Sub SetArchs()
    With m_spel
        .Arch(1, 30, 30)
        .Arch(2, 60, 60)
        .Jump("P1 C1")
        .Jump("P2 C2")
        End With
End Sub
```

```
void SetArchs()
{
    m_spel.Arch(1, 30, 30);
    m_spel.Arch(2, 60, 60);
    m_spel.Jump("P1 C1");
    m_spel.Jump("P2 C2");
}
```

15.3.12 Arm方法, Spel類別

描述

選擇目前機器人手臂。

語法

Sub Arm (ArmNumber As Integer)

參數

 ArmNumber 整數值或運算式(有效範圍:0~15)。最多可選擇16種不同的手臂。手臂0是標準(預設)機器人手臂。手臂1-15 是透過ArmSet指令定義的輔助手臂。

另請參閱

ArmSet, GetArm, Tool

Arm範例

VB 例:

m_spel.Arm(1)

C# 例:

m_spel.Arm(1);

15.3.13 ArmClr方法 · Spel類別

描述

清除(取消定義)目前機器人的手臂。

語法

Sub ArmClr (ArmNumber As Integer)

參數

 ArmNumber 介於1-15的整數運算式。手臂0是標準(預設)機器人手臂,且無法清除。手臂1-15是透過ArmSet指令定義的輔助 手臂。

另請參閱

ArmSet, GetArm, Tool

ArmClr範例

VB 例:

m_spel.ArmClr(1)

C# 例:

m_spel.ArmClr(1);

15.3.14 ArmDef方法, Spel類別

描述

傳回機器人手臂是否定義。

語法

Function ArmDef (ArmNumber As Integer) As Boolean

參數

 ArmNumber 0~15的整數值。手臂0是始終設定的標準(預設)機器人手臂。手臂1-15是使用ArmSet方法定義的輔助手臂。

傳回值

若指定手臂已定義,會傳回True,否則會傳回False。

另請參閱

ArmSet, GetArm, Tool

ArmDef範例

VB 例:

x = m_spel.ArmDef(1)

C# 例:

x = m_spel.ArmDef(1);
15.3.15 ArmSet方法 · Spel類別

描述

定義輔助的機器人手臂。

語法

Sub ArmSet (ArmNumber As Integer, Param1 As Single, Param2 As Single, Param3 As Single, Param4 As Single, Param5 As Single)

參數

- ArmNumber 整數:有效範圍介於1-15。
- Param1

(SCARA機器人適用)肘部中心線到新方向軸中心線的水平距離。(亦即新輔助手臂方向軸中心線所在的位置。) (Cartesian機器人適用)原始X位置的X軸方向位置偏移(公釐)。

Param2

(SCARA機器人適用)從標準肘部中心線與標準方向軸中心線所形成的直線,到新輔助手臂肘部中心線與新方向軸 中心線所形成的直線,兩者之間的偏移(度)。(這兩條直線應該在肘部中心線相交,形成的角度即為Param2。) (Cartesian機器人適用)原始Y位置的Y軸方向位置偏移(公釐)。

Param3

(SCARA & Cartesian機器人適用)新方向軸中心與舊方向軸中心之間的Z高度偏移差距。(此為距離。)

Param4

(SCARA機器人適用)肩部中心線到新輔助軸肘部方向之肘部中心線的距離。(Cartesian機器人適用)此為虛擬參數 (指定0)

Param5

(SCARA & Cartesian機器人適用)新方向軸與舊方向軸的角度偏移(度)。

另請參閱

Arm, Tool, TLSet

ArmSet範例

VB 例:

```
Sub SetArms()
    With m_spel
        .ArmSet(1, 1.5, 0, 0, 0, 0)
        .ArmSet(2, 3.2, 0, 0, 0, 0)
        End With
End Sub
```

```
void SetArms()
{
    m_spel.ArmSet(1, 1.5, 0, 0, 0, 0);
    m_spel.ArmSet(2, 3.2, 0, 0, 0, 0);
}
```

15.3.16 Atan方法 · Spel類別

描述

傳回數值運算式的圓弧正切函數。

語法

Function Atan (number As Double) As Double

參數

number
 代表角度值之正切函數的數值運算式。

傳回值

指定值的圓弧正切函數

另請參閱

Atan2

Atan範例

VB 例:

Dim angle As Doubleangle = m_spel.Atan(.7)

C# 例:

double angle;

angle = m_spel.Atan(.7);

15.3.17 Atan2方法, Spel類別

描述

傳回虛線連接點(0,0)和(X,Y)的角度(以徑度表示)。

語法

Function Atan2 (Dx As Double, Dy as Double) As Double

參數

- Dx 代表X座標的數值運算式。
- Dy 代表Y座標的數值運算式。

傳回值

包含角度的雙精度值。

另請參閱

Atan

Atan2範例

VB 例:

Dim angle As Double

angle = m_{spel} .Atan2(-25, 50)

C# 例:

double angle;

```
angle = m_{spel}.Atan2(-25, 50);
```

15.3.18 ATCLR 方法 · Spel類別

描述

清除并初始化關節的有效轉矩。

語法

Sub ATCLR ()

另請參閱

ATRQ, PTCLR, PTRQ

ATCLR範例

VB 例:

m_spel.ATCLR()

C# 例:

m_spel.ATCLR();

15.3.19 AtHome方法, Spel類別

描述

若目前機器人位於起始點位置,會傳回True。

語法

Function AtHome () As Boolean

傳回值

若目前機器人位於起始點位置,會傳回True,否則會傳回False。

另請參閱

Home

AtHome範例

VB 例:

```
If m_spel.AtHome() Then
    lblCurPos.Text = "Robot is at home position"
Else
    lblCurPos.Text = "Robot is not at home position"
End If
```

15.3.20 ATRQ 方法, Spel類別

描述

傳回指定關節的有效轉矩。

語法

Function ATRQ (JointNumber As Integer) As Single

參數

JointNumber
 管道#的整數值(範圍:1~機器人的關節數)

傳回值

0~1的實數值。

<mark>另請参閲</mark> ATCLR, PTCLR, PTRQ

ATRQ範例

VB 例:

```
Dim val As Single
Dim i As Integer
For i = 1 To 4
val = m_spel.ATRQ(i)
Next i
```

```
float avgTorque;
for(int i = 1; i <= 4; i++)
   avgTorque = m_spel.ATRQ(i);
```

15.3.21 AvgSpeed 方法, Spel類別

描述

傳回指定關節速度的平均絕對值。

語法

Function AvgSpeed (JointNumber As Integer) As Single

參數

JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)

傳回值

0~1的實數值。

備註

本方法傳回指定關節速度的平均絕對值。本方法可以顯示馬達的負載狀態。結果由介於0到1之間的實數表示。最大平均速度是1。

在執行本方法前,必須先執行AvgSpeedClear方法。

本方法具有時間限制。執行AvgSpeedClear方法后,請在60秒内執行本方法。超過60秒,就會出現錯誤4008。

對於虛擬控制器和乾運轉,從命令速度而不是實際速度,計算速度的平均絕對值。 本方法不支援PG附加軸。

另請參閱

AvgSpeedClear, PeakSpeed, PeakSpeedClear

AvgSpeed範例

VB 例:

```
Dim val As Single
Dim i As Integer
For i = 1 To 4
val = m_spel.AvgSpeed(i)
Next i
```

```
float avgSpeed;
for(int i = 0; i <=4; i++)
    avgSpeed = m_spel.AvgSpeed(i);
```

15.3.22 AvgSpeedClear 方法, Spel類別

描述

清除並初始化關節速度的平均絕對值。

語法

Sub AvgSpeedClear ()

備註

本方法清除指定關節速度的平均絕對值。 執行AvgSpeed方法前,必須先執行本方法。 本方法不支援PG附加軸。

另請參閱

AvgSpeed, PeakSpeed, PeakSpeedClear

AvgSpeedClear範例

VB 例:

m_spel.AvgSpeedClear()

C# 例:

m_spel.AvgSpeedClear();

15.3.23 AxisLocked方法, Spel類別

描述

若指定軸受伺服系統控制,會傳回True。

語法

Function AxisLocked (AxisNumber As Integer) As Boolean

參數

 AxisNumber 代表管道#的數值運算式。數值可以介於1-9。

傳回值

若指定軸受伺服系統控制,會傳回True。

另請參閱

SLock, SFree

AxisLocked範例

VB 例:

```
If m_spel.AxisLocked(1) Then
    lblAxis1.Text = "Robot axis \#1 is locked"
Else
    lblAxis1.Text = "Robot axis \#1 is free"
End If
```

```
if (m_spel.AxisLocked(1))
    lblAxis1.Text = "Robot axis \#1 is locked";
else
    lblAxis1.Text = "Robot axis \#1 is free";
```

15.3.24 Base方法, Spel類別

描述

定義基座座標系統。

語法

Sub Base (OriginPoint As SpelPoint [, XAxisPoint As SpelPoint] [, YAxisPoint As SpelPoint] [, Alignment As SpelBaseAlignment])

參數

- OriginPoint 代表基座座標系統之原點的SpelPoint。
- XAxisPoint 位於基座座標系統X軸上任何位置的SpelPoint。
- YAxisPoint 位於基座座標系統Y軸上任何位置的SpelPoint。
- Alignment 當提供XAxisPoint和YAxisPoint參數時,請使用Alignment參數指定要與基座對齊的軸。(可省略)

另請參閱

Local

Base範例

VB 例:

```
Dim originPoint As New SpelPoint
originPoint.X = 50
originPoint.Y = 50
m_spel.Base(originPoint)
```

```
SpelPoint originPoint = new SpelPoint();
originPoint.X = 50;
originPoint.Y = 50;
m_spel.Base(originPoint);
```

15.3.25 BGo方法, Spel類別

描述

在所選本地座標系統中執行PTP相對動作。

語法

Sub BGo (PointNumber As Integer) Sub BGo (Point As SpelPoint) Sub BGo (Point As SpelPoint, AttribExpr As String) Sub BGo (PointExpr As String)

參數

每個語法都有一個參數,用以指定在BGo動作期間手臂移動的結束點。此為PTP動作結束時的最終位置。

- PointNumber
 透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
- Point
 透過使用SpelPoint資料類型來指定結束點。
- AttribExpr 透過使用字串運算式來指定結束點屬性。
 可包括CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。
- PointExpr 透過使用字串運算式來指定結束點。
 可包括CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。

另請參閱

Accel, Speed Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move BMove, TGo, TMove Find, Till

BGo範例

VB 例:

```
使用點編號指定的點
m_spel.Tool(1)
m_spel.BGo(100)

使用SpelPoint指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.BGo(pt)

使用屬性式
m_spel.BGo(pt, "Till")

使用點運算式指定的點
m_spel.BGo("P0 /L /2")

使用平行處理
m_spel.BGo("P1 \!D50; On 1; D90; Off 1\!")
```

' 使用標籤指定的點 m_spel.BGo("pick")

```
// 使用點編號指定的點
m spel.Tool(1);
m spel.BGo(100);
// 使用SpelPoint指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P*");
pt.X = 125.5;
m_spel.BGo(pt);
// 使用屬性式
m_spel.BGo(pt, "Till");
// 使用點運算式指定的點
m_spel.BGo("P0 /L /2");
m_spel.BGo("P1 :Z(-20)");
// 使用平行處理
m_spel.BGo("P1 \!D50; On 1; D90; Off 1\!");
// 使用標籤指定的點
m_spel.BGo("pick");
```

15.3.26 BMove方法 · Spel類別

描述

在所選本地座標系統中執行線性插補相對動作。

語法

Sub BMove (PointNumber As Integer) Sub BMove (Point As SpelPoint) Sub BMove (Point As SpelPoint, AttribExpr As String) Sub BMove (PointExpr As String)

參數

每個語法都有一個參數,用以指定在BMove動作期間手臂移動的結束點。此為線性插補動作結束時的最終位置。

- PointNumber
 透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
- Point
 透過使用SpelPoint資料類型來指定結束點。
- AttribExpr 透過使用字串運算式來指定結束點屬性。
 可包括ROT、CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。
- PointExpr 透過使用字串運算式來指定結束點。
 可包括ROT、CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move BGo, TGo, TMove Find, Till

Move範例

VB 例:

```
使用點編號指定的點
m_spel.Tool(1)
m_spel.BMove(100)

使用SpelPoint指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.BMove(pt)

使用點運算式指定的點
m_spel.BMove("P0 /L /2 ROT")
m_spel.BMove("P1 :Z(-20)")

使用平行處理
m_spel.BMove("P1 \!D50; On 1; D90; Off 1\!")

使用標籤指定的點
m_spel.BMove("pick")
```

// 使用點編號指定的點
m_spel.Tool(1);
m_spel.BMove(100);

// 使用SpelPoint指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P*");
pt.X = 125.5;
m_spel.BMove(pt);

// 使用點運算式指定的點 m_spel.BMove("P0 /L /2");

// 使用平行處理 m_spel.BMove("P1 \!D50; On 1; D90; Off 1\!");

// 使用標籤指定的點 m spel.BMove("pick");

15.3.27 Box方法 · Spel類別

描述

指定在盒體內定義的接近檢查區域。

語法

Sub Box (AreaNumber As Integer, MinX As Single, MaxX As Single, MinY As Single, MaxY As Single, MinZ As Single, MaxZ As Single)

Sub Box (AreaNumber As Integer, MinX As Single, MaxX As Single, MinY As Single, MaxY As Single, MinZ As Single, MaxZ As Single, PolarityOn As Boolean)

參數

- AreaNumber 1~15的整數值,代表所設定區域的編號。
- MinX
 接近檢查區域的最小X座標位置。
- MaxX 接近檢查區域的最大X座標位置。
- MinY 接近檢查區域的最小Y座標位置。
- MaxY

數值或運算式,用於指定所設定區域的上限位置Y座標(實數)。

- MinZ 數值或運算式,用於指定所設定區域的下限位置Z座標(實數)。
- MaxZ 數值或運算式,用於指定所設定區域的上限位置Z座標(實數)。
- PolarityOn
 可以省略設定在使用對應遠端輸出時的遠端輸出邏輯。若要在夾具末端位於盒體區域中時將I/O輸出設為開啟,
 請使用True。若要在夾具末端位於盒體區域中時將I/O輸出設為關閉,請使用False。

另請參閱

BoxClr, BoxDef, Plane

Box範例

VB 例:

m_spel.Box(1, -5, 5, -10, 10, -20, 20)

C# 例:

m_spel.Box(1, -5, 5, -10, 10, -20, 20);

15.3.28 BoxClr方法 · Spel類別

描述

清除box(接近檢查區域)的定義。

語法

Sub BoxClr (BoxNumber As Integer)

參數

 BoxNumber 介於1至15的整數運算式,代表區域編號。

另請參閱

Box, BoxDef

BoxClr範例

VB 例:

m_spel.BoxClr(1)

C# 例:

m_spel.BoxClr(1);

15.3.29 BoxDef方法, Spel類別

描述

傳回Box是否定義。

語法

Function BoxDef (BoxNumber As Integer) As Boolean

參數

 BoxNumber 介於1至15的整數運算式,代表區域編號。

傳回值

若指定box已定義, 會傳回True, 否則會傳回False。

另請參閱

Box, BoxClr

BoxDef範例

VB 例:

x = m_spel.BoxDef(1)

C# 例:

x = m_spel.BoxDef(1);

15.3.30 Brake 方法, Spel類別

描述

讀取或設定關節制的制動狀態。

語法

Sub Brake (JointNumber As Integer, State As Boolean) Function Brake (JointNumber As Integer) As Boolean

參數

- JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)
- State
 開啓制動器:使用On。解除制動器:使用Off。

傳回值

0 = 制動器Off

1 = 制動器On

備註

本方法可以對垂直六軸型機器人(含N系列)的單個關節開啓或解除制動。此方法旨在僅供維護人員使用。

執行本方法, 會初始化機器人的控制參數。

▲ 警告

解除制動器時要小心。確保關節被正確固定。如果關節沒有正確固定,則可能導致墜落,從而導致機器人故障或操 作員受傷。

\Lambda 注意

在執行Brake Off命令之前,請確保將緊急停止開關放在手邊。

當控制器處於緊急停止狀態時,馬達制動器將被鎖定。執行Brake Off命令時,機械臂可能會由於自生重量而下降。

另請參閱

Reset, SFree, SLock

Brake範例

VB 例:

```
Dim state As Boolean
state = m spel.Brake(1)
```

```
bool state;
state = m spel.Brake(1);
```

15.3.31 BTst方法 · Spel類別

描述

以數值傳回1位元的狀態。

語法

Function BTst (Number As Integer, BitNumber As Integer) As Boolean

參數

- Number 以運算式或數值指定位元測試的數字。
- BitNumber 指定要測試的位元(介於0至31的整數)。

傳回值

若指定位元已設定、會傳回True、否則會傳回False。

另請參閱

On, Off

BTst範例

VB 例:

 $x = m_spel.BTst(data, 2)$

C# 例:

x = m_spel.BTst(data, 2);

15.3.32 BuildProject方法, Spel類別

描述

建置透過Project屬性所指定的Epson RC+ 8.0專案。

語法

Sub BuildProject ()

另請參閱

Project, ProjectBuildComplete, ProjectOverwriteWarningEnabled

BuildProject範例

VB 例:

```
With m_spel
    .Project = "c:\EpsonRC80\projects\myproj\myproj.sprj"
    If Not .ProjectBuildComplete() Then
        .BuildProject()
    End If
End With
```

```
m_spel.Project = @"c:\EpsonRC80\projects\myproj\myproj.sprj";
if(\!m_spel.ProjectBuildComplete())
    m spel.BuildProject();
```

15.3.33 Call方法, Spel類別

描述

調用(執行)可選擇性傳回數值的SPEL+函數。

語法

Function Call (FuncName As String [, Parameters As String]) As Object

參數

 FuncName 叫用的函數名稱。
 Parameters 選用。指定引數清單。使用以逗號(,)分隔的引數。

傳回值

SPEL+函數的傳回值。 資料類型符合函數的資料類型。

備註

使用Call方法調用SPEL+函數並擷取傳回值。將Call的結果指派至變數時,請確定使用正確的資料類型,否則會發生 類型不相符錯誤。

您也可以從Visual Basic應用程式調用在SPEL+程式碼中宣告的DLL函數。

\Lambda 注意

如欲停止以Call方法執行的函數時,請使用Stop方法。 若以Stop方法停止,則Call方法的傳回值為null。 以Call方法執行的函數,無法透過Pause、Halt方法暫停任務。 如欲暫停,請使用Xqt方法。

另請參閱

Xqt

Call範例

```
' Visual Basic程式碼
Dim errCode As Integer
errCode = m spel.Call("GetPart", """Test"",2")
// C# 程式碼
int errCode;
errCode = m spel.Call("GetPart", """Test"", 2");
' SPEL+ 函數
Function GetPart (Info$ As String, Timeout As Integer) As Integer
   Long errNum
OnErr GoTo GPErr
   Print Info$
   errNum = 0
   Jump Pl
   On vacuum
   Wait Sw(vacOn) = On, Timeout
   If TW = True Then
        errNum = VAC TIMEOUT
   EndIf
```

```
GetPart = errNum
Exit Function
GPErr:
GetPart = Err
Fend
```

15.3.34 CalPls 方法, Spel類別

描述

讀取或設定用於校準的位置姿態脈衝值。

語法

Function CalPls (JointNumber As Integer) As Integer

Sub CalPls (J1Pulses As Integer, J2Pulses As Integer, J3Pulses As Integer, J4Pulses As Integer, [J5Pulses As Integer], [J6Pulses As Integer], [J7Pulses As Integer], [J8Pulses As Integer], [J9Pulses As Integer])

參數

 J1Pulses - J9Pulses 代表第1~9關節的脈衝值的整數 J5Pulses - J9Pulses可以省略

傳回值

如果省略脈衝值,則顯示當前設定的脈衝值。

備註

輸入並保存正確的脈衝值,用於校準位置。

本方法僅供維護時使用。當馬達的原點不匹配機械臂的機械原點時(例如更換馬達時)使用。匹配原點的操作稱爲校準。

在正常情況下,校準位置的脈衝值與本方法中設定的脈衝值匹配。但在維護(例如更換馬達)後,這兩個值不匹配,需 要校準。

校準的一種方法是,將關節移動到特定位置並執行Calib。執行Calib會將校準特定位置的脈衝值,更改為本方法中設定的脈衝值(校準位置的正確脈衝值)。

必須設置Hofs值才能執行校準。要自動計算Hofs值,請先將關節移動到特定的位置,然後執行Calib。控制器會根據 校準位置的脈衝值,自動計算Hofs的值。

\Lambda 注意

CalPls值不能透過關閉電源變更

關閉控制器然後重新啟動時,CalPls的值不會被初始化。更改CalPls值的唯一方法是執行Calib命令。

另請參閱

Hofs

CalPls範例

VB 例:

```
Dim val As Single
Dim i As Integer
For i = 1 To 4
val = m_spel.CalPls(i)
Next i
```

```
float val;
for(int i = 1; i <= 4; i++)
        val = m_spel.CalPls(i);
```

15.3.35 ClearPoints方法 · Spel類別

描述

清除目前機器人記憶體中的點。

語法

Sub ClearPoints ()

另請參閱

LoadPoints, Robot, SavePoints, SetPoint

ClearPoints範例

VB 例:

```
With m_spel
    .ClearPoints()
    .SetPoint(1, 100, 200, -20, 0, 0, 0)
    .Jump(1)
End With
```

```
m_spel.ClearPoints();
m_spel.SetPoint(1, 100, 200, -20, 0, 0, 0);
m_spel.Jump(1);
```

Rev.2

15.3.36 Connect方法, Spel類別

描述

連接Spel類別執行個體與控制器。

語法

Sub Connect (ConnectionName As String) Sub Connect (ConnectionName As String, ConnectionPassword As String) Sub Connect (ConnectionNumber As Integer) Sub Connect (ConnectionNumber As Integer, ConnectionPassword As String)

參數

- ConnectionName 表示連接名稱的字串。
- ConnectionNumber 代表連接編號的整數值。連接編號是Epson RC+中[安裝] - [電腦與控制器連接]對話方塊的編號。若要使用上次 成功連接的目標,請指定 -1。
- ConnectionPassword 代表連線密碼的字串。若控制器具有用於連接控制器的密碼,並且密碼未在Epson RC+[安裝]-[電腦與控制器連 接]對話方塊中設定,則必須設定用於連接控制器的密碼。

備註

當Spel類別執行個體需要與控制器進行通訊時,它會自動連線。若要明確地連線至控制器,請使用Connect方法。

另請參閱

Disconnect, Initialize

\Lambda 注意

如果在[電腦與控制器通信]對話框中更改連接號,則此方法的 "ConnectionNumber" 參數值將受到影響。請檢查 RC+API端的程式。

Connect範例

VB 例:

```
Try
  m_spel.Connect(1)
Catch ex As RCAPINet.SpelException
  MsgBox(ex.Message)
End Try
```

```
try{
    m_spel.Connect(1);
}
catch(RCAPINet.SpelException ex){
    MessageBox.Show(ex.Message);
}
```

Rev.2

15.3.37 Continue方法, Spel類別

描述

發生暫停時,恢復控制器的所有任務。

語法

Sub Continue ()

備註

使用Continue可恢復因Pause方法或安全防護打開而暫停的所有任務。 當安全防護在任務執行期間打開時,機器人將會減速至停止狀態,且機器人馬達將會關閉。關閉安全防護後,您可使 用Continue恢復週期。

另請參閱

Pause, Start, Stop

Continue範例

VB 例:

```
Sub btnContinue_Click( _______ByVal sender As System.Object, _______ByVal e As System.EventArgs) Handles btnContinue.Click
    btnPause.Enabled = True
    btnContinue.Enabled = False
    Try
    m_spel.Continue()
    Catch ex As RCAPINet.SpelException
    MsgBox(ex.Message)
    End Try
End Sub
```

```
C# 例:
```

```
void btnContinue_Click(object sender, EventArgs e)
{
    btnPause.Enabled = true;
    btnContinue.Enabled = true;
    try{
        m_spel.Continue();
     }
        catch(RCAPINet.SpelException ex){
        MessageBox.Show(ex.Message);
     }
}
```

15.3.38 Ctr方法, Spel類別

描述

傳回指定輸入計數器的計數器值。

語法

Function Ctr (BitNumber As Integer) As Integer

參數

 BitNumber 被設為計數器的輸入位編號。僅可同時啟用16個計數器。

傳回值

傳回計數器值。(0至65535的整數)

另請參閱

CtReset

Ctr範例

VB 例:

lblCounter.Text = m_spel.Ctr(1).ToString()

C# 例:

lblCounter.Text = m_spel.Ctr(1).ToString();

15.3.39 CtReset方法, Spel類別

描述

重置指定輸入計數器的計數器值。同時將輸入定義為計數器輸入。

語法

Sub CtReset (BitNumber As Integer)

參數

 BitNumber 被設為計數器的輸入位編號。僅可同時啟用16個計數器。

另請參閱

Ctr

CtReset範例

VB 例:

m_spel.CtReset(2)

C# 例:

m_spel.CtReset(2);

15.3.40 Curve方法 · Spel類別

描述

定義沿著曲徑移動手臂所需的資料與點。可以在路徑中定義許多點資料,以改善路徑的精確度。

語法

Sub Curve (FileName As String, Closure As Boolean, Mode As Integer, NumOfAxis As Integer, PointList As String)

參數

FileName

指定代表點資料保存檔案之路徑和名稱的字串的檔案末尾將加上.crv副檔名。當執行Curve指令時,將會建立 fileName。

Closure

指定是否要將路徑的最後一個點連接至第一個點的Boolean運算式。

Mode

指定是否要在U軸的正切方向自動插補手臂。

模式設定	正切修正
0	No
2	Yes

NumOfAxes

介於2-4的整數運算式,用以指定曲線動作期間受控軸的數量,如下所述: 2:在沒有Z軸移動或U軸旋轉的情況下,於XY平面產生曲線。3:在沒有U軸旋轉的情況下,於XYZ平面產生曲 線。(Theta 1、Theta2及Z)4:在U軸旋轉的情況下,於XYZ平面產生曲線。(控制4個軸)

pointList

{ point expression | P(start:finish) } [, output command] ...

此參數實際上是連續的點編號,以及用逗號分隔的選用輸出陳述式,或是以冒號分隔的遞增點範圍。如果點資料 完整並依遞增或遞減順序列出,可使用一個冒號合併兩個點編號,例如P(1:5)。以逗號分隔的連續點通常如下所 示:

Curve MyFile, O, 0, 4, P1, P2, P3, P4

或以冒號指定,如下所示:

Curve MyFile, O, 0, 4, P(1:4)

上述範例是使用 P1、P2、P3、P4 來指定曲線。輸出指令可省略,在曲線動作中控制輸出操作時使用。此指令指定 I/O 或記憶體 I/O 的開啟或關閉。輸出指令會在前一個機械手臂通過連續點數據的特定點後執行。1 個 Curve 陳述式中最多可包含 16 個輸出指令。下述範例是機械手臂通過 P2 後執行「On 2」指令,之後機械手臂通過從P3 至 P10 的所有點。Curve "MyFile", C, 0, 4, P1, P2, ON 2, P(3:10)

備註

使用Curve來定義以CVMove方法執行的樣條路徑。如需詳細資訊,請參閱SPEL+的Curve命令。

另請參閱

Curve (SPEL+ Statement), CVMove Method

Curve範例

VB 例:

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1, P(4:7)")
m_spel.CVMove("mycurveFile")
```

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1, P(4:7)");
m_spel.CVMove("mycurveFile");
```

15.3.41 CVMove方法 · Spel類別

描述

執行由Curve指令定義的連續樣條路徑動作。

語法

Sub CVMove (FileName As String [, OptionList As String])

參數

- FileName 字串,代表透過Curve命令建立的檔案名稱。
- OptionList 包含Till規格的字串運算式。

備註

使用CVMove來執行以Curve方法定義的路徑。如需詳細資訊,請參閱SPEL+命令的CVMove。若需要以CP執行 CVMove,建議您從SPEL+任務來執行CVMove,而不要從應用程式執行。這是因為要讓CP動作正確執行,系統必 須事先知道下一個動作目標的位置。由於一次只能執行一個RC+ API命令,因此系統無法事先知道下一個目標的位 置。

另請參閱

Curve, CVMove(SPEL+)

CVMove範例

VB 例:

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1, P(4:7)")
m_spel.CVMove("mycurveFile", "CP Till Sw(1) = 1")
m_spel.CVMove("mycurveFile")
```

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1, P(4:7)");
m_spel.CVMove("mycurveFile", "CP Till Sw(1) = 1");
m_spel.CVMove("mycurveFile");
```

15.3.42 CX、CY、CZ、CU、CV、CW、CR、CS、CT方法, Spel類別

描述

從一點擷取座標值。- CV和CW用於6軸機器人- CS和CT用於附加軸 - CR用於Joint 7軸機器人

語法

Function CX (PointExpr As String) As Single Function CY (PointExpr As String) As Single

Function CZ (PointExpr As String) As Single

Function CU (PointExpr As String) As Single

Function CV (PointExpr As String) As Single

Function CW (PointExpr As String) As Single

Function CR (PointExpr As String) As Single

Function CS (PointExpr As String) As Single

Function CT (PointExpr As String) As Single

參數

PointExpr

字串運算式,用於指定搜索指定座標的點。只要從P0開始,即可使用任何有效的點運算式。P*也可以用於從目前 位置擷取座標。

傳回值

指定的座標值。 CX、CY、CZ的傳回值:實際值(公釐) CU、CV、CW的傳回值:實際值(度) CR、CS、CT的傳回值:實際值

另請參閱

GetPoint, SetPoint

CX、CY、CZ、CU、CV、CW、CR、CS、CT範例 VB例:

```
Dim x As Single, y As Single
x = m_spel.CX("P1")
y = m_spel.CY("P*")
```

```
float x, y;
x = m_spel.CX("P1");
y = m_spel.CY("P*");
```

15.3.43 Delay方法 · Spel類別

描述

指定的延遲時間(毫秒)。

語法

Sub Delay (Milliseconds As Integer)

參數

 Milliseconds 延遲的毫秒數。

Delay範例

VB 例:

m_spel.Delay(500)

C# 例:

m_spel.Delay(500);

15.3.44 DegToRad方法 · Spel類別

描述

將度轉換成徑度。

語法

Function DegToRad (degrees As Double) As Double

參數

degrees
 轉換成徑度的度數。

傳回值

包含徑度的雙精度值。

另請參閱

RadToDeg

DegToRad範例

VB 例:

Dim rad As Double

rad = m_spel.DegToRad(45)

C# 例:

double rad;

rad = m_spel.DegToRad(45);

15.3.45 Disconnect方法, Spel類別

描述

從目前連線中斷Spel類別執行個體的連線。

語法

Sub Disconnect ()

備註

使用Disconnect可中斷目前控制器的連線。

另請參閱

Connect, Initialize

Disconnect範例

VB 例:

```
Try
  m_spel.Disconnect()
Catch ex As RCAPINet.SpelException
  MsgBox(ex.Message)
End Try
```

```
try{
    m_spel.Disconnect();
}
catch(RCAPINet.SpelException ex){
MessageBox.Show(ex.Message);}
```
15.3.46 ECP方法, Spel類別

描述

選擇目前的ECP定義。

語法

Sub ECP (ECPNumber As Integer)

參數

 ECPNumber 0~15的整數值,用於在下一個動作命令從16個ECP設定中指定使用哪一個(可省略)。ECP 0時會停用ECP選 擇。

另請參閱

ECPSet

ECP範例

VB 例:

```
m_spel.ECP(1)
m_spel.Move("P1 ECP")
```

```
m_spel.ECP(1);
m_spel.Move("P1 ECP");
```

15.3.47 ECPCIr方法 · Spel類別

描述

清除(取消定義)目前機器人的外部控制點。

語法

Sub ECP (ECPNumber As Integer)

參數

 ECPNumber 整數值,代表要清除的ECP編號(ECP 0為預設值,無法清除)。

另請參閱

ECP, ECPDef

ECP範例

VB 例:

m_spel.ECPClr(1)

C# 例:

m_spel.ECPClr(1);

15.3.48 ECPDef方法, Spel類別

描述

傳回ECP定義狀態。

語法

Function ECPDef (ECPNumber As Integer) As Boolean

參數

 ECPNumber 代表要傳回狀態之ECP的整數值。

傳回值

若指定ECP已定義,會傳回True,否則會傳回False。

另請參閱

ECP, ECPClr

ECP範例

VB 例:

x = m_spel.ECPDef(1)

C# 例:

x = m_spel.ECPDef(1);

15.3.49 ECPSet方法, Spel類別

描述

定義ECP(外部控制點)。

語法

Sub ECPSet (ECPNumber As Integer, Point As SpelPoint) Sub ECPSet (ECPNumber As Integer, XCoord as Double, YCoord as Double, ZCoord as Double, UCoord as Double [, VCoord As Double] [, WCoord as Double])

參數

- ECPNumber
 是一個代表要定義的外部控制點的運算式,或者1~15的整數運算式
- Point
 包含點資料的SpelPoint。
- XCoord 外部控制點X座標。
- YCoord 外部控制點Y座標。
- ZCoord 外部控制點Z座標。
- UCoord 外部控制點U座標。
- VCoord 外部控制點V座標(可省略)
- WCoord 外部控制點W座標(可省略)

另請參閱

ArmSet, ECP, GetECP, TLSet

ECPSet範例

VB 例:

m_spel.ECPSet(1, 100.5, 99.3, 0, 0)

C# 例:

m_spel.ECPSet(1, 100.5, 99.3, 0, 0);

15.3.50 EnableEvent方法, Spel類別

描述

針對EventReceived事件啟用特定系統事件。

語法

Sub EnableEvent (EventNumber As RCAPINet.SpelEvents, Enabled as Boolean)

參數

- EventNumber 要啟用或停用的事件。
- Enabled
 設為True可啟用事件,設為False則停用事件。

另請參閱

EventReceived

EnableEvent範例

VB 例:

```
With m_spel
   .EnableEvent(RCAPINet.SpelEvents.ProjectBuildStatus, True)
   .BuildProject()
End With
```

```
m_spel.EnableEvent(RCAPINet.SpelEvents.ProjectBuildStatus, true);
m_spel.BuildProject();
```

15.3.51 ExecuteCommand方法, Spel類別

描述

將命令傳送至Epson RC+ 8.0並等待完成。

語法

Sub ExecuteCommand (Command As String [, ByRef Reply As String])

參數

- Command
 包含SPEL+命令的字串。
- Reply 傳回選擇性的回覆。

備註

一般而言,不需要使用ExecuteCommand。執行Spel方法可以執行大部分的操作。不過,有時還是需執行SPEL+多 重陳述式。多重陳述式是一行命令,包含以分號分隔的多個陳述式。使用ExecuteCommand可執行多重陳述式。

例: SPELCom1.ExecSPELCmd "JUMP pick; ON tipvac" 最大命令行長度為200個字元。

另請參閱

Pause

ExecuteCommand範例

VB 例:

```
m_spel.ExecuteCommand("JUMP P1\!D50; ON 1\!")
```

C# 例:

m_spel.ExecuteCommand("JUMP P1\!D50; ON 1\!");

描述

傳回指定的現場匯流排的狀態。

語法

Function FBusIO_GetBusStatus (BusNumber As Integer) As Integer

參數

BusNumber
 現場匯流排系統的編號的整數值。此數字始終為16,是連接到控制器PC側的現場匯流排主機板的總線的ID。

傳回值

0 - OK

- 1-未連接
- 2 關機

備註

注意 僅當啓用了現場匯流排主機板選配件功能時,才可以用本方法。

另請參閱

FBusIO_GetDeviceStatus, FBusIO_SendMsg, IsOptionActive

FBusIO_GetBusStatus 範例

VB 例:

```
Dim val As Integer
val = m_spel.FBusIO_GetBusStatus(16)
```

```
int busStatus;
busStatus = m spel.FbusIO GetBusStatus(16);
```

15.3.53 FBusIO_GetDeviceStatus 方法, Spel類別

描述

傳回指定的現場匯流排裝置的狀態。

語法

Function FBusIO_GetDeviceStatus (BusNumber As Integer, DeviceID As Integer) As Integer

參數

- BusNumber 現場匯流排系統的編號的整數值。此數字始終為16,是連接到控制器PC側的現場匯流排主機板的總線的ID。
 DeviceID
 - 代表裝置的現場匯流排ID的整數值

傳回值

0 - OK

- 1 未連接
- 2 關機
- 3-同步錯誤:裝置正在初始化,或裝備的傳輸速率不正確。

備註

注意

僅當啓用了現場匯流排主機板選配件功能時,才可以用本方法。

另請參閱

FBusIO_GetBusStatus, FBusIO_SendMsg, IsOptionActive

FBusIO_GetDeviceStatus 範例

VB 例:

```
Dim val As Integer
val = m spel.FBusIO GetDeviceStatus(16, 10)
```

```
dev
ideStatus = m_spel.FbusIO_GetDeviceStatus(16, 10);
```

15.3.54 FBusIO_SendMsg 方法, Spel類別

描述

向現場匯流排I/O裝置發送訊息,並傳回回答。

語法

Sub FBusIO_SendMsg (BusNumber As Integer, DeviceID As Integer, MsgParam As Integer, SendData As Byte(), ByRef RecvData As Byte())

參數

- BusNumber
 現場匯流排系統的編號的整數值。此數字始終為16,是連接到控制器PC側的現場匯流排主機板的總線的ID。
- DeviceID
 代表裝置的現場匯流排ID的整數值
- MsgParam
 代表訊息參數的整數運算式。不可用於DeviceNet。
- SendData 在位元組類型的陣列中,指定發送給裝置的資料。此陣列尺寸必須和發送的位元組尺寸相同。如果不發送訊息, 請指定為"0"。
- RecvData
 在位元組類型的陣列中,指定從裝置接收的資料。此陣列會根據接收到的位元組數,自動轉換尺寸大小。

備註

注意

僅當啓用了現場匯流排主機板選配件功能時,才可以用本方法。

另請參閱

FBusIO_GetBusStatus, FBusIO_GetDeviceStatus, IsOptionActive

FBusIO_SendMsg 範例

VB 例:

```
'向DeviceNet發送訊息
Dim recvData() as Byte
Dim sendData(6) as Byte
Array.Clear(sendData, 0, sendData.Length)
sendData(0) = 14 '命令
sendData(1) = 1 '類別
sendData(3) = 1 '範例
sendData(5) = 7 '屬性
' DeviceNet的 MsgParam是0
m_spel.FbusIO_SendMsg(16, 1, 0, sendData, recvData)
' 向Profibus發送訊息
Dim recvData() As Byte;
m_spel.FbusIO_SendMsg(16, 1, 56, Nothing, recvData);
```

```
// 向DeviceNet發送訊息
byte[] sendData, recvData;
byte[] sendData = new byte[6];
Array.Clear(sendData, 0, sendData.Length);
```

sendData[0] = 14; //命令
sendData[1] = 1; //類別
sendData[3] = 1; //範例
sendData[5] = 7; //屬性
// DeviceNet的MsgParam是0
m_spel.FbusIO_SendMsg(16, 201, 0, sendData, out recvData);
// 向Profibus發送訊息
byte[] recvData;

m_spel.FbusIO_SendMsg(16, 1, 56, null, out recvData);

15.3.55 FGGet 方法, Spel類別

描述

取得力覺引導序列,或者力覺引導物件的結果。

語法

Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As Boolean) Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As Double) Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As Integer) Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As String)

參數

- Sequence 力覺引導序列的名稱,或者代表力覺引導序列名稱的字串變數
- Object
 Force Guide對象名稱或代表Force Guide對象名稱的字串變數。取得Force Guide序列的結果時可省略。
- Property 獲得值的結果的名稱
- Result
 代表傳回值的變數。數字和類型因結果而異。

另請參閱

FGRun

FGGet範例

VB 例:

```
Dim val As Integer
m_spel.MotorsOn = True
m_spel.FGRun("Sequence1")
m spel.FGGet("Sequence1","", SpelForceProps.EndStatus, val)
```

```
int errCode;
m_spel.MotorsOn = true;
```

```
m_spel.FGRun("Sequence1");
m_spel.FGGet("Sequence1","", SpelForceProps.EndStatus, val);
```

15.3.56 FGRun 方法, Spel類別

描述

執行力覺引導序列。

語法

Sub FGRun (Sequence As String)

參數

 Sequence 力覺引導序列的名稱,或者代表力覺引導序列名稱的字串變數

備註

執行指定的力覺引導序列。力覺引導序列從執行了FGRun語句的位置開始。在執行之前,先用移動命令Go語句或 Move語句移動到預期的起點位置。

FGRun會在指定的力覺引導序列結束之後,進入下一個語句。 使用SGGet取得FGRun中執行的力覺引導序列的結果。 當在CP參數或CP語句中啟用了路徑運動,請等待其停止,然後執行力覺引導序列。

開始執行時,滿足以下條件之一,則錯誤。 程式中指定的機器人和RobotNumber屬性中指定的機器人相異 在Robot語句中指定正確的機器人。 程式中指定的機器人類型和RobotType屬性中指定的機器人類型相異 在Robot語句中指定正確的機器人。 程式中指定的工具編號和RobotTool屬性中指定的工具編號相異 在Tool語句中指定正確的工具編號。 馬達關閉狀態 在Motor語句中打開馬達。 力覺控制機能執行中 在FCEnd語句中停止力覺控制。 輸送帶追蹤運行中,請以Cnv_AbortTrack語句停止輸送帶追蹤。 轉矩控制模式啟用中 在TC語句中禁用轉矩控制模式。

執行FGRun會自動重寫以下屬性,請不要和以下屬性一起使用。 FM物件 AvgForceClear屬性 PeakForceClear屬性

程式執行中, 無法執法執行本方法。

另請參閱

FGGet

FGRun範例

VB 例:

```
Dim errCode As Integer
m spel.MotorsOn = True
```

```
m_spel.FGRun("Sequence1")
errCode = m_spel.FGGet("Sequence1", SpelForceProps.EndStatus, val)
```

```
int errCode;
m_spel.MotorsOn = true;
m_spel.FGRun("Sequence1");
errCode = m_spel.FGGet("Sequence1", SpelForceProps.EndStatus, val);
```

15.3.57 Find方法, Spel類別

描述

設定要在動作指令期間儲存座標的條件。

語法

Sub Find (Condition As String)

參數

 Condition 指定輸入狀態作為觸發。

另請參閱

Go, Jump

Find範例

VB 例:

m_spel.Find("Sw(5) = On")

C# 例:

m_spel.Find("Sw(5) = On");

15.3.58 Fine方法, Spel類別

描述

指定並顯示目標點的定位準確度。

語法

Sub Fine (J1MaxErr As Integer, J2MaxErr As Integer, J3MaxErr As Integer, J4MaxErr As Integer , J5MaxErr As Integer , J6MaxErr As Integer [, J7MaxErr As Integer][, J8MaxErr As Integer] [, J9MaxErr As Integer])

參數

 J1MaxErr - J9MaxErr 介於0-32767的整數,代表各點允許的定位誤差。關節7、8及9的值為選用。

另請參閱

Weight

Fine範例

VB 例:

m_spel.Fine(1000, 1000, 1000, 1000, 0, 0)

C# 例:

m_spel.Fine(1000, 1000, 1000, 1000, 0, 0);

15.3.59 GetAccel方法, Spel類別

描述

傳回指定的加速/減速值。

語法

Function GetAccel (ParamNumber As Integer) As Integer

參數

ParamNumber

可包含下列數值的整數運算式:1:加速設定值;2:減速設定值;3:在Jump中使用的起始加速設定值;4:在 Jump中使用的起始減速設定值;5:在Jump中使用的接近加速設定值;6:在Jump中使用的接近減速設定值。

傳回值

包含指定加速/減速值的整數。

另請參閱

Accel

GetAccel範例

VB 例:

Dim x As Integer
x = m_spel.GetAccel(1)

C# 例:

int x; x = m_spel.GetAccel(1);

15.3.60 GetArm方法, Spel類別

描述

傳回目前機器人的目前手臂編號。

語法

Function GetArm () As Integer

傳回值

包含目前手臂編號的整數。

另請參閱

Arm, ArmSet, Robot, Tool

GetArm範例

VB 例:

saveArm = m_spel.GetArm()
m_spel.Arm(2)

C# 例:

m_spel.Arm(2);

15.3.61 GetConnectionInfo方法, Spel類別

描述

傳回有關控制器連線的資訊。

語法

Function GetConnectionInfo() As SpelConnectionInfo()

傳回值

SpelConnectionInfo的陣列。

另請參閱

GetControllerInfo

傳回值

GetConnectionInfo會傳回SpelConnectionInfo的陣列。連線資訊是在Epson RC+的[Setup]-[PC to Controller Communication]對話方塊中設定。

GetConnectionInfo範例

VB 例:

Dim info() As SpelConnectionInfo

info = m_spel.GetConnectionInfo()

C# 例:

SpelConnectionInfo[] info = m_spel.GetConnectionInfo();

15.3.62 GetControllerInfo方法, Spel類別

描述

傳回有關目前控制器的資訊。

語法

Function GetControllerInfo() As SpelControllerInfo

傳回值

SpelControllerInfo執行個體。

另請參閱

GetErrorMessage, GetRobotInfo, GetTaskInfo

備註

GetControllerInfo會傳回SpelControllerInfo類別的新執行個體,其中包含控制器資訊屬性。

GetControllerInfo範例

VB 例:

```
SpelControllerInfo info;
string msg;
info = m_spel.GetControllerInfo();
msg = "Project Name:" + info.ProjectName + "\r\n ProjectID :" +
"info.ProjectID";
MessageBox.Show(msg);
```

15.3.63 GetCurrentConnectionInfo 方法, Spel類別

描述

傳回目前控制器的資訊。

語法

Function GetCurrentConnectionInfo () As SpelConnectionInfo

傳回值

SpelConnectionInfo

另請參閱

GetConnectionInfo, GetControllerInfo

GetCurrentConnectionInfo範例 VB例:

Dim info As SpelConnectionInfo

```
info = m_spel.GetCurrentConnectionInfo()
```

C# 例:

SpelConnectionInfo info;

info = m_spel.GetCurrentConnectionInfo();

15.3.64 GetCurrentUser方法, Spel類別

描述

傳回目前Epson RC+ 8.0使用者。

語法

Function GetCurrentUser () As String

傳回值

包含目前使用者的字串變數。

另請參閱

Login

GetCurrentUser範例

VB 例:

Dim currentUser As String

```
currentUser = m_spel.GetCurrentUser()
```

C# 例:

string currentUser;

```
currentUser = m_spel.GetCurrentUser();
```

15.3.65 GetECP方法, Spel類別

描述

傳回目前機器人的目前ECP編號。

語法

Function GetECP () As Integer

傳回值

包含目前ECP數量的整數。

另請參閱

ECP, ECPSet

GetECP範例

VB 例:

saveECP = m_spel.GetECP()
m_spel.ECP(2)

C# 例:

saveECP = m_spel.GetECP();
m_spel.ECP(2);

15.3.66 GetErrorMessage方法, Spel類別

描述

傳回指定錯誤或警告代碼的錯誤訊息。

語法

Function GetErrorMessage (ErrorCode As Integer) As String

參數

 ErrorCode 傳回相關錯誤訊息的錯誤代碼。

傳回值

包含錯誤訊息的字串。

另請參閱

ErrorCode

GetErrorMessage範例

VB 例:

```
Dim msg As String
```

```
If m_spel.ErrorOn Then
   msg = m_spel.GetErrorMessage(m_spel.ErrorCode)
   MsgBox(msg)
End If
```

```
string msg;
if (m_spel.ErrorOn) {
    msg = m_spel.GetErrorMessage(m_spel.ErrorCode);
    MessageBox.Show(msg);
}
```

15.3.67 GetIODef方法, Spel類別

描述

取得輸入、輸出或記憶體I/O位元、位元組或字元的定義資訊。

語法

Sub GetIODef(Type As SpelIOLabelTypes, Index As Integer, ByRef Label as String, ByRef Description As String)

參數

- Type 指定I/O類型,如下所示: InputBit = 1 InputByte = 2 InputWord = 3 OutputBit = 4 OutputByte = 5 OutputWord = 6 MemoryBit = 7 MemoryByte = 8 MemoryWord = 9 InputReal = 10 OutputReal = 11
- Index 指定位元或埠號。
- Label 傳回標籤。
- Description 傳回描述。

傳回值

數值會在Label和Description參數中傳回。

備註

使用GetIODef可取得目前專案中所有I/O所使用的標籤與描述。

另請參閱

SetIODef

GetIODef範例

VB 例:

```
Dim label As String
Dim desc As String
m spel.GetIODef(SpelIOLabelTypes.InputBit, 0, label, desc)
```

C# 例:

string label, desc;

m_spel.GetIODef(SpelIOLabelTypes.InputBit, 0, out label, out desc);

15.3.68 GetJRange 方法, Spel類別

描述

針對指定關節, 傳回允許動作區域的脈衝值(範圍設定值)。

語法

Function GetJRange (JointNumber As Integer, Bound As Integer) As Integer

參數

JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)

Bound
 用證書指定以下2個值之一。1:指定下限脈衝值
 2:指定上限脈衝值

傳回值

傳回指定關節的範圍設定值(整數值、單位:脈衝)。

另請參閱

JRange

GetJRange範例

VB 例:

```
Dim val1 As Integer
Dim val2 As Integer
val1 = m_spel.GetJRange(1, 1)
val2 = m_spel.GetJRange(1, 2)
```

```
int minRange, maxRange;
minRange = m_spel.GetJRange(1, 1);
maxRange = m_spel.GetJRange(1, 2);
```

15.3.69 GetLimitTorque方法, Spel類別

描述

針對目前機器人的指定關節, 傳回限制力矩。

語法

Function GetLimitTorque (JointNumber As Integer) As Integer

參數

 JointNumber 代表管道#的整數值。

傳回值

介於1至9的整數值,代表指定關節的限制力矩設定。

另請參閱

GetRealTorque, GetRobotPos, LimitTorque

GetLimitTorque範例

VB 例:

```
Dim jlLimitTorque As Integer
jlLimitTorque = m_spel.GetLimitTorque(1)
```

```
int j1LimitTorque;
j1LimitTorque = m_spel.GetLimitTorque(1);
```

15.3.70 GetLimZ方法 · Spel類別

描述

傳回目前LimZ設定。

語法

Function GetLimZ () As Single

傳回值

包含LimZ值的實際值。

另請參閱

LimZ, Jump

GetLimZ範例

VB 例:

saveLimZ = m_spel.GetLimZ()
m_spel.LimZ(-22)

C# 例:

saveLimZ = m_spel.GetLimZ();
m_spel.LimZ(-22);

描述

擷取某一機器人點的座標資料。

語法

Function GetPoint (PointNumber As Integer) As SpelPoint Function GetPoint (PointName As String) As SpelPoint

參數

- PointNumber
 目前機器人的控制器點記憶體中某個點的整數運算式。
- PointName
 包括修飾字元的字串。可使用點標籤Pxxx、P及「」。

傳回值

SpelPoint值

另請參閱

SetPoint

GetPoint範例

VB 例:

```
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 25.0
m_spel.Go(pt)
```

```
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 25.0;
m_spel.Go(pt);
```

15.3.72 GetRealTorque方法, Spel類別

描述

針對目前機器人的指定關節, 傳回力矩。

語法

Function GetRealTorque (JointNumber As Integerer) As Double

參數

 JointNumber 所需之關節的整數運算式。

傳回值

介於0至1的雙精度值,代表目前功率模式及指定關節的最大力矩部分。

另請參閱

GetLimitTorque, GetRobotPos

GetRealTorque範例

VB 例:

Dim j1Torque As Integer
j1Torque = m_spel.GetRealTorque(1)

```
double j1Torque;
j1Torque = m_spel.GetRealTorque(1);
```

15.3.73 GetRobotInfo 方法, Spel類別

描述

傳回機器人的資訊。

語法

Function GetRobotInfo (RobotNumber As Integer) As SpelRobotInfo

參數

 RobotNumber 指定機器人的編號

傳回值

SpelRobotInfo

另請參閱

GetControllerInfo, GetTaskInfo

GetRobotInfo範例

VB 例:

```
SpelRobotInfo info;
string msg;
info = m_spel.GetRobotInfo(1);
msg = "Robot Model: " + info.RobotModel +
        "\r\n Robot Serial: " + info.RobotSerial;
MessageBox.Show(msg);
```

15.3.74 GetRobotPos方法 · Spel類別

描述

傳回目前機器人位置。

語法

Function GetRobotPos(PosType As SpelRobotPosType, Arm As Integer, Tool As Integer, Local As Integer) As Single()

參數

- PosType 指定要傳回之位置資料的類型。
- Arm 指定機器人手臂的整數運算式。
- Tool 指定機器人工具的整數運算式。
- Local 指定機器人本地的整數運算式。

傳回值

包含9個元素的單一資料類型陣列。傳回的資料係依指定的PosType而定。 World X, Y, Z, U, V, W, R, S, T Joint J1, J2, J3, J4, J5, J6, J7, J8, J9 Pulse Pls1, Pls2, Pls3, Pls4, Pls5, Pls6, Pls7, Pls8, Pls9

另請參閱

GetPoint

GetRobotPos範例

VB 例:

```
Dim values() As Single
values = m_spel.GetRobotPos(SpelRobotPosType.World, 0, 0, 0)
```

```
float[] values;
values = m_spel.GetRobotPos(SpelRobotPosType.World, 0, 0, 0);
```

15.3.75 GetSpeed方法 · Spel類別

描述

傳回目前機器人的三個速度設定的其中之一。

語法

Function GetSpeed (ParamNumber As Integer) As Integer

參數

 ParamNumber 求得下列任一值的整數運算式:1:PTP動作速度;2:Jump起始速度;3:Jump接近速度。

傳回值

介於1-100的整數運算式。

另請參閱

Speed

GetSpeed範例

VB 例:

Dim ptpSpeed As Integer
ptpSpeed = m_spel.GetSpeed(1)

C# 例:

ptp
Speed = m_spel.GetSpeed(1);

15.3.76 GetTaskInfo 方法, Spel類別

描述

傳回任務資訊。

語法

Function GetTaskInfo (TaskName As String) As SpelTaskInfo Function GetTaskInfo (TaskNumber As Integer) As SpelTaskInfo

參數

- TaskName 指定任務名稱
- TaskNumber 指定任務#

傳回值

SpelTaskInfo

另請參閱

GetControllerInfo, GetRobotInfo

GetTaskInfo範例

VB 例:

```
SpelTaskInfo info;
string msg;
info = m_spel.GetTaskInfo(1);
msg= "Task Name: " + info.TaskName +
        "\r\n Task State: " + info.State;
MessageBox.Show(msg);
```

15.3.77 GetTool方法 · Spel類別

描述

傳回目前機器人的目前工具編號。

語法

Function GetTool () As Integer

傳回值

包含目前工具編號的整數。

另請參閱

Arm, TLSet, Tool

GetTool範例

VB 例:

saveTool = m_spel.GetTool()
m_spel.Tool(2)

C# 例:

saveTool = m_spel.GetTool();
m_spel.Tool(2);

15.3.78 GetVar方法, Spel類別

描述

傳回控制器中SPEL+全域保留變數的值。

語法

Function GetVar(VarName As String) As Object

參數

VarName
 SPEL+全域保留變數的名稱。對於陣列,可以傳回整個陣列或只傳回一個元素。

傳回值

您可使用GetVar在控制器的目前專案中搜尋任何全域保留變數的數值。必須先建置專案,再搜尋數值。 若要擷取整個陣列,請在VarName中提供陣列名稱。若要擷取一個陣列的某個元素,請在VarName中提供子腳本。

備註

您可使用GetVar在控制器的目前專案中擷取任何全域保留變數的數值。在擷取數值之前,必須成功建置專案。

另請參閱

SetVar

GetVar範例

在SPEL+專案中會宣告變數:

```
Global Preserve Integer g_myIntVar
Global Preserve Real g_myRealArray(10)
Global Preserve String g_myStringVar$
Function main
...
Fend
```

在Visual Basic專案中: 由於已將g_myIntVar宣告為整數,請將用來取得g_myInVar值的Visual Basic變數宣告為整數。若為g_myRealArray, 請宣告為Single變數的陣列。

```
Dim myIntVar As Integer
Dim myRealArray() As Single
Dim myStringVar As String
myIntVar = m_spel.GetVar("g_myIntVar")
myRealArray = m_spel.GetVar("g_myRealArray")
myStringVar = m_spel.GetVar("g_myStringVar$")
```

C #專案中的範例: 由於已將g_myIntVar宣告為整數,請將用來取得g_myInVar值的C#變數宣告為整數。若為g_myRealArray,請宣告為 Float變數的陣列。

```
int myIntVar;
float[] myRealArray;
string myStringVar;
myIntVar = m_spel.GetVar("g_myIntVar");
myRealArray = m_spel.GetVar("g_myRealArray");
myStringVar = m_spel.GetVar("g_myStringVar$");
```

15.3.79 Go方法 · Spel類別

描述

以PTP的形式將手臂從目前位置移至指定點或XY位置。GO指令可同時移動任何組合的機器人軸。

語法

Sub Go (PointNumber As Integer) Sub Go (Point As SpelPoint) Sub Go (Point As SpelPoint, AttribExpr As String) Sub Go (PointExpr As String)

參數

每個語法都有一個參數,用以指定在Go動作期間手臂移動的結束點。此為PTP動作結束時的最終位置。

- PointNumber 從目前機器人的點記憶體(被保存在控制器中)中使用先前示教點的編號,以指定終點。
 Point
- 透過使用SpelPoint資料類型來指定目標點。
- AttribExpr 透過使用字串運算式來指定結束點屬性。
 可包括CP、LJM、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。
- PointExpr 透過使用字串運算式來指定終點。
 可包括CP、LJM、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。

另請參閱

Accel, Speed Arc, Arc3, CVMove, Jump, Jump3, Jump3CP, Move BGo, BMove, TGo, TMove Arch, Find, Sense, Till

Go範例

VB 例:

```
使用點編號指定的點
m_spel.Tool(1)
m_spel.Go(100)

使用SpelPoint指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.Go(pt)

使用點運算式指定的點
m_spel.Go("P0 /L /2")
m_spel.Go("P1 :Z(-20)")

使用平行處理
m_spel.Go("P1 \!D50; On 1; D90; Off 1\!")

使用標籤指定的點
m_spel.Go("pick")
```
// 使用點編號指定的點 m_spel.Tool(1); m_spel.Go(100);

// 使用SpelPoint指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.Go(pt);

// 使用點運算式指定的點 m_spel.Go("P0 /L /2"); m_spel.Go("P1 :Z(-20)");

// 使用平行處理 m_spel.Go("P1 \!D50; On 1; D90; Off 1\!");

// 使用標籤指定的點
m_spel.Go("pick");

15.3.80 Halt方法, Spel類別

描述

暫停執行指定的任務。

語法

Sub Halt (TaskNumber As Integer) Sub Halt (TaskName As String)

參數

- TaskNumber 要暫停之任務的任務#。任務#的範圍介於1至32。
- TaskName 要暫停之包含任務名稱的字串運算式。

⚠ 注意

以Call方法執行的函數·無法透過Halt方法暫停任務。 如欲暫停·請使用Xqt方法。

另請參閱

Resume, Xqt

Halt範例

VB 例:

m_spel.Halt(3)

C# 例:

m_spel.Halt(3);

15.3.81 Here方法 · Spel類別

描述

在目前位置示教一個點。

語法

Sub Here (PointNumber As Integer) Sub Here (PointName As String)

參數

- PointNumber 整數運算式,代表目前機器人的點記憶體中的點。只要從0開始,即可使用任何有效的點編號。
- PointName 點標籤的字串運算式。

另請參閱

SetPoint

Here範例

VB 例:

m_spel.Here("P20")

C# 例:

m_spel.Here("P20");

15.3.82 HideWindow方法, Spel類別

描述

隱藏先前與ShowWindow一起顯示的Epson RC+ 8.0視窗。

語法

Sub HideWindow (WindowID As SpelWindows)

參數

 視窗ID 要顯示的Epson RC+ 8.0視窗的ID。

另請參閱

RunDialog, ShowWindow

HideWindow範例

VB 例:

```
Sub btnHideIOMonitor_Click ______
ByVal sender As System.Object, _____
ByVal e As System.EventArgs) ______
Handles btnHideIOMonitor.Click
```

```
\mbox{m\_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor)} End Sub
```

```
void btnHideIOMonitor_Click(object sender, EventArgs e)
{
    m_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor);
}
```

15.3.83 Home方法 · Spel類別

描述

將機器人手臂移至使用HomeSet方法設定的使用者定義起始點位置。

語法

Sub Home ()

另請參閱

HomeSet, MCal

Home範例

VB 例:

With m_spel .MotorsOn = True .Home() End With

C# 例:

m_spel.MotorsOn = true; m_spel.Home();

15.3.84 Hofs 方法, Spel類別

描述

讀取或設置從每個關節的編碼器原點到軟體原點的校正脈衝值。

語法

Function Hofs (JointNumber As Integer) As Integer

Sub Hofs (J1Pulses As Integer, J2Pulses As Integer, J3Pulses As Integer, J4Pulses As Integer, [J5Pulses As Integer], [J6Pulses As Integer], [J7Pulses As Integer], [J8Pulses As Integer], [J9Pulses As Integer])

參數

 J1Pulses - J9Pulses 代表第1~9關節的脈衝值的整數 J5Pulses - J9Pulses可以省略

傳回值

傳回指定關節的校正脈衝值(整數值、單位:脈衝)。

備註

本方法可以顯示或設定原點校正脈衝值。本方法可以設定編碼器原點(Z相)到機械原點的偏位值。

機器人的動作控制基於安裝在每個關節上的編碼器原點,但是編碼器原點不一定與機器人機械原點匹配。因此,可以 使用本方法設定校正脈衝值,使編碼器遠點和軟體原點匹配。

<u> 注</u>意

除非必須,否則切勿更改Hofs值

Hofs值在出廠時已經精確設定。不必要的更改此值可能會導致定位錯誤或機器人意外行爲。除非必須.否則切勿更改Hofs值。

自動計算Hofs值

要自動計算Hofs值,請將機械臂移動到要校準的位置,然後執行Calib。這樣控制器就會根據CalPls脈衝值和校準 位置脈衝值,自動計算Hofs值。

保存和復原Hofs

可以在功能表中選擇[System Configuration]-[Robot]-[Calibration],然後用保存或復原功能,保存或復原Hofs。

另請參閱

CalPls, Home, Hordr, MCal

Hofs範例

VB 例:

```
Dim val As Integer
val = m_spel.Hofs(1)
```

```
int val;
val = m_spel.Hofs(1);
```

15.3.85 HomeSet方法, Spel類別

描述

指定Home方法所使用的位置。

語法

Sub HomeSet (J1Pulses As Integer, J2Pulses As Integer, J3Pulses As Integer, J4Pulses As Integer, J5Pulses As Integer, J6Pulses As Integer [, J7Pulses As Integer] [, J8Pulses As Integer][, J9Pulses As Integer])

參數

J1Pulses - J9Pulses
 每個關節的Home位置編碼器脈衝值。關節7、8及9為選用。

另請參閱

Home, MCal

HomeSet範例

VB 例:

```
    設定目前位置的起始點位置
    With m_spel
    .HomeSet(.Pls(1), .Pls(2), .Pls(3), .Pls(4), 0, 0)
    End With
```

15.3.86 Hordr方法, Spel類別

描述

指定傳回HOME位置的軸順序。

語法

Sub Hordr (Home1 As Integer, Home2 As Integer, Home3 As Integer, Home4 As Integer, Home5 As Integer, Home6 As Integer [, Home7 As Integer] [, Home8 As Integer] [, Home9 As Integer])

參數

• Home 1 - 9

通知哪個軸應在起始點程序的各步驟期間復歸原點的位元模式。0至所有軸之間的任何軸數量都可在第一個步驟期間復歸原點。在指定R、S或T時可以指定Home 7 – 9。

另請參閱

Home, HomeSet, Mcordr

Hordr範例

VB 例:

m_spel.Hordr(2, 13, 0, 0, 0, 0)

C# 例:

m_spel.Hordr(2, 13, 0, 0, 0, 0);

15.3.87 Hour方法 · Spel類別

描述

傳回累計系統運作時間(小時)。

語法

Function Hour () As Single

傳回值

表示時間的整數運算式。

Hour範例

VB 例:

Dim hoursRunning As Long hoursRunning = m_spel.Hour()

C# 例:

float hoursRunning; hoursRunning = m_spel.Hour();

15.3.88 ImportPoints方法 · Spel類別

描述

將點文件導入至目前機器人的目前專案。

語法

Sub ImportPoints (SourcePath As String, ProjectFileName As String [, RobotNumber As Integer])

參數

- SourcePath 字串運算式,代表導入目前專案的檔案以及其特定路徑。副檔名為.pts。
- ProjectFileName 字串運算式,代表只要有目前機器人或提供RobotNumber,即被導入到指定機器人專案中的特定檔案。副檔名 為.pts。
- RobotNumber 要使用點文件之機器人的整數運算式。指定0可讓其變為共同點文件。

另請參閱

SavePoints

ImportPoints範例

VB 例:

```
With m_spel
    .ImportPoints("c:\mypoints\model1.pts", "robot1.pts")
End With
```

C# 例:

m_spel.ImportPoints(@"c:\mypoints\model1.pts", "robot1.pts");

15.3.89 In方法 · Spel類別

描述

傳回指定輸入埠的狀態。每個連接埠包含8個輸入位元(一位元組)。

語法

Function In (PortNumber As Integer) As Integer Function In (Label As String) As Integer

參數

- PortNumber
 代表輸入埠的整數。每個埠由8位元(1位元組)組成。
- Label
 包含輸入位元組標籤的字串運算式。

傳回值

介於0至255的整數,代表輸入埠的狀態。

另請參閱 InBCD, Out, OpBCD, Sw

In範例

VB 例:

Dim port1Value As Integer
port1Value = m_spel.In(1)

```
int port1Value;
port1Value = m_spel.In(1);
```

15.3.90 InBCD方法, Spel類別

描述

傳回8個輸入的輸入狀態(使用BCD格式)。(二進位十進碼)

語法

Function InBCD (PortNumber As Integer) As Integer Function InBCD (Label As String) As Integer

參數

- PortNumber 代表輸入埠的整數
- Label 代表輸入位元組標籤的字串運算式。

傳回值

介於0至9的整數,代表輸入埠的狀態。 **另請參閱** In, Out, OpBCD, Sw

InBCD範例

VB 例:

Dim port1Value As Integer
port1Value = m_spel.InBCD(1)

```
int port1Value;
port1Value = m_spel.InBCD(1);
```

描述

指定目前機器人的裝載慣性和離心率。

語法

Sub Inertia (LoadInertia As Single, Eccentricity As Single)

參數

- LoadInertia
 雙精度值,用以指定夾具末端關節(包含夾具末端和工件)中心周圍的總慣性力矩(kgm2)。
- Eccentricity 雙精度值,用以指定夾具末端關節(包含夾具末端和工件)中心周圍的離心率(公釐)。

另請參閱

Weight

Inertia範例

VB 例:

m_spel.Inertia(0.02, 1.0)

C# 例:

m_spel.Inertia(0.02, 1.0);

15.3.92 Initialize方法, Spel類別

描述

初始化Spel類別執行個體。

語法

Sub Initialize ()

備註

一般而言,Spel類別執行個體會在第一個方法執行時自動初始化。當Epson RC+8.0載入至記憶體時,初始化可能需要幾秒鐘的時間。在某些情況下,您可能會想在啟動期間先於應用程式中調用initialize。 根據ServerInstance,將RC+作為伺服器進程啟動。每個ServerInstance對應於一個控制器和一個專案。如果使用ServerInstance屬性,則必須在執行初始化之前設置它。

另請參閱

Connect, Disconnect, ServerInstance

Initialize範例

VB 例:

m_spel.Initialize()

C# 例:

m_spel.Initialize();

15.3.93 InReal 方法 · Spel類別

描述

將2個字元(32位元)的輸入資料, 讀取為32位元浮點資料(符合IEEE754標準)。

語法

Function InReal (PortNumber As Integer) As Single

參數

 PortNumber 代表輸入埠的整數

傳回值

返回輸入埠的狀態,在32位元浮點資料(符合IEEE754標準)中返回。

另請參閱

In, InBCD, InW, Out, OutW, OutReal

InReal範例

VB 例:

Dim val As Single
val = m_spel.InReal(32)

C# 例:

float val; val = m_spel.InReal(32);

15.3.94 InsideBox方法, Spel類別

描述

傳回接近檢查區域的檢查狀態。

語法

Function InsideBox (BoxNumber As Integer) As Boolean

參數

 BoxNumber 介於1至15的整數運算式,代表要傳回狀態的接近檢查區域。

傳回值

若機器人夾具末端位於指定工作空間內,會傳回True,否則會傳回False。

另請參閱

Box, InsidePlane

InsideBox範例

VB 例:

Dim isInside As Boolean
isInside = m_spel.InsideBox(1)

```
bool isInside;
isInside = m_spel.InsideBox(1);
```

15.3.95 InsidePlane方法 · Spel類別

描述

傳回接近檢查平面的檢查狀態。

語法

Function InsidePlane (PlaneNumber As Integer) As Boolean

參數

 PlaneNumber 介於1至15的整數運算式,代表要傳回狀態的接近檢查平面。

傳回值

若機器人夾具末端位於指定工作空間內,會傳回True,否則會傳回False。

另請參閱

InsideBox, Plane

InsidePlane範例

VB 例:

Dim isInside As Boolean
isInside = m_spel.InsidePlane(1)

```
bool isInside;
isInside = m_spel.InsidePlane(1);
```

15.3.96 InW方法 · Spel類別

描述

傳回指定輸入字元埠的狀態。每個字元埠包含16個輸入位元。

語法

Function InW (PortNumber As Integer) As Integer Function InW (Label As String) As Integer

參數

- PortNumber 代表輸入埠的整數
- Label
 包含輸入字元標籤的字串運算式。

傳回值

介於0至65535的整數值, 代表輸入埠

另請參閱

In, InBCD, Out, OpBCD, Sw

InW範例

VB 例:

Dim data As Integer
data = m_spel.InW(0)

```
int data;
data = m_spel.InW(0);
```

15.3.97 IsOptionActive方法 · Spel類別

描述

傳回軟體選配件的狀態。

語法

Function IsOptionActive (option As SpelOptions) As Boolean

參數

 option 代表選配件編號的整數。

傳回值

False - 無效 True - 有效

另請參閱 GetControllerInfo

IsOptionActive範例

VB 例:

Dim ret As Boolean
ret = m_spel.IsOptionActive(SpelOptions.FieldbusMaster)

C# 例:

bool ret; ret = m_spel.IsOptionActive(SpelOptions.FieldbusMaster);

15.3.98 JRange方法, Spel類別

描述

定義指定機器人關節的允許工作範圍(以脈衝為單位)。

語法

Sub JRange (JointNumber As Integer, LowerLimitPulses As Integer, UpperLimitPulses As Integer)

參數

- jointNumber 介於1-9的整數,代表要指定之JRange的關節。
- LowerLimitPulses
 代表指定關節下限範圍之編碼器脈衝計數位置的整數。
- UpperLimitPulses
 代表指定關節上限範圍之編碼器脈衝計數位置的整數。

另請參閱

XYLim

JRange範例

VB 例:

m_spel.JRange(1, -30000, 30000)

```
m_spel.JRange(1, -30000, 30000);
```

15.3.99 JS方法 · Spel類別

描述

Jump Sense會偵測手臂是否在完成JUMP指令之前停止(使用SENSE輸入),或手臂是否完成JUMP移動。

語法

Function JS () As Boolean

傳回值

若動作期間偵測到SENSE輸入,會傳回True,否則會傳回False。

另請參閱

Jump, Jump3, Jump3CP, Sense, Till

JS範例

VB 例:

```
With m_spel
    .Sense("Sw(1) = On")
    .Jump("P1 Sense")
    stoppedOnSense = .JS()
End With
```

```
m_spel.Sense("Sw(1) = On");
m_spel.Jump("P1 Sense");
stoppedOnSense = m_spel.JS();
```

15.3.100 JTran方法, Spel類別

描述

執行相對關節移動。

語法

Sub JTran (JointNumber As Integer, Distance As Single)

參數

- JointNumber 要移動的特定關節。
- Distance
 要移動的距離。旋轉關節的單位為度,線性關節的單位為公釐。

另請參閱

PTran, Pulse

JTran範例

VB 例:

・以正方向將關節1移動45度。m_spel.JTran(1, 45.0)

C# 例:

//以正方向將關節1移動45度。 m_spel.JTran(1, 45.0);

15.3.101 Jump方法 · Spel類別

描述

利用PTP動作將手臂從目前位置移至指定點(先垂直向上移動,接著水平移動,最後向下垂直移至最終目標點)。

語法

Sub Jump (PointNumber As Integer) Sub Jump (Point As SpelPoint) Sub Jump (Point As SpelPoint, AttribExpr As String) Sub Jump (PointExpr As String)

參數

每個語法都有一個參數,用以指定在Jump動作期間手臂移動的結束點。此為PTP動作結束時的最終位置。 PointNumber 透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標點。

- Point 透過使用SpelPoint資料類型來指定目標點。
- AttribExpr 透過使用字串運算式來指定結束點屬性。
 可包括C拱形編號、搜尋運算式(LimZ、CP、SYNC、Sense、Till或Find)和並行處理陳述式。
- PointExpr 透過使用字串運算式來指定終點。
 可包括C拱形編號、搜尋運算式(LimZ、CP、SYNC、Sense、Till或Find)和並行處理陳述式。

另請參閱

Accel, Speed, Arc, Arc3, CVMove, Go, Jump3, Jump3CP, Move BGo, BMove, TGo, TMove Arch, Find, Sense, Till

Jump範例

VB 例:

```
    使用點編號指定的點

m spel.Tool(1)
m spel.Jump(100)
' 使用SpelPoint指定的點
Dim pt As SpelPoint
pt = m spel.GetPoint("P*")
pt.X = 125.5
m spel.Jump(pt)

    使用點運算式指定的點

m spel.Jump("P0 /L /2")
m spel.Jump("P1 :Z(-20)")
m spel.Jump("P1 C0")
m spel.Jump("P1 C0 LimZ -10")
m spel.Jump("P1 C0 Sense Sw(0)=On")

    使用平行處理

m spel.Jump("P1 \!D50; On 1; D90; Off 1\!")
• 使用標籤指定的點
m_spel.Jump("pick")
```

C# 例:

// 使用點編號指定的點 m spel.Tool(1); m spel.Jump(100); // 使用SpelPoint指定的點 SpelPoint pt; pt = m spel.GetPoint("P0"); pt.X = 125.5; m spel.Jump(pt); // 使用點運算式指定的點 m_spel.Jump("P0 /L /2"); m_spel.Jump("P1 :Z(-20)"); m_spel.Jump("P1 C0"); m_spel.Jump("P1 C0 LimZ -10"); m_spel.Jump("P1 C0 Sense Sw(0)=On"); // 使用平行處理 m_spel.Jump("P1 \!D50; On 1; D90; Off 1\!"); // 使用標籤指定的點

m_spel.Jump("pick");

15.3.102 Jump3方法, Spel類別

描述

使用兩個CP動作與一個PTP動作之組合的3D閘道動作。機器人會移到起始點、接近點和目標點。

語法

Sub Jump3 (DepartPoint As Integer, ApproPoint As Integer, DestPoint As Integer) Sub Jump3 (DepartPoint As SpelPoint, ApproPoint As SpelPoint, DestPoint As SpelPoint) Sub Jump3 (DepartPoint As String, ApproPoint As String, DestPoint As String)

參數

- DepartPoint
 目前位置之上的起始點(使用點編號或字串點運算式)。
- ApproPoint
 目標位置之上的結束點(使用點編號或字串點運算式)。
- DestPoint 動作的目標位置(使用點編號或字串點運算式)。
 可包括C拱形編號、搜尋運算式(CP、LJM、SYNC、Sense、Till或Find)和並行處理陳述式。

另請參閱

Accel, AccelR, AccelS, Speed, SpeedR, SpeedS Arc, Arc3, CVMove, Go, Jump, Jump3CP, Move BGo, BMove, TGo, TMove Arch, Find, Sense, Till

Jump3範例

VB 例:

```
    使用點編號指定的點

m spel.Tool(1)
m spel.Jump3(1, 2, 3)
' 使用SpelPoint指定的點
Dim pd As SpelPoint
Dim pa As SpelPoint
Dim pt As SpelPoint
pd = m spel.GetPoint("P*")
pd.Z = 125.5
pa = m spel.GetPoint("P2")
pa.Z = 125.5
pt = m spel.GetPoint("P2")
m spel.Jump3(pd, pa, pt)

    使用點運算式指定的點

m_spel.Jump3("P1", "P2", "P3 C0")
m spel.Jump3("P1", "P2", "P3 C0 Sense Sw(0)=On")
m spel.Jump3("P0 -TLZ(10), P1 -TLZ(10), P1")

    使用平行處理

m spel.Jump3("P1", "P2", "P3 \!D50; On 1; D90; Off 1\!")
• 使用標籤指定的點
m spel.Jump3("depart", "approach", "place")
```

```
// 使用點編號指定的點
m spel.Tool(1);
m spel.Jump3(1, 2, 3);
// 使用SpelPoint指定的點
SpelPoint pd, pa, pt;
pd = m spel.GetPoint("P1");
pd.Z = 125.5;
pa = m_spel.GetPoint("P2");
pa.Z = 125.5;
pt = m_spel.GetPoint("P2");
m_spel.Jump3(pd, pa, pt);
// 使用點運算式指定的點
m_spel.Jump3("P1", "P2", "P3 C0");
m_spel.Jump3("P1", "P2", "P3 C0 Sense Sw(0)=On");
m spel.Jump3("P0 -TLZ(10), P1 -TLZ(10), P1");
// 使用平行處理
m spel.Jump3("P1", "P2", "P3 \!D50; On 1; D90; Off 1\!");
```

// 使用標籤指定的點
m_spel.Jump3("depart", "approach", "place");

Rev.2

15.3.103 Jump3CP方法, Spel類別

描述

使用三個CP動作之組合的3D閘道動作。

語法

Sub Jump3CP (DepartPoint As Integer, ApproPoint As Integer, DestPoint As Integer) Sub Jump3CP(DepartPoint As SpelPoint, ApproPoint As SpelPoint, DestPoint As SpelPoint) Sub Jump3CP(DepartPoint As String, ApproPoint As String, DestPoint As String)

參數

- DepartPoint
 目前位置之上的起始點(使用點編號或字串點運算式)。
- ApproPoint
 目標位置之上的結束點(使用點編號或字串點運算式)。
- DestPoint 動作的目標位置(使用點編號或字串點運算式)。
 可包括ROT、C拱形編號、搜尋運算式(CP、LJM、SYNC、Sense、Till或Find)和並行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS Arc, Arc3, CVMove, Go, Jump, Jump3, Move BGo, BMove, TGo, TMove Arch, Find, Sense, Till

Jump3CP範例

VB 例:

```
    使用點編號指定的點

m spel.Tool(1)
m spel.Jump3CP(1, 2, 3)
' 使用SpelPoint指定的點
Dim pd As SpelPoint
Dim pa As SpelPoint
Dim pt As SpelPoint
pd = m spel.GetPoint("P*")
pd.Z = 125.5
pa = m spel.GetPoint("P2")
pa.Z = 125.5
pt = m spel.GetPoint("P2")
m spel.Jump3CP(pd, pa, pt)

    使用點運算式指定的點

m spel.Jump3CP("P1",
                     "P2", "P3 C0")
m spel.Jump3CP("P1", "P2", "P3 C0 Sense Sw(0)=On")
m spel.Jump3CP("P1", "P2", "P3 \!D50; On 1; D90; Off 1\!")

    使用平行處理

m spel.Jump3CP("P0 -TLZ(10), P1 -TLZ(10), P1")

    使用標籤指定的點

m_spel.Jump3CP("depart", "approch", "place")
```

// 使用點編號指定的點
m_spel.Tool(1);
m_spel.Jump3CP(1, 2, 3);

```
// 使用spelPoint指定的點
SpelPoint pd, pa, pt;
pd = m_spel.GetPoint("P0");
pd.Z = 125.5;
pa = m_spel.GetPoint("P2");
pa.Z = 125.5;
pt = m_spel.GetPoint("P2");
m_spel.Jump3CP(pd, pa, pt);
```

// 使用點運算式指定的點
m_spel.Jump3CP("P1", "P2", "P3 C0");
m_spel.Jump3CP ("P1", "P2", "P3 C0 Sense Sw(0)=On");
m_spel.Jump3CP("P0 -TLZ(10), P1 -TLZ(10), P1");

// 使用平行處理 m_spel.Jump3CP("P1", "P2", "P3 \!D50; On 1; D90; Off 1\!");

// 使用標籤指定的點
m_spel.Jump3CP("depart", "approch", "place");

15.3.104 LimitTorque方法, Spel類別

描述

針對目前機器人,設定高功率模式的上限力矩。

語法

Sub LimitTorque (AllJointsMax As Integer)

Sub LimitTorque (J1Max As Integer, J2Max As Integer, J3Max As Integer, J4Max As Integer, J5Max As Integer, J6Max As Integer)

參數

- AllJointsMax
 在高功率模式中,每個關節之所需力矩上限的整數運算式。
- J1Max J6Max
 在高功率模式中,每個關節之所需力矩上限的整數運算式。

傳回值

介於1至9的整數值,代表指定關節的限制力矩設定。

另請參閱

GetRealTorque, GetRobotPos

LimitTorque範例

VB 例:

```
Dim jlLimitTorque As Integer
jlLimitTorque = m_spel.LimitTorque(1)
```

```
int jlLimitTorquel
jlLimitTorque = m spel.LimitTorque(1);
```

15.3.105 LimZ方法, Spel類別

描述

設定JUMP命令之Z軸高度的預設值。

語法

Sub LimZ (ZLimit As Single)

參數

ZLimit
 Z軸可移動範圍內的座標值。

另請參閱

Jump

LimZ範例

VB 例:

saveLimZ = m_spel.GetLimZ()
m_spel.LimZ(-22)

C# 例:

saveLimZ = m_spel.GetLimZ();
m_spel.LimZ(-22);

15.3.106 LoadPoints方法, Spel類別

描述

將SPEL+點文件載入至目前機器人的控制器點記憶體。

語法

Sub LoadPoints (PointFileName As String [, Merge As Boolean])

參數

- PointFileName
 目前專案中的有效點文件。
- Merge 將目前點整合至指定點檔案時進行設定(可省略)。

另請參閱

ImportPoints, SavePoints

LoadPoints範例

VB 例:

```
With m_spel
    .LoadPoints("part1.pts")
End With
```

```
m_spel.LoadPoints("part1.pts");
```

15.3.107 Local方法, Spel類別

描述

定義本地座標系統。

語法

Sub Local (LocalNumber As Integer, OriginPoint As SpelPoint, [XAxisPoint As SpelPoint], [YAxisPoint As SpelPoint])

Sub Local (LocalNumber As Integer, LocalPoint1 As Integer, BasePoint1 As Integer, LocalPoint2 As Integer, BasePoint2 As Integer)

Sub Local (LocalNumber As Integer, LocalPoint1 As String, BasePoint1 As String, LocalPoint2 As String, BasePoint2 As String)

參數

- LocalNumbe 本地座標系統編號。總共可定義15個本地座標系統(1至15的整數值)。
- OriginPoint 本地座標系統之原點的SpelPoint變數。
- XAxisPoint
 SpelPoint變數,代表本地座標系X軸上的點(可省略)。
- YAxisPoint
 SpelPoint變數,代表本地座標系Y軸上的點(可省略)。
- LocalPoint1, LocalPoint2
 透過整數或字串指定,便是本地座標系統的點數據。
- BasePoint1,BasePoint2
 以代表基座座標系的點數據之整數或字串進行設定。

另請參閱

Base

Local範例

VB 例:

```
Dim originPoint As New SpelPoint
originPoint.X = 100
originPoint.Y = 50
m spel.Local(1, originPoint)
```

```
SpelPoint originPoint = new SpelPoint();
originPoint.X = 100;
originPoint.Y = 50;
m spel.Local(1, originPoint);
```

15.3.108 LocalClr方法, Spel類別

描述

清除針對目前機器人定義的本地(Local)。

語法

Sub LocalClr (LocalNumber As Integer)

參數

 LocalNumber 在15個本地座標系中,要清除(未定義)的座標系編號(整數值1~15)。

另請參閱

Local, LocalDef

LocalClr範例

VB 例:

m_spel.LocalClr(1)

C# 例:

m_spel.LocalClr(1);

15.3.109 LocalDef方法 · Spel類別

描述

傳回本地定義狀態。

語法

Function LocalDef (LocalNumber As Integer) As Boolean

參數

 LocalNumber 要傳回狀態之本地座標的整數運算式(1~15)。

傳回值

若指定本地已定義、會傳回True、否則會傳回False。

另請參閱

Local, LocalClr

LocalDef範例

VB 例:

Dim localExists As Boolean
localExists = m_spel.LocalDef(1)

```
bool localExists;
localExists = m_spel.LocalDef(1);
```

15.3.110 Login方法 · Spel類別

描述

以其他使用者身分登錄至Epson RC+8.0。

語法

Sub Login (LoginID As String, Password As String)

參數

- LoginID
 包含使用者登錄ID的字串運算式。
- Password
 包含使用者密碼的字串運算式。

備註

您可在應用程式中使用Epson RC+ 8.0安全功能。例如,您可顯示一允許不同使用者登錄至系統的功能表。各類使用 者都有專屬的安全權限。如需安全功能的詳細資訊,請參閱Epson RC+ 使用指南。 若已啟用安全功能且未執行LogIn,則Visual Basic應用程式將會以訪客使用者身分登錄。若在Epson RC+ 8.0中啟用 自動登錄,應用程式將會以目前Windows使用者身分自動登錄(若已在Epson RC+ 8.0中設定該使用者)。

另請參閱

GetCurrentUser

Login範例

VB 例:

```
With m_spel
    .Project = "c:\EpsonRC80\projects\myproject\myproject.sprj"
    .LogIn("operator", "oprpass")
End With
```

```
m_spel.Project = @"c:\EpsonRC80\projects\myproject\myproject.sprj";
m_spel.LogIn("operator", "oprpass");
```

15.3.111 MCal方法 · Spel類別

描述

針對搭載增量編碼器的機器人執行機器校準。

語法

Sub MCal ()

另請參閱 MCalComplete, MotorsOn

MCal範例

VB 例:

```
If Not m_spel.MCalComplete() Then
        m_spel.MCal()
End If
```
15.3.112 MCalComplete方法 · Spel類別

描述

若MCal成功完成,會傳回True。

語法

Function MCalComplete () As Boolean

傳回值

若MCal已經完成,會傳回True,否則會傳回False。

另請參閱

MCal

MCalComplete範例

VB 例:

```
If m_spel.MCalComplete() Then
    lblStatus.Text = "MCal Complete"
Else
    lblStatus.Text = "MCal Not Complete"
End If
```

```
if (m_spel.MCalComplete())
    lblStatus.Text = "MCal Complete";
else
    lblStatus.Text = "MCal Not Complete";
```

15.3.113 Mcordr方法, Spel類別

描述

指定機器校準MCal的移動軸順序。

語法

Sub MCordr (Step1 As Integer, Step2 As Integer, Step3 As Integer, Step4 As Integer, Step5 As Integer, Step6As Integer, [Step7 As Integer], [Step8 As Integer], [Step9 As Integer])

參數

• Step 1 - 9

通知哪個軸應在MCal程序的各步驟期間復歸原點的位元模式。0至所有軸之間的軸或任何軸數量可在第一個步驟 期間復歸原點。Step 7 - 9為選用,用於具有超過7個軸的機器人。

另請參閱

Home, HomeSet, Hordr, Mcal

Mcordr範例

VB 例:

m_spel.Mcordr(2, 13, 0, 0, 0, 0)

C# 例:

m_spel.Mcordr(2, 13, 0, 0, 0, 0);

15.3.114 MemIn方法, Spel類別

描述

傳回指定記憶體I/O位元組埠的狀態。每個連接埠包含8個記憶體I/O位元。

語法

Function MemIn (PortNumber As Integer) As Integer Function MemIn (Label As String) As Integer

參數

- PortNumber 代表任一個記憶體I/O埠的整數運算式。
- Label
 包含記憶體I/O位元組標籤的字串運算式。

傳回值

介於0至255的整數值,代表輸入埠狀態。

另請參閱

In, InBCD, MemOut, MemSw, Sw, Off, On, Oport

MemIn範例

VB 例:

data = m_spel.MemIn(1)

C# 例:

data = m_spel.MemIn(1);

15.3.115 MemInW方法, Spel類別

描述

傳回指定記憶體I/O字元埠的狀態。每個字元埠包含16個記憶體I/O位元。

語法

Function MemInW (PortNumber As Integer) As Integer Function MemInW (Label As String) As Integer

參數

- PortNumber 代表記憶體I/O埠的整數值。
- Label
 包含記憶體I/O字元標籤的字串運算式。

傳回值

介於0至65535的整數值,代表輸入埠狀態。

另請參閱

In, InBCD, MemIn, MemSw, Sw, Off, On, Oport

MemInW範例

VB 例:

data = m_spel.MemInW(1)

C# 例:

data = m_spel.MemInW(1);

15.3.116 MemOff方法, Spel類別

描述

關閉記憶體I/O的指定位元。

語法

Sub MemOff (BitNumber As Integer) Sub MemOff (Label As String)

參數

- BitNumber 代表記憶體I/O位元的整數運算式。
- Label 代表記憶體I/O位元標籤的字串運算式。

另請參閱

In, InBCD, MemOut, MemSw, Sw, Off, On, Oport

MemOff範例

VB 例:

m_spel.MemOff(500)

C# 例:

m_spel.MemOff(500);

15.3.117 MemOn方法, Spel類別

描述

開啟記憶體I/O的指定位元。

語法

Sub MemOn (BitNumber As Integer) Sub MemOn (Label As String)

參數

- BitNumber 代表記憶體I/O位元的整數運算式。
- Label 代表記憶體I/O位元標籤的字串運算式。

另請參閱

In, InBCD, MemOut, MemSw, Sw, Off, On, Oport

MemOn範例

VB 例:

m_spel.MemOn(500)

C# 例:

m_spel.MemOn(500);

15.3.118 MemOut方法, Spel類別

描述

根據使用者指定的8位元值,同時設定8個記憶體I/O位元。

語法

Sub MemOut (PortNumber As Integer, Value As Integer) Sub MemOut (Label As String, Value As Integer)

參數

- PortNumber 代表記憶體I/O位元組的整數運算式。
- Label 包含記憶體I/O位元組標籤的字串運算式。
- Value 整數運算式,代表指定位元組的輸出組輸出模式。有效範圍:0~255。

另請參閱

In, InBCD, MemIn, MemSw, Sw, Off, On, Oport

MemOut範例

VB 例:

m_spel.MemOut(2, 25)

C# 例:

m_spel.MemOut(2, 25);

15.3.119 MemOutW方法, Spel類別

描述

根據使用者指定的16位元值,同時設定8個記憶體I/O位元。

語法

Sub MemOutW (PortNumber As Integer, Value As Integer) Sub MemOutW (Label As String, Value As Integer)

參數

- PortNumber 代表記憶體I/O字元的整數運算式。
- Label
 包含記憶體I/O字元標籤的字串運算式。
- Value
 使用運算式或數值指定輸出資料(介於0至65535的整數)。

另請參閱

In, InBCD, MemIn, MemSw, Sw, Off, On, Oport

MemOutW範例

VB 例:

m_spel.MemOutW(2, 25)

C# 例:

m_spel.MemOutW(2, 25);

描述

傳回指定的記憶體I/O位元狀態。

語法

Function MemSw (BitNumber As Integer) As Boolean Function MemSw (Label As String) As Boolean

參數

- BitNumber 代表任一個記憶體I/O位元的整數運算式。
- Label 代表記憶體I/O位元標籤的字串運算式。

傳回值

若指定記憶體I/O位元開啟, 會傳回True, 否則會傳回False。

另請參閱

In, InBCD, MemIn, Sw, Off, On, Oport

MemSw範例

VB 例:

```
If m_spel.MemSw(10) Then
    m_spel.On(2)
End If
```

```
if (m_spel.MemSw(10))
m_spel.On(2);
```

15.3.121 Move方法, Spel類別

描述

使用CP動作(直綫移動),將手臂從目前位置移至指定點。

語法

Sub Move (PointNumber As Integer) Sub Move (Point As SpelPoint) Sub Move (Point As SpelPoint, AttribExpr As String) Sub Move (PointExpr As String)

參數

每個語法都有一個參數,用以指定在Move動作期間手臂移動的結束點。此為線性插補動作結束時的最終位置。 PointNumber

從目前機器人的點記憶體(被保存在控制器中)中使用先前示教點的編號,以指定目標位置。

- Point
 透過使用SpelPoint資料類型來指定目標位置。
- AttribExpr 透過使用字串運算式來指定結束點屬性。
 可包括ROT、ECP、CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。
- PointExpr 透過使用字串運算式來指定目標位置。
 可包括ROT、ECP、CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP BGo, BMove, TGo, TMove Arch, ECP, Find, Till

Move範例

VB 例:

```
使用點編號指定的點
m_spel.Tool(1)
m_spel.Move(100)
使用SpelPoint指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.Move(pt)
使用點運算式指定的點
m_spel.Move("P0 /L /2 ROT")
m_spel.Move("P1 :Z(-20)")
使用平行處理
m_spel.Move("P1 \!D50; On 1; D90; Off 1\!")
使用標籤指定的點
m_spel.Move("pick")
```

// 使用點編號指定的點 m_spel.Tool(1); m_spel.Move(100);

// 使用SpelPoint指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.Move(pt);

// 使用點運算式指定的點
m_spel.Move("P0 /L /2 ROT");
m_spel.Move("P1 :Z(-20)");

// 使用平行處理 m_spel.Move("P1 \!D50; On 1; D90; Off 1\!");

// 使用標籤指定的點 m_spel.Move("pick");

15.3.122 Off方法, Spel類別

描述

關閉指定輸出。

語法 Sub Off (BitNumber As Integer) Sub Off (Label As String)

參數

- BitNumber 代表任一個標準或擴展輸出的整數運算式。
- Label
 包含輸出位元標籤的字串運算式。

另請參閱

On, Oport, Out, OutW

Off範例

VB 例:

m_spel.Off(1)

C# 例:

m_spel.Off(1);

15.3.123 OLRate 方法, Spel類別

描述

傳回指定關節的過載率。

語法

Function OLRate (JointNumber As Integer) As Single

參數

JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)

傳回值

傳回指定關節的過載率。傳回值範圍是0.0~2.0。

備註

OLRate會檢查循環是否造成關節過載。使用高負載的循環應用,會由於溫度和電流導致伺服錯誤。OLRate會檢查機器人是否容易出現伺服錯誤。

執行循環時,可以執行另一個任務來監視OLRate。如果有關節的OLRate超過1.0,則會發生伺服錯誤。

當過載時,最有可能發生伺服錯誤。可以在測試循環時,使用OLRate來確認速度和加減速度,以確保實際使用過程 中不會發生伺服錯誤。

爲獲得有效值,請在機器人操作期間執行本方法。

正常負載狀態下,不能使用本方法。

OLRate範例

VB 例:

```
Dim data As Single
data = m_spel.OLRate(1)
```

```
float data;
data = m_spel.OLRate(1);
```

15.3.124 On方法 · Spel類別

描述

開啟指定輸出。

語法

Sub On (BitNumber As Integer) Sub On (Label As String)

參數

- BitNumber 代表任一個標準或擴展輸出的整數運算式。
- Label
 包含輸出位元標籤的字串運算式。

另請參閱

Off, Oport, Out, OutW

On範例

VB 例:

m_spel.On(1)

C# 例:

m_spel.On(1);

描述

使用BCD(二進位十進碼)格式,同時設定8個輸出位元。

語法

OpBCD (PortNumber As Integer, Value As Integer) OpBCD (Label As String, Value As Integer)

參數

- PortNumber 代表I/O埠的整數。每個埠由8個輸出位(1位元組)組成。
- Value
 整數(0~99),代表指定埠的輸出模式。第二個數字(一位數)代表埠的下位4個輸出位;第一個數字(十位數) 代表埠的上位4個輸出位。

另請參閱

Off, Out, Sw

OpBCD範例

VB 例:

m_spel.OpBCD(1, 25)

C# 例:

m_spel.OpBCD(1, 25);

描述

傳回指定輸出位元的狀態。

語法

Function Oport (BitNumber As Integer) As Boolean Function Oport (Label As String) As Boolean

參數

- BitNumber 代表標準及擴展輸出位的整數運算式。
- Label 代表輸出位標籤的字串運算式。

傳回值

若指定輸出位元已開啟、會傳回True、否則會傳回False。

另請參閱 Off, On, OpBCD, Out, Sw

Oport範例

VB 例:

```
If m_spel.Oport(1) Then
    m_spel.On(2)
End If
```

15.3.127 Out方法, Spel類別

描述

同時讀取或設定8個輸出位元(一個位元組)。

語法

Sub Out (PortNumber As Integer, Value As Integer) Sub Out (Label As String, Value As Integer) Function Out (PortNumber As Integer) As Integer Function Out (Label As String) As Integer

參數

- PortNumber 代表輸出埠的整數。
- Label 代表輸出位元組標籤的字串運算式。
- Value
 整數(0~255),代表輸出埠的輸出模式。若以十六進位表示,範圍為&H0至&HFF。

傳回值

介於0-255的整數(包含連接埠值)。

另請參閱

InBCD, OpBCD, Oport, OutW, Sw

Out範例

VB 例:

m_spel.Out(1, 240)

C# 例:

m_spel.Out(1, 240);

15.3.128 OutReal 方法 · Spel類別

描述

讀取或設定輸出埠狀態的32位元浮點資料(符合IEEE754標準)。

語法

Function OutReal (WordPortNumber As Integer) As Single Sub OutReal (WordPortNumber As Integer, Value As Single)

參數

- WordPortNumber 代表輸出埠的整數。
- Value
 代表輸出資料的實數值

傳回值

返回輸出埠的狀態,在32位元浮點資料(符合IEEE754標準)中返回。

另請參閱

In, InBCD, InReaql, InW, Out, OutW

OutReal範例

VB 例:

Dim val As Single
val = m_spel.OutReal(32)

```
float val;
val = m_spel.OutReal(32);
```

15.3.129 OutW方法, Spel類別

描述

同時讀取或設定16個輸出位元(一個字元)。

語法

Sub OutW (PortNumber As Integer, Value As Integer) Sub OutW (Label As String, Value As Integer) Function OutW (PortNumber As Integer) As Integer Function OutW (Label As String) As Integer

參數

- PortNumber 代表輸出埠的整數。
- Label 代表輸出字元標籤的字串運算式。
- Value 整數(0~65535),代表輸出埠的輸出模式。若以十六進位表示,範圍為&H0至&HFFFF。

傳回值

介於0-65535的整數(包含連接埠值)。

另請參閱

InBCD, OpBCD, Oport, Out, Sw

OutW範例

VB 例:

m_spel.OutW(1, 240)

C# 例:

m_spel.OutW(1, 240);

15.3.130 PAgl方法, Spel類別

描述

傳回所選旋轉軸的關節角度,或所選線性軸的位置(來自指定點)。

語法

Function PAgl (PointNumber As Integer, JointNumber As Integer) As Single Function PAgl (Point As SpelPoint, JointNumber As Integer) As Single Function PAgl (Label As String, JointNumber As Integer) As Single

參數

- PointNumber
 代表目前機器人點記憶體中某個點之點編號的整數運算式。
- Point 先前初始化的SpelPoint。
- Label
 包含目前機器人點記憶體中某個點之點標籤的字串運算式。
- JointNumber 代表所需之關節編號的整數運算式。數值可以介於1~9。

傳回值

包含指定關節之角度的單一數值(以度或公釐為單位)。

另請參閱

Agl, Pls, CX - CT

PAgl範例

VB 例:

```
Dim tlAngle As Single
tlAngle = m spel.PAgl(1, 1)
```

```
float tlAngle;
tlAngle = m_spel.PAgl(1, 1);
```

15.3.131 Pallet方法, Spel類別

描述

定義棧板。

語法

Sub Pallet (PalletNumber As Integer, Point1 As String, Point2 As String, Point3 As String [, Point4 As String], rows As Integer, columns As Integer)

參數

- PalletNumber 以0至15的整數表示的棧板編號。
- Point1
 定義第一個棧板位置的點變數。
- Point2
 定義第二個棧板位置的點變數。
- Point3
 定義第三個棧板位置的點變數。
- Point4
 定義第四個棧板位置的點變數。
- Rows
 棧板橫向的點數目。每個數值均為介於1至32767的整數。
- Columns 棧板縱向的點數目(1~32767的整數)。

另請參閱

Jump, Go, SetPoint

Pallet範例

VB 例:

m_spel.Pallet(1, 1, 2, 3, 4, 3, 4)

C# 例:

m_spel.Pallet(1, 1, 2, 3, 4, 3, 4);

15.3.132 Pass方法, Spel類別

描述

指定通過指定點附近而不停止動作的PTP動作。

語法

Sub Pass(PointNumber As Integer) Sub Pass(PassExpr As String)

參數

 PointNumber 從目前機器人的點記憶體(被保存在控制器中)中使用先前示教點的編號,以指定通過點。

PassExpr

使用字串運算式來進行指定。 Point specification [, {On | Off | MemOn | MemOff} bit number [,point specification ...]] [LJM [Orientation flag]]

Point specification 指定點編號、P(運算式)或點標籤。如果點資料完整並依遞增或遞減順序列出,可使用一個冒號合 併兩個點編號,例如P(1:5)。

Bit number 使用整數或輸出標籤指定I/O輸出位或記憶體I/O位元進行開啟/關閉。 LJM 選用。使用LJM函數轉換起始座標、接近座標及目標座標。 Orientation flag 選用。指定LJM函數的方向旗標參數。

另請參閱

Accel, Go, Jump, Speed

Pass範例

VB 例:

```
m_spel.Jump(1)
m_spel.Pass(2) '使手臂接近P2, 並在到達P2之前執行以下命令
m_spel.On(2)
m_spel.Pass(3)
m_spel.Pass(4)
m_spel.Off(0)
m_spel.Pass(5)
```

```
m_spel.Jump(1);
mspel.Pass(2); //使手臂接近P2, 並在到達P2之前執行以下命令
m_spel.On(2);
m_spel.Pass(3);
m_spel.Pass(4);
m_spel.Off(0);
m_spel.Pass(5);
```

15.3.133 Pause方法, Spel類別

描述

使控制器中的所有正常SPEL+任務暫停。若機器人正在移動,將會立即減速至停止。

語法

Sub Pause ()

⚠ 注意

以Call方法執行的函數·無法透過Pause方法暫停。 如欲暫停·請使用Xqt方法。

另請參閱

Continue, EventReceived, Stop

Pause範例

VB 例:

```
Sub btnPause_Click()_
ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles btnPause.Click
m_spel.Pause()
btnPause.Enabled = False
btnContinue.Enabled = True
End Sub
```

```
void btnPause_Click(object sender, EventArgs e)
{
    m_spel.Pause();
    btnPause.Enabled = false;
    btnContinue.Enabled = true;
}
```

15.3.134 PDef方法, Spel類別

描述

傳回指定點的定義狀態。

語法

Function PDef (PointNumber As Integer) As Boolean

參數

PointNumber
 代表目前機器人點記憶體中某個點之點編號的整數運算式。

傳回值

若指定點已定義,會傳回True,否則會傳回False。

另請參閱

PDel

PDef範例

VB 例:

Dim plDefined As Boolean
plDefined = m_spel.PDef(1)

C# 例:

bool plDefined;
plDefined = m_spel.PDef(1);

15.3.135 PDel方法 · Spel類別

描述

刪除指定位置資料。

語法

Sub PDel (FirstPointNumber As Integer [, LastPointNumber As Integer])

參數

- FirstPointNumber 指定範圍內欲刪除之第一個點的整數運算式。
- LastPointNumber 指定範圍內欲刪除之最後一個點的整數運算式。如果省略,則只會刪除在FirstPointNumber中指定的點。

另請參閱

PDef, LoadPoints, Clear, SavePoints

PDel範例

VB 例:

```
m_spel.PDel(1, 10)
m_spel.SavePoints("model1.pts")
```

```
m_spel.PDel(1, 10);
m spel.SavePoints("model1.pts");
```

描述

傳回指定關節的峰值速度。

語法

Function PeakSpeed (JointNumber As Integer) As Single

參數

JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)

傳回值

傳回-1~1範圍中的實數值。

備註

本方法傳回一個帶符號的值,該值是關節的最大速度的絕對值。最大速度為1。峰值速度是-1~1範圍中的實數值。

請先執行PeakSpeedClear, 然後再顯示本方法來顯示關節的峰值速度。

如果是虛擬控制器或者乾運轉,就使用命令速度而不是實際速度來計算。 本方法不支援PG附加軸。

另請參閱

AvgSpeed, AvgSpeedClear, PeakSpeedClear

PeakSpeed範例

VB 例:

```
Dim val As Single
val = m_spel.PeakSpeed(1)
```

```
float val;
val = m spel.PeakSpeed(1);
```

15.3.137 PeakSpeedClear 方法 · Spel類別

描述

清除並初值化關節的峰值速度。

語法

Sub PeakSpeedClear ()

備註

使用本方法可以清除指定關節的峰值速度值。 執行PeakSpeed方法之前,必須先執行本方法。

本方法不支援PG附加軸。

另請參閱

AvgSpeed, AvgSpeedClear, PeakSpeed

PeakSpeedClear範例

VB 例:

m_spel.PeakSpeedClear()

C# 例:

m_spel.PeakSpeedClear();

15.3.138 PF_Abort 方法, Spel類別

描述

强制中斷指定零件的Part Feeding程序動作。

語法

Sub PF_Abort (PartID As Integer)

參數

 PartID 代表零件ID的整數值(1 16)

備註

立即中斷指定零件的Part Feeding程序。 與PF_Stop方法相異,本方法會中斷正在執行的回呼函式。 如果Part Feeding程序還沒有開始,就不會發生任何動作。

PF_Abort範例

VB 例:

m_spel.PF_Abort(1)

C# 例:

m_spel.PF_Abort(1);

15.3.139 PF_Backlight 方法, Spel類別

描述

打開或關閉給料器的嵌入式背光燈。

語法

Sub PF_Backlight (FeederNumber As Integer, State As Boolean)

參數

- FeederNumber 代表給料器編號的整數值
- State 指定On (True) 或 Off (False)

備註

如果系統自動運行視覺處理時,背光燈會自動On/Off。 若使用PF_Vision回呼函式,就可以使用本方法打開或關閉背光燈。

PF_Backlight範例

VB 例:

m_spel.PF_Backlight(1, True)

```
m_spel.PF_Backlight(1, true);
```

15.3.140 PF_BacklightBrightness 方法, Spel類別

描述

設定給料器嵌入式背光燈的亮度。

語法

Sub PF_Backlightness (FeederNumber As Integer, Brightness As Integer)

參數

- FeederNumber 代表給料器編號的整數值
- SBrightness
 用0 ~100 的數字指定亮度。

備註

一般情況下,可以使用[Part Feeding Configuration]對話方塊設定嵌入式背光燈的亮度。當要在運行中更改亮度時,就可以使用本方法。

PF_BacklightBrightness範例

VB 例:

m_spel.PFBacklightness(1, 80)

```
m_spel.PF_BacklightBrightness(1, 80);
```

描述

從零件ID取得零件名稱。

語法

Function PF_Name (PartID As Integer) As String

參數

 PartID 代表零件ID的整數值(1 16)

傳回值

傳回指定零件ID的名稱的字串。

備註

如果指定的零件ID無效,就傳回""(空白字符)。

PF_Name範例

VB 例:

Dim part1Name As String
Part1Name = m_spel.PF_Name(1)

```
string part1Name;
part1Name = m_spel.PF_Name(1);
```

15.3.142 PF_Number方法, Spel類別

描述

從零件名稱取得零件ID。

語法

Function PF_Number (PartName As String) As Integer

參數

 PartName 代表零件名稱的字串。

傳回值

傳回指定零件名稱的零件ID (整数1~16)。

備註

如果零件名稱不存在就傳回-1。 如果出現重複名稱的零件,就返回ID最小的零件。

PF_Number範例

VB 例:

```
Dim part1ID As Integer
Part1ID = m_spel.PF_Number("Part1")
```

```
int part1ID;
part1ID = m_spel.PF_Number("Part1");
```

15.3.143 PF_Start方法, Spel類別

描述

開始指定零件的Part Feeding程序。

語法

Sub PF_Start (PartID1 As Integer, [PartID2 As Integer], [PartID3 As Integer], [PartID4 As Integer])

參數

- PartID1 代表主零件ID的整數值(1 16)。
- PartID2
 代表從零件ID的整數值(1 16)。可以省略
- PartID3 代表從零件ID的整數值(1 16)。可以省略
- PartID4
 代表從零件ID的整數值(1 16)。可以省略

備註

在開始本方法之前,請先進行以下操作。

- 選擇要使用的機器人
- 馬達On
- 如要輸出日誌, 請執行PF_InitLog

執行本方法, 會產生一個新的任務, 并將控制回復給主叫程式。

發生這種情況時, Status回呼函式會在以下條件中執行。Part Feeding程序不會開始。

條件	Status回呼函式參數的Status值
零件ID無效	PF_STATUS_BAD_ID
零件參數設定無效 (沒有選擇Enabled選項)	PF_STATUS_BAD_PARAMETER
沒有完成校準	PF_STATUS_CAL_NOT_COMPLETE
發生錯誤	PF_STATUS_ERROR

本方法不能同時複數執行。如果執行了,就會繼續已經執行的處理。不會發生錯誤。 請在正常任務中執行PF_Start方法。如果在後台執行則會發生錯誤。

注意

如果在本方法中指定了不存在的零件ID,就會發生7600錯誤。

PF_Start範例

VB 例:

m_spel.PF_Start(1)

C# 例:

m_spel.PF_Start(1);

15.3.144 PF_Stop方法, Spel類別

描述

發佈Part Feeding程序終止的要求。 如果有正在執行的回呼函式,會等待執行完成。 之後,將會執行PF_CycleStop回呼函式,停止程序。

語法

Sub PF_Stop (PartID As Integer)

參數

PartID
 代表零件ID的整數值(1 16)

備註

停止Part Feeding程序。 與PFAbort方法相異,本方法會等待執行中的回呼函式完成。 回呼函式執行完成後,將會執行PF_CycleStop回呼函式。 如果沒有開始任何Part Feeding程序,就不會發生任何動作。

PF_Stop範例

VB 例:

m_spel.PF_Stop(1)

C# 例:

m_spel.PF_Stop(1);

15.3.145 PLabel 方法, Spel類別

描述

取得或設定指定點編號中定義的點標籤。

語法

Function PLabel (PointNumber As Integer) As String Sub PLabel (PointNumber As Integer, PointName As String)

參數

- PointNumber 代表點編號的整數值
- PointName 指定的點資料中使用的標籤的字串

傳回值

傳回指定點編號相對應的標籤。

另請參閱

PDef

PLabel範例

VB 例:

```
Dim pt1Label As String
Pt1Label = m_spel.PLabel(1)
```

```
string pt1Label;
pt1Label = m_spel.PLabel(1);
```

15.3.146 Plane方法, Spel類別

描述

定義平面。

語法

Sub Plane (PlaneNumber As Integer, Point As SpelPoint) Sub Plane (PlaneNumber As Integer, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single)

參數

- PlaneNumber 介於1-15的整數運算式,代表15個平面中要定義的平面。
- Point
 代表接近檢查平面之座標資料的點資料。
- X
 代表接近檢查平面之座標資料的點X座標。
- Y

代表接近檢查平面之座標資料的點Y座標。

• Z

代表接近檢查平面之座標資料的點Z座標。

• U

代表接近檢查平面之座標資料的點U座標。

- V
 代表接近檢查平面之座標資料的點V座標。
- W
 代表接近檢查平面之座標資料的點W座標。

另請參閱

PlaneClr, PlaneDef

Plane範例

VB 例:

m_spel.Plane(1, -5, 5, -10, 10, -20, 20)

C# 例:

m spel.Plane(1, -5, 5, -10, 10, -20, 20);
15.3.147 PlaneClr方法 · Spel類別

描述

清除(取消定義)平面。

語法

Sub PlaneClr (PlaneNumber As Integer)

參數

• PlaneNumber 介於1-15的整數, 代表15個平面中要清除的平面。

另請參閱

Plane, PlaneDef

PlaneClr範例

VB 例:

m_spel.PlaneClr(1)

C# 例:

m_spel.PlaneClr(1);

15.3.148 PlaneDef方法 · Spel類別

描述

傳回平面是否定義。

語法

Function PlaneDef (PlaneNumber As Integer) As Boolean

參數

 PlaneNumber 介於1至15的整數運算式,代表平面編號。

傳回值

若指定平面已定義,會傳回True,否則會傳回False。

另請參閱

Plane, PlaneClr

PlaneDef範例

VB 例:

x = m_spel.PlaneDef(1)

C# 例:

x = m_spel.PlaneDef(1);

描述

傳回目前位置各軸的目前編碼器脈衝計數。

語法

Function Pls (JointNumber As Integer) As Integer

參數

 JointNumber 要取得目前編碼器脈衝計數的特定軸。(1至9)

傳回值

包含指定關節之目前脈衝計數的整數。

另請參閱

Agl, Pulse

Pls範例

VB 例:

j1Pulses = m_spel.Pls(1)

C# 例:

j1Pulses = m_spel.Pls(1);

描述

清除並初值化峰值轉矩。

語法

Sub PTCLR ()

備註

本方法可清除指定關節的峰值轉矩。 執行PTRQ方法之前,必須先執行本方法。

另請參閱

ATCLR, ATRQ, PTRQ

PTCLR範例

VB 例:

m_spel.PTCLR ()

C# 例:

m_spel.PTCLR();

15.3.151 PTPBoost方法, Spel類別

描述

設定短距離PTP(點至點)動作的提升參數。

語法

Sub PTPBoost (BoostValue As Integer [, DepartBoost As Integer] [, ApproBoost As Integer])

參數

- BoostValue 介於0-100的整數。
- DepartBoost 整數(0~100),代表Jump動作時的Z座標起始調整值(可省略)。
- ApproBoost 整數 (0~100), 代表Jump動作時的Z座標接近調整值(可省略)。

另請參閱

PTPBoostOK

PTPBoost範例

VB 例:

```
m_spel.PTPBoost(50)
m_spel.PTPBoost(50, 30, 30)
```

```
m_spel.PTPBoost(50);
m_spel.PTPBoost(50, 30, 30);
```

15.3.152 PTPBoostOK方法, Spel類別

描述

傳回從目前位置到目標位置的PTP(點至點)動作是否為短距離。

語法

Function PTPBoostOK (PointNumber As Integer) As Boolean Function PTPBoostOK (Point As SpelPoint) As Boolean Function PTPBoostOK (PointExpr As String) As Boolean

參數

每個語法都有一個參數,用以指定要檢查的目標點。

- PointNumber 相對於目前機器人,從控制器的點記憶體中使用先前示教點的編號,以指定目標位置。
- Point 透過使用SpelPoint資料類型來指定目標位置。
- PointExpr
 透過使用字串運算式來指定目標位置。

傳回值

若將會使用PTPBoost, 會傳回True, 否則會傳回False。

另請參閱

PTPBoost

PTPBoostOK範例

VB 例:

```
If m_spel.PTPBoostOK(1) Then
    m_spel.Go(1)
End If
```

15.3.153 PTran方法, Spel類別

描述

執行相對關節移動(以脈衝為單位)。

語法

Sub PTran (JointNumber As Integer, Pulses As Integer)

參數

- JointNumber 要移動的關節
- Pulses
 要移動的脈衝數

另請參閱

JTran, Pulse

PTran範例

VB 例:

・以正方向將關節1移動5000脈衝。m_spel.PTran(1, 5000)

C# 例:

// 以正方向將關節1移動5000脈衝。 m_spel.PTran(1, 5000);

15.3.154 PTRQ方法 · Spel類別

描述

傳回指定關節的峰值轉矩。

語法

Function PTRQ (JointNumber As Integer) As Single

參數

JointNumber
 管道#的整數值(範圍: 1~ 機器人的關節數)

傳回值

傳回0~1的實數值。

另請參閱

ATCLR, ATRQ, PTCLR

PTRQ範例

VB 例:

Dim peakTorque As Single
peakTorque = m_spel.PTRQ(1)

C# 例:

float peakTorque;
peakTorque = m_spel.PTRQ(1);

15.3.155 Pulse方法, Spel類別

描述

透過PTP控制方式,將機器人手臂移至透過所有機器人關節的脈衝值所指定的點。

語法

Sub Pulse (J1Pulses As Integer, J2Pulses As Integer, J3Pulses As Integer, J4Pulses As Integer [, J5Pulses As Integer] [, J6Pulses As Integer] [, J7Pulses As Integer] [, J8Pulses As Integer] [, J9Pulses As Integer])

參數

 J1Pulses - J9Pulses 代表第1~9關節的脈衝值的整數 J5Pulses - J9Pulses可以省略



脈衝值必須介於各關節指定範圍內。

另請參閱

Go, Move, Jump

Pulse範例

VB 例:

```
m_spel.Pulse(5000, 1000, 0, 0)
```

C# 例:

m_spel.Pulse(5000, 1000, 0, 0);

15.3.156 Quit方法, Spel類別

描述

終止執行指定的任務。

語法

Sub Quit (TaskNumber As Integer) Sub Quit (TaskName As String)

參數

- TaskNumber 要中斷之任務的任務#(1~32)
- TaskName
 包含任務名稱的字串運算式。



如欲停止以Call方法執行的函數時,請使用Stop方法。

另請參閱

Halt, Resume, Xqt

Quit範例

VB 例:

m_spel.Quit(3)

C# 例:

m_spel.Quit(3);

15.3.157 RadToDeg方法 · Spel類別

描述

將徑度轉換成度。

語法

Function RadToDeg (Radians As Double) As Double

參數

Radians
 包含要轉換成度之徑度的雙精度運算式。

傳回值

包含以度為單位之轉換值的雙精度值。 另請參閱

DegToRad

RadToDeg範例

VB 例:

Dim deg As Double

deg = m_spel.RadToDeg(1)

C# 例:

double deg; deg = m_spel.RadToDeg(1);

15.3.158 RebootController方法, Spel類別

描述

對目前接連的控制器進行重新啟動。

語法

Sub RebootController (ShowStatusDialog As Boolean)

參數

 ShowStatusDialog 設定在重新啟動完成前,是否顯示狀態對話方塊的畫面。 True=顯示畫面、False=不顯示畫面

備註

使用ShowStatusDialog,顯示帶有進度條的對話方塊。可以使用對話方塊或者Abort方法,停止操作。

另請參閱

Abort

RebootController範例

VB 例:

m_spel.RebootController(True)

C# 例:

m_spel.RebootController(true);

15.3.159 RebuildProject方法, Spel類別

描述

完整重建在Project屬性中指定的目前專案。

語法

Sub RebuildProject ()

另請參閱

BuildProject, EnableEvent, EventReceived, Project, ProjectBuildComplete

RebuildProject範例

VB 例:

```
With m_spel
    .Project = "c:\EpsonRC80\projects\myproject\myproject.sprj"
    .RebuildProject()
End With
```

```
m_spel.Project = @"c:\EpsonRC80\projects\myproject\myproject.sprj";
m_spel.RebuildProject();
```

15.3.160 Recover方法, Spel類別

描述

Recover會將機器人移回安全防護打開時的所在位置。

語法

Function Recover () As Boolean

備註

Recover方法可以在安全防護關閉後用來開啟機器人馬達,並將機器人緩慢移回安全防護打開時的所在位置。成功完成Recover後,您可執行Cont方法來繼續該週期。若成功完成Recover,將會傳回True。若在恢復動作期間發生暫停、終止或安全防護打開的情況,Recover將會傳回False。

傳回值

若恢復動作已經完成,會傳回True,否則會傳回False。

另請參閱

Continue, Pause

Recover範例

VB 例: 此範例是先執行recover, 再執行continue

```
Sub btnCont_Click(
    ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btnCont.Click
    Dim sts As Boolean
    Dim answer As Integer
    sts = m_spel.Recover()
    If sts = False Then
        Exit Sub
End If
    answer = MsgBox("Ready to continue?", vbYesNo)
    If answer = vbYes Then
        m_spel.Continue()
EndIF
End sub
```

此範例顯示只要按下按鈕便可使用按鈕來執行recover的方法。若在恢復動作期間放開按鈕,則會發出pause並終止恢 復動作。若按住按鈕直到完成恢復動作,會隨即顯示一訊息。

```
Sub btnRecover_MouseDown(
    ByVal sender As System.Object,
    ByVal e As System.Windows.Forms.MouseEventArgs) _
    Handles btnRecover.MouseDown
Dim sts As Boolean
sts = m_spel.Recover()
If sts = True Then
    MsgBox("Recover complete")
EndIf
End Sub
Sub btnRecover_MouseUp(
    ByVal sender As System.Object,
    ByVal e As System.Windows.Forms.MouseEventArgs) _
Handles btnRecover.MouseUp
```

```
m_spel.Pause()
End Sub
```

C#例: 此範例是先執行recover, 再執行continue

```
void btnCont_Click(object sender, EventArgs e)
{
    bool sts;
    DialogResult answer;
    sts = m_spel.Recover();
    if (sts == true) {
        answer = MessageBox.Show("Continue?", "",
            MessageBoxButtons.YesNo);
        If (answer == DialogResult.Yes)
            m_spel.Continue();
    }
}
```

此範例顯示只要按下按鈕便可使用按鈕來執行recover的方法。若在恢復動作期間放開按鈕,則會發出pause並終止恢 復動作。若按住按鈕直到完成恢復動作,會隨即顯示一訊息。

```
void btnCont_Click(object sender, EventArgs e)
{
    bool sts;
    sts = m_spel.Recover();
    if (sts == true)
        MessageBox.Show("Recover complete");
}
void btnRecover_MouseUp(oject sender, EventArgs e)
{
    m_spel.Pause();
}
```

15.3.161 Reset方法 · Spel類別

描述

將控制器重置為初始狀態。

語法

Sub Reset ()

另請參閱

ResetAbort

Reset範例

VB 例:

m_spel.Reset

C# 例:

m_spel.Reset();

15.3.162 ResetAbort方法 · Spel類別

描述

重置以Stop方法設定的終止旗標。

語法

Sub ResetAbort ()

備註

當執行Stop方法且沒有其他Spel方法處於週期中時,下一個Spel方法將會產生使用者終止錯誤。程序到此結束;不論 是在何時發出Stop,目前執行Spel方法的常式都將接收到錯誤。使用ResetAbort可清除此狀況。



ResetAbortEnabled屬性必須設為True·ResetAbort功能才可正常運作。

另請參閱

Abort, Reset, ResetAbortEnabled

ResetAbort範例

VB 例:

```
Sub btnMcal_Click() Handles btnMcal.Click
  m_spel.ResetAbort()
  m_spel.MCal()
End Sub
```

```
void btnMCal_Click(object sender, EventArgs e)
{
    m_spel.ResetAbort();
    m_spel.MCal();
}
```

15.3.163 Resume方法 · Spel類別

描述

使遭到Halt方法暫停的任務重新開始。

語法

Sub Resume (TaskNumber As Integer) Sub Resume (TaskName As String)

參數

- TaskNumber
 已中斷之任務的任務#。任務#的範圍介於1至32。
- TaskName
 包含任務名稱的字串運算式。

另請參閱

Quit, Xqt

Resume範例

VB 例:

m_spel.Resume(2)

C# 例:

m_spel.Resume(2);

15.3.164 RunDialog方法, Spel類別

描述

執行Epson RC+ 8.0對話方塊。

語法

Sub RunDialog (DialogID As SpelDialogs [, Parent As Form])

參數

- DialogID 要執行之Epson RC+ 8.0對話方塊的ID。
- Parent 視窗的父.NET表單(選用)。

另請參閱

ShowWindow

RunDialog範例

VB 例:

15.3.165 SavePoints方法 · Spel類別

描述

保存目前機器人的點至檔案。

語法

Sub SavePoints (PointFileName As String)

參數

 PointFileName 在目前專案中保存之點的檔案名稱。

另請參閱

LoadPoints

SavePoints範例

VB 例:

```
With m_spel
.SavePoints("part1.pts")
End With
```

C# 例:

m_spel.SavePoints("part1.pts");

15.3.166 Sense方法, Spel類別

描述

指定輸入條件,符合條件時會停止目標位置上方的機器人,以完成進行中的Jump。

語法

Sub Sense (Condition As String) As Boolean

參數

 Condition 指定I/O條件。有關詳細資訊,請參閱SPEL+語言參考的「Sense」。

另請參閱

Jump, JS

Sense範例

VB 例:

```
With m_spel
    .Sense("Sw(1) = On")
    .Jump("P1 SENSE")
    stoppedOnSense = .JS()
End With
```

```
m_spel.Sense("Sw(1) = On");
m_spel.Jump("P1 SENSE");
stoppedOnSense = m_spel.JS();
```

15.3.167 SetIODef方法, Spel類別

描述

設定輸入、輸出或記憶體I/O位元、位元組或字元的I/O標籤與描述。

語法

Sub SetIODef (Type As SpelLabelTypes, Index As Integer, Label As String, Description As String)

參數

- Type 如下指定I/O類型: InputBit = 1 InputByte = 2 InputWord = 3 OutputBit = 4 OutputByte = 5 OutputWord = 6 MemoryBit = 7 MemoryByte = 8 MemoryWord = 9 InputReal = 10 OutputReal = 11
- Index 指定位元或埠號。
- Label 指定新建標籤。
- Description 指定新建描述。

備註

使用SetIODef可定義任何I/O點的標籤與描述。

另請參閱

GetIODef

SetIODef範例

VB 例:

```
Dim label, desc As String
label = "StartCycle"
desc = "Starts the robot cycle"
m_spel.SetIODef(SpelLabelTypes.InputBit, 0, label, desc)
```

```
string label, desc;
label = "StartCycle";
desc = "Starts the robot cycle";
m spel.SetIODef(SpelLabelTypes.InputBit, 0, label, desc);
```

15.3.168 SetPoint方法, Spel類別

描述

設定目前機器人某個點的座標資料。

語法

Sub SetPoint(PointNumber As Integer, Point As SpelPoint) Sub SetPoint(PointLabel As String, Point As SpelPoint)

Sub SetPoint(PointNumber As Integer, X As Single, Y As Single, Z As Single, U As Single) Sub SetPoint(PointLabel As String, X As Single, Y As Single, Z As Single, U As Single)

Sub SetPoint(PointNumber As Integer, X As Single, Y As Single, Z As Single, U As Single,Local As Integer, Hand As SpelHand)

Sub SetPoint(PointLabel As String, X As Single, Y As Single, Z As Single, U As Single,Local As Integer, Hand As SpelHand)

Sub SetPoint(PointNumber As Integer, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single) Sub SetPoint(PointLabel As String, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single)

Sub SetPoint(PointNumber As Integer, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single, Local As Integer, Hand As SpelHand, Elbow As SpelElbow, Wrist As SpelWrist, J4Flag As Integer, J6Flag As Integer) Sub SetPoint(PointLabel As String, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single, Local As Integer, Hand As SpelHand, Elbow As SpelElbow, Wrist As SpelWrist, J4Flag As Integer, J6Flag As Integer)

Sub SetPoint(PointNumber As Integer, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single, S As Single, T As Single) Sub SetPoint(PointLabel As String, X As Single, Y As Single, Z As Single, U As Single, V As Single, W As Single, S As Single, T As Single)

Sub SetPoint(PointNumber As Integer, PointExpr As String) Sub SetPoint(PointLabel As String, PointExpr As String)

參數

- PointNumber 指定目前機器人點記憶體中某個點之點編號的整數運算式。
- X 指定點的X座標。
- Y 指定點的Y座標。
- Z 指定點的Z座標。
- U 指定點的U座標。
- V 指定點的V座標。
- W
 指定點的W座標。
- S 指定點的S座標。
- T 指定點的T座標。

- Local 指定點的本地編號。沒有本地時請使用0。
- Hand 指定點的手部方向。
- Elbow 指定點的肘部姿勢。
- Wrist 指定點的腕部方向。
- PointExpr 使用運算式指定的點 注意 請勿將整數值輸入至X、Y、Z、U、V、W、S及T參數。使用單一變數或直接輸入單一類型值。

另請參閱

GetPoint, LoadPoints, SavePoints

SetPoint範例

要執行的RC+專案的SPEL+程式範例:

```
Global Integer P_WorkDetect
Global Real offsetX, offsetY, heightZ, offsetU
Function SetOffset
P_WorkDetect = 5
    offsetX = 0.5
    offsetY = 0.3
    heightZ = 10.0
    offsetU = 1.2
Fend
```

1 CHU

VB 例:

```
Dim pt As SpelPoint
' 取得P1的座標
pt = m_spel.GetPoint(1)
' 變更座標
pt.U = pt.U - 10.5
' 指定姿勢旗標
pt.J1Flag = 1
m_spel.SetPoint(1, pt)
' 執行RC+專案的SPEL+函數
m_spel.Call("SetOffset")
' 使用字串運算式來設定座標及姿勢
m_spel.SetPoint(2, "P(P_WorkDetect) +X(offsetX) +Y(offsetY) :Z(heightZ) +U
(offsetU) /A /J1F1")
```

C# 例:

SpelPoint pt;
// 取得P1的座標
pt = m_spel.GetPoint(1);
// 變更座標
pt.U = pt.U - 10.5;
// 指定姿勢旗標
pt.J1Flag = 1;

m_spel.SetPoint(1, pt);
// 執行RC+專案的SPEL+函數
m_spel.Call("SetOffset")
// 使用字串運算式來設定座標及姿勢
m_spel.SetPoint(2, "P(P_WorkDetect) +X(offsetX) +Y(offsetY) :Z(heightZ) +U
(offsetU) /A /J1F1");

15.3.169 SetVar方法, Spel類別

描述

設定控制器中SPEL+全域保留變數的值。

語法

Sub SetVar (VarName As String, Value As Object)

參數

- VarName SPEL+全域保留變數名稱
- Value 新建值。

備註

您可使用SetVar來設定單一變數及陣列變數的值。請參閱以下範例。

另請參閱

GetVar

SetVar範例

VB 例:

```
m_spel.SetVar("g_myIntVar", 123)
```

```
Dim i, myArray(10) As Integer
For i = 1 To 10
  myArray(i) = i
Next i
m_spel.SetVar("g_myIntArray", myArray)
```

```
m_spel.SetVar("g_myIntArray(1)", myArray(1))
```

15.3.170 SFree方法 · Spel類別

描述

從伺服系統控制,釋放指定的機器人軸。

語法

Sub SFree () Sub SFree (ParamArray Axes() As Integer)

參數

Axes
 包含要釋放之各機器人軸的某一元素之整數參數陣列。您可指定介於1-9的管道#。

另請參閱

SLock

SFree範例

VB 例:

' SFree軸1和2 m_spel.SFree(1, 2)

C# 例:

// SFree軸1和2
m_spel.SFree(1, 2);

15.3.171 ShowWindow方法, Spel類別

描述

顯示Epson RC+ 8.0視窗。

語法

Sub ShowWindow (WindowID As SpelWindows [, Parent As Form])

參數

- 視窗ID 要顯示的Epson RC+ 8.0視窗的ID。
- Parent 視窗的父.NET表單(選用)。

備註

您可使用Parent參數指定視窗的.NET父項。若無法使用.NET父表單,您必須省略Parent參數並使用 ParentWindowHandle屬性來設定父項的控制代碼。

另請參閱

HideWindow, ParentWindowHandle, RunDialog, ServerOutOfProcess

ShowWindow範例

VB 例:

```
Sub btnShowIOMonitor_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnShowIOMonitor.Click
    m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, Me)
End Sub
```

15.3.172 SimGet 方法 · Spel類別

描述

取得模擬器的各種物件屬性的設定值。

語法

Sub SimGet (Object As String, Property As SpelSimProps, ByRef Value As Boolean)

Sub SimGet (RobotName As String, HandName As String, Property As SpelSimProps, ByRef Value As Boolean) Sub SimGet (Object As String, Property As SpelSimProps, ByRef Value As Double)

Sub SimGet (RobotName As String, HandName As String, Property As SpelSimProps, ByRef Value As Double) Sub SimGet (Object As String, Property As SpelSimProps, ByRef Value As Integer)

Sub SimGet (RobotName As String, HandName As String, Property As SpelSimProps, ByRef Value As Integer) Sub SimGet (Object As String, Property As SpelSimProps, ByRef Value As Boolean)

Sub SimGet (RobotName As String, HandName As String, Property As SpelSimProps, ByRef Value As String)

參數

Object 代表取得了屬性值的物件名稱的字串變數

- RobotName
 代表安裝了"Hand"中指定夾具的機器人名稱的字串變數
- HandName
 代表取得屬性值的夾具名稱的字串變數
- Property 取得值的屬性的名稱。
- Value
 代表傳回值得變數

備註

本方法用於取得模擬器中的各種物件的屬性設定值。

有關各屬性的詳細資訊,請參閱以下手冊。 "Epson RC+ 8.0 SPEL+ 語言參考 - SimGet"

另請參閱

SimSet

SimGet範例

VB 例:

Dim posX As Double
m_spel.SimGet("SBox_1", SpelSimProps.PositionX, posX)

```
double posX;
SimGet("SBox_1", SpelSimProps.PositionX, out posX);
```

15.3.173 SimResetCollision方法, Spel類別

描述

重設碰撞偵測。

語法

Sub SimResetCollision ()

備註

執行本方法后,如果機器人和物件之間沒有發生碰撞,則重設碰撞狀態並更新模擬器的3D顯示。如果機器人和物件 之間發生碰撞,則不會解除碰撞狀態,模擬器的3D顯示也不會更新。

有關詳細資訊,請參閱以下手冊。 "Epson RC+ 8.0 SPEL+ 語言參考 - SimSet"

另請參閱

SimSet

SimResetCollision範例 VB例:

m_spel.SimResetCollision ()

_

C# 例:

m_spel.SimResetCollision ();

15.3.174 SimSet 方法, Spel類別

描述

設定模擬器中各種物件的屬性。還可以進行機器人動作和物件操作,以及模擬器設定等操作。

語法

Sub SimSet (Object As String, Property As SpelSimProps, Value As Boolean)

Sub SimSet (RobotName As String, HandName As String, Property As SpelSimProps, Value As Boolean) Sub SimSet (Object As String, Property As SpelSimProps, Value As Integer)

Sub SimSet (RobotName As String, HandName As String, Property As SpelSimProps, Value As Integer) Sub SimSet (Object As String, Property As SpelSimProps, Value As Double)

Sub SimSet (RobotName As String, HandName As String, Property As SpelSimProps, Value As Double) Sub SimSet (Object As String, Property As SpelSimProps, Value As String)

Sub SimSet (RobotName As String, HandName As String, Property As SpelSimProps, Value As String)

參數

- Object
 代表取得了屬性值的物件名稱的字串變數
- RobotName
 代表安裝了"Hand"中指定夾具的機器人名稱的字串變數
- HandName
 代表取得屬性值的夾具名稱的字串變數
- Property 設定了新值的屬性的名稱
- Value
 新值的運算式

備註

本方法用於設定和操作模擬器中的各種物件的屬性、機器人動作個和更改模擬器設定。

屬性不能指定SpelSimProps.Type。

有關各屬性的詳細資訊,請參閱以下手冊。 "Epson RC+ 8.0 SPEL+ 語言參考 - SimSet"

另請參閱

SimGet

SimSet範例

VB 例:

```
m spel.SimSet ("SBox 1", SpelSimProps.PositionX, 100.0)
```

C# 例:

m_spel.SimSet("SBox_1", SpelSimProps.PositionX, 100.0);

15.3.175 SimSetParent 方法, Spel類別

描述

設定物件的操作。

語法

Sub SimSetParent (Object As String) Sub SimSetParent (Object As String, ParentObject As String)

參數

- Object 代表設定了父層物件的物件名稱的字串變數
- ParentObject
 代表父層物件的字串變數

備註

對於"Object"中指定的物件,會將"ParentObject"中指定物件設置為其父層物件。可以省略"ParentObject"。這種情況下"Object"中指定的物件則為父層物件。例如,如果"Object"中指定的物件是某個物件的子物件,則會解除對其子物件的設定。 此外,如果"Object"中指定的物件是零件或者機械臂安裝設備,則無法指定父層物件。 有關SetParent 中可以指定的物件,請參閱以下手冊。

"Epson RC+ 8.0 SPEL+ 語言參考 - SimSet"

如果使用攝像機物件,只有設定為固定攝像機時,才可以使用SetParent。

另請參閱

SimSet

SimSetParent範例

VB 例:

m_spel.SimSetParent ("SBox_1")

C# 例:

m_spel.SimSetParent ("SBox_1");

15.3.176 SimSetPick 方法, Spel類別

描述

使用指定的機器人抓取物件。

語法

Sub SimSetPick (RobotName As String, Object As String) Sub SimSetPick (RobotName As String, Object As String, ToolNumber As Integer)

參數

- RobotName 代表Pick的機器人名稱的字串變數
- Object 代表被Pick的物件名稱的字串變量
- ToolNumber 代表Pick時使用的Tool編號的運算式

備註

"Robot"中指定的機器人會抓取"Object"中指定的物件。被抓取的物件則會被登錄為機器人的一部分。然後在Tool中 指定任意工具編號,就可以用指定的工具進行抓取。如果不指定"Tool",則會使用Tool0進行抓取。 無法抓取已注冊為零件的物件或設為機械手安裝設備的物件。此外,也無法抓取攝像機。

有關詳細資訊,請參閱以下手冊。 "Epson RC+ 8.0 SPEL+ 語言參考 - SimSet"

另請參閱

SimGet, SimSet, SimSetPlace

SimSetPick範例

VB 例:

```
m_spel.SimSetPick ("Robot1", "SBox_1", 1)
```

C# 例:

m_spel.SimSetPick ("Robot1", "SBox_1", 1);

15.3.177 SimSetPlace 方法, Spel類別

描述

使用指定的機器人放置物件。

語法

Sub SimSetPlace (RobotName As String, Object As String)

參數

- RobotName 代表Place機器人名稱的字串變數
- Object 字串變數,代表被Place的物件名稱。

備註

"Robot"中指定的機器人會放置"Object"中指定的物件。被防止的物件將被解除該機器人零件的注册。 不可放置已被解除零件注册的物件。

有關詳細資訊,請參閱以下手冊。 "Epson RC+ 8.0 SPEL+ 語言參考 - SimSet"

另請參閱

SimGet, SimSet, SimSetPick

SimSetPlace範例

VB 例:

m_spel.SimSetPlace ("Robot1", "SBox_1")

C# 例:

m spel.SimSetPlace ("Robot1", "SBox 1");

15.3.178 Shutdown方法 · Spel類別

描述

關閉或重啟Windows。

語法

Sub Shutdown (Mode As SpelShutdownMode)

參數

Mode
 0 = 關閉Windows。1 = 重啟Windows。

另請參閱

Reset

Shutdown範例

VB 例:

' 重啟Windows m_spel.Shutdown(1)

C# 例:

// 重啟 Windows m_spel.Shutdown(1);

15.3.179 SLock方法, Spel類別

描述

將指定的機器人軸傳回伺服系統控制。

語法

Sub SLock () Sub SLock (ParamArray Axes() As Integer)

參數

Axes
 包含要鎖定之各機器人軸的一個元素的整數參數陣列。您可指定介於1-9的管道#。

另請參閱

SFree

SLock範例

VB 例:

'將軸1和2傳回伺服系統控制。

m_spel.SLock(1, 2)

C# 例:

// 將軸1和2傳回伺服系統控制。 m_spel.SLock(1, 2);
15.3.180 Speed方法, Spel類別

描述

指定要搭配PTP指令Go、Jump及Pulse使用的手臂速度。

語法

Sub Speed (PointToPointSpeed As Integer[, JumpDepartSpeed As Integer] [, JumpApproSpeed As Integer])

參數

- PointToPointSpeed 指定要搭配PTP指令Go、Jump及Pulse使用的手臂速度。
- JumpDepartSpeed
 Jump命令時,代表向上動作速度的整數(1~100)。
- JumpApproSpeed
 Jump命令時,代表向下動作速度的整數(1~100)。

另請參閱

Accel, Jump, Go

Speed範例

VB 例:

m_spel.Speed(50)

C# 例:

m_spel.Speed(50);

15.3.181 SpeedR方法, Spel類別

描述

指定使用ROT時的工具旋轉速度。

語法

Sub SpeedR (RotationSpeed As Single)

參數

RotationSpeed
 以0.1以上實數值指定工具姿勢變化的速度(單位:deg/sec)。

另請參閱

Arc, Arc3, BMove, Jump3CP, Power, TMove

SpeedR範例

VB 例:

m_spel.SpeedR(100)

C# 例:

m_spel.SpeedR(100);

描述

指定要搭配連續路徑指令Jump3CP、Move、Arc及CVMove使用的手臂速度。

語法

Sub SpeedS (LinearSpeed As Single [, JumpDepartSpeed As Single][, JumpApproSpeed As Single])

參數

- LinearSpeed 指定在CP命令的Jump3CP、Move、Arc及CVMove使用的手臂速度。
- JumpDepartSpeed 實數或運算式(單位:mm/sec),代表Jump、Jump3CP的起始速度。
- JumpApproSpeed 實數或運算式(單位:mm/sec),代表Jump、Jump3CP的接近速度。

另請參閱

AccelS, Jump3CP, Move, TMove

SpeedS範例

VB 例:

m_spel.SpeedS(500)

C# 例:

m_spel.SpeedS(500);

15.3.183 Start方法, Spel類別

描述

啟動一個SPEL+程式。

語法

Sub Start (ProgramNumber As Integer)

參數

 ProgramNumber 對於要開始的程式編號以及SPEL+中內建的64個main函數,如以下對照表所示。範圍介於0至63。

程式編號	SPEL+函數名稱
0	main
1	main1
2	main2
3	main3
4	main4
5	main5
63	main63

備註

執行Start時,控制將會立即回到調用程式。您無法啟動正在執行的程式。請注意,Start會清除控制器中的全域變數 且會載入預設的機器人點。

另請參閱

Continue, Pause, Stop, Xqt

Start範例

VB 例:

```
Sub btnStart_Click(
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles btnStart.Click
        m_spel.Start(0)
End Sub
```

```
void btnStart_Click(object sender, EventArgs e)
{
    m_spel.Start(0);
}
```

15.3.184 StartBGTask方法, Spel類別

描述

啟動一個SPEL+任務作為背景任務。

語法

Sub StartBGTask (FuncName As String)

參數

 FuncName 指定要執行之函數的名稱。

備註

使用StartBGTask可在控制器中啟動Spel+背景任務。背景任務必須在控制器中啟用。

請注意,BGMain會在控制器切換至自動模式時自動啟動,因此通常不需要使用StartBGTask。 StartBGTask是提供用於需要停止所有任務、接著重啟背景任務的情況。

另請參閱

Call, Start, Stop, Xqt

StartBGTask範例

VB 例:

```
    停止所有任務,包含背景任務
    m_spel.Stop(SpelStopType.StopAllTasks)
    …
    m_spel.RebuildProject()
```

```
・ 重新開始主要背景任務m_spel.StartBGTask("BGMain")
```

C# 例:

```
// 停止所有任務,包含背景任務
m_spel.Stop(SpelStopType.StopAllTasks);
...
m_spel.RebuildProject();
```

// 重新開始主要背景任務
m_spel.StartBGTask("BGMain");

15.3.185 Stat方法 · Spel類別

描述

傳回控制器狀態。

語法

Function Stat (Address As Integer) As Integer

參數

 Address 指定代表控制器狀態的地址。(0至-2的整數)

傳回值

傳回代表控制器狀態的4位元組值。(請參閱下表。)

地址	Bit		位元開啟時的控制器狀態
0	0-15	&H1- &H8000	任務1至16正在執行(Xqt)或處於停止狀態
	16	&H10000	任務正在執行
	17	&H20000	暫停狀態
	18	&H40000	錯誤狀態
	19	&H80000	TEACH模式
	20	&H100000	緊急停止狀態
	21	&H200000	低運行功率模式
	22	&H400000	安全防護打開
	23	&H800000	啟動開關開啟
	24	&H1000000	未定義
25 &H2000000 未		&H2000000	未定義
	26	&H4000000	測試模式
	27	&H8000000	T2模式狀態
28-31		28-31	未定義
1	0	&H1	符合JumpSense陳述式中的條件時,目標位置上方的停止日誌。 (此日誌會在執行其他Jump陳述式時消除)。
	1	&H2	滿足Go/Jump/MoveTill陳述式中的條件時,中間移動位置的停止日誌。(此日 誌會在執行其他Go/Jump/MoveTill陳述式時消除)
	2	&H4	未定義

地址	Bit		位元開啟時的控制器狀態
	3	&H8	偵測到Trap陳述式時停止進行中動作的日誌。
	4	&H10	馬達開啟狀態
	5	&H20	起始點位置
	6	&H40	低運行功率模式
	7	&H80	未定義
	8	&H100	關節#4已接合。
	9	&H200	關節#3已接合。
	10	&H400	關節#2已接合。
	11	&H800	關節#1已接合。
	12	&H1000	關節#6已接合。
	13	&H2000	關節#5已接合。
	14	&H4000	T軸已接合。
	15	&H8000	S軸已接合。
	16	&H10000	關節#7已接合。
	17-31		未定義
2	0-15	&H1- &H8000	任務17至32正在執行(Xqt)或處於停止狀態

另請參閱

EStopOn, PauseOn, SafetyOn

Stat範例

VB 例:

Dim ctr_stat As Integer
ctr_stat = m_spel.Stat(0)

```
int ctr_stat;
ctr_stat = m_spel.Stat(0);
```

15.3.186 Stop方法 · Spel類別

描述

停止控制器中執行的所有一般SPEL+任務,以及選擇性停止所有背景任務。

語法

Sub Stop () Sub Stop (SpelStopType StopType)

參數

StopType

指定僅停止正常任務(StopNormalTasks)或停止所有任務(StopAllTasks)。(可省略)若省略,則會指定 StopNormalTasks。

\Lambda 注意

如果在ResetAbortEnabled設為True時執行Stop方法,則執行Start或Reset方法時會出現10101錯誤。

若要解除錯誤,請在執行Stop方法後執行ResetAbort方法。

另請參閱

Continue, Pause, ResetAbort, ResetAbortEnabled, Start, SpelStopType

Stop範例

VB 例:

```
Sub btnStop_Click( _______
ByVal sender As System.Object, _____
ByVal e As System.EventArgs) ______
Handles btnStop.Click
m_spel.Stop()
End Sub
```

```
void btnStop_Click(object sender, EventArgs e)
{
    m_spel.Stop();
}
```

15.3.187 Sw方法 · Spel類別

描述

傳回選取的輸入位元狀態。

語法

Function Sw (BitNumber As Integer) As Boolean Function Sw (Label As String) As Boolean

參數

- BitNumber 代表任一個標準或擴展輸入的整數運算式。
- Label
 包含輸入位元標籤的字串運算式。

傳回值

若指定輸入位元已開啟、會傳回True、否則會傳回False。

另請參閱

In, InBCD, MemSw, Off, On, Oport

Sw範例

VB 例:

```
If m_spel.Sw(1) Then
    m_spel.On(2)
End If
```

```
if (m_spel.Sw(1))
    m_spel.On(2);
```

15.3.188 TargetOK方法 · Spel類別

描述

傳回指示目前位置到目標位置的PTP(點至點)動作是否可行的狀態。

語法

Function TargetOK (PointNumber As Integer) As Boolean Function TargetOK (Point As SpelPoint) As Boolean Function TargetOK (PointExpr As String) As Boolean

參數

每個語法都有一個參數,用以指定要檢查的目標點。

 PointNumber 從目前機器人的點記憶體(被保存在控制器中)中使用先前示教點的編號,以指定終點。
 Point 透過使用SpelPoint資料類型來指定目標點。

PointExpr
 透過使用字串運算式來指定終點。

傳回值

若目標可以從目前位置移動,會傳回True,否則會傳回False。

另請參閱

Go, Jump, Move, TGo, TMove

TargetOK範例

VB 例:

```
If m_spel.TargetOK("P1 /F") Then
    m_spel.Go("P1 /F")
End If
```

```
if (m_spel.TargetOK("P1 /F"))
    m spel.Go("P1 /F");
```

15.3.189 TasksExecuting方法, Spel類別

描述

若正在執行任何SPEL+任務,會傳回True。

語法

Function TasksExecuting () As Boolean

傳回值

若正在執行任何SPEL+任務,會傳回True,否則會傳回False。

另請參閱 TaskState N

TaskState, Xqt

TasksExecuting範例

VB 例:

tasksRunning = m_spel.TasksExecuting()

C# 例:

tasksRunning = m_spel.TasksExecuting();

15.3.190 TaskState方法, Spel類別

描述

傳回任務的狀態。

語法

Function TaskState (TaskNumber As Integer) As SpelTaskState Function TaskState (TaskName As String) As SpelTaskState

參數

- TaskNumber 傳回執行狀態的任務#。
- TaskName
 包含任務名稱的字串運算式。

傳回值

SpelTaskState值。

另請參閱 TasksExecuting, Xqt

TaskState範例

VB 例:

```
Dim taskState As SpelTaskState
taskState = m_spel.TaskState(2)
```

```
SpelTaskState taskState;
taskState = m_spel.TaskState(2);
```

15.3.191 TeachPoint方法, Spel類別

描述

執行用以允許操作員步進示教一個點的對話方塊。

語法

Function TeachPoint (PointFile As String, PointNumber As Integer, Prompt As String) As Boolean Function TeachPoint (PointFile As String, PointName As String, Prompt As String) As Boolean Function TeachPoint (PointFile As String, PointNumber As Integer, Prompt As String, Parent As Form) As Boolean Function TeachPoint (PointFile As String, PointName As String, Prompt As String, Parent As Form) As Boolean

參數

- PointFile
 包含點檔案名稱的字串。
- PointNumber 要示教的點編號。
- PointName 點標籤的字串運算式。
- Prompt
 包含顯示於示教對話方塊底部之說明文字的字串。
- Parent 視窗的父.NET表單(選用)。

傳回值

若操作員按一下示教按鈕, 會傳回True; 若操作員按一下取消, 會傳回False。

備註

使用TeachPoints可讓操作員在控制器中示教一個機器人點。執行TeachPoints時,點文件會從控制器載入。按一下示教按鈕時,點會在控制器中示教,且點文件會儲存至控制器。

TeachPoint範例

VB 例:

```
Sub btnTeachPick_Click( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles btnTeachPick.Click
Dim sts As Boolean
Dim prompt As String
prompt = "Jog to Pick position and click Teach"
sts = m_spel.TeachPoint("points.pts", 1, prompt)
End Sub
```

C# 例:

```
void btnTeachPick_Click(object sender, EventArgs e){
   bool sts;
   string prompt;

   prompt = "Jog to Pick position and click Teach";
   sts = m_spel.TeachPoint("points.pts", 1, prompt);
```

}

15.3.192 Till方法, Spel類別

描述

指定事件條件,符合條件時會減速並停止中間位置的機器人,以完成進行中的動作命令(Jump、Go、Move等)。

語法

Sub Till (Condition As String) As Boolean

參數

 Condition 指定I/O條件。有關詳細資訊,請參閱SPEL+語言參考的「Till」。

另請參閱

Go, Jump, JS, Sense, TillOn

Till範例

VB 例:

```
With m_spel
    .Till("Sw(1) = On")
    .Go("P1 TILL")
End With
```

```
m_spel.Till("Sw(1) = On");
m_spel.Go("P1 TILL");
```

Rev.2

15.3.193 TillOn方法, Spel類別

描述

若till條件在最後Go/Jump/Move陳述式期間發生停止狀況, 會傳回True。

語法

Function TillOn () As Boolean 傳回值

急问追

若機器人因Till條件而停止,會傳回True,否則會傳回False。

備註

使用TillOn可檢查是否已在最後動作命令期間用Till開啟Till條件。 TillOn相當於 ((Stat(1) And 2) <> 0)

另請參閱

Jump, Till

TillOn範例

VB 例:

```
If m_spel.TillOn() Then
    m_spel.Jump(2)
End If
```

```
if (m_spel.TillOn())
    m_spel.Jump(2);
```

15.3.194 TGo方法 · Spel類別

描述

在目前工具座標系統中執行PTP相對動作。

語法

Sub TGo (PointNumber As Integer) Sub TGo (Point As SpelPoint) Sub TGo (Point As SpelPoint, AttribExpr As String) Sub TGo (PointExpr As String)

參數

每個語法都有一個參數,用以指定在TGo動作期間手臂移動的結束點。此為PTP動作結束時的最終位置。

- PointNumber 從目前機器人的點記憶體(被保存在控制器中)中使用先前示教點的編號,以指定終點。
 Point
- 透過使用SpelPoint資料類型來指定目標點。
- AttribExpr 透過使用字串運算式來指定結束點屬性。
 可包括CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。
- PointExpr 透過使用字串運算式來指定終點。
 可包括CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。

另請參閱

Accel, Speed Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move BGo, BMove, TMove Find, Till

TGo範例

VB 例:

```
    使用點編號指定的點
        m_spel.Tool(1)
        m_spel.TGo(100)

    使用SpelPoint指定的點
        Dim pt As SpelPoint
        pt = m_spel.GetPoint("P*")
        pt.X = 125.5
            m_spel.TGo(pt)

    使用點運算式指定的點
        m_spel.TGo("P0")
            m_spel.TGo("XY(0, 0, -20, 0)")

    使用平行處理
        m_spel.TGo("P1 \!D50; On 1; D90; Off 1\!")
        m_enemotic definition
        m_enemotic definition
        m_enemotic definition
        Dimensional definition
        m_enemotic definit
        <pm_enemotic definition</p>
        <pm_enemotic
```

m_spel.TGo("pick")

// 使用點編號指定的點 m_spel.Tool(1); m_spel.TGo(100);

// 使用SpelPoint指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.TGo(pt);

// 使用點運算式指定的點 m_spel.TGo("P0 /L /2"); m_spel.TGo("P1 :Z(-20)");

// 使用平行處理 m_spel.TGo("P1 \!D50; On 1; D90; Off 1\!");

// 使用標籤指定的點
m_spel.TGo("pick");

15.3.195 TLCIr方法, Spel類別

描述

清除(取消定義)工具座標系統。

語法

Sub TLClr (ToolNumber As Integer)

參數

 ToolNumber 整數,代表要清除哪個工具座標系(Tool 0為預設工具,無法清除)。

另請參閱

Tool, TLDef

TLCIr範例

VB 例:

m_spel. TLClr(1)

C# 例:

m_spel. TLClr(1);

15.3.196 TLDef方法, Spel類別

描述

傳回工具定義狀態。

語法

Function TLDef (ToolNumber As Integer) As Boolean

參數

 ToolNumber 代表要傳回狀態之工具的整數運算式。

傳回值

若指定工具已定義,會傳回True,否則會傳回False。

另請參閱

Tool, TLClr

TLDef範例

VB 例:

m_spel.TLDef(1)

C# 例:

m_spel.TLDef(1);

15.3.197 TLSet方法, Spel類別

描述

定義工具座標系統。

語法

Sub TLset (ToolNumber As Integer, Point As SpelPoint) Sub TLset (ToolNumber As Integer, XCoord As Single, YCoord As Single, ZCoord As Single, UCoord As Single, VCoord As Single)

參數

- ToolNumber 整數運算式(1~15),代表從15個工具中定義哪個工具。Tool 0為預設工具,無法變更。
- Point
 包含點資料的SpelPoint。
- XCoord 工具座標系統原點X座標。
- YCoord 工具座標系統原點Y座標。
- ZCoord 工具座標系統原點Z座標。
- UCoord 繞Z軸的工具座標系統旋轉。
- VCoord 繞Y軸的工具座標系統旋轉。
- WCoord 繞X軸的工具座標系統旋轉。

另請參閱

Arm, ArmSet, GetTool, Tool

TLSet範例

VB 例:

m_spel.TLSet(1, .5, 4.3, 0, 0, 0, 0)

C# 例:

m_spel.TLSet(1, .5, 4.3, 0, 0, 0, 0);

15.3.198 TMove方法, Spel類別

描述

在目前工具座標系統中執行線性插補相對動作。

語法

Sub TMove (PointNumber As Integer) Sub TMove (Point As SpelPoint) Sub TMove (Point As SpelPoint, AttribExpr As String) Sub TMove (PointExpr As String)

參數

每個語法都有一個參數,用以指定在TMove動作期間手臂移動的結束點。此為線性插補動作結束時的最終位置。

- PointNumber 從目前機器人的點記憶體(被保存在控制器中)中使用先前示教點的編號,以指定目標位置。
 Point
- 透過使用SpelPoint資料類型來指定目標點。
- AttribExpr 透過使用字串運算式來指定終點屬性。
 可包括ROT、CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。
- PointExpr 透過使用字串運算式來指定目標位置。
 可包括ROT、CP、SYNC、Till或Find的搜尋運算式以及並行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move BGo, BMove, TGo Find, Till

TMove範例

VB 例:

```
使用點編號指定的點
m_spel.Tool(1)
m_spel.TMove(100)
使用SpelPoint指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.TMove(pt)
使用點運算式指定的點
m_spel.TMove("P0")
m_spel.TMove("Y1(0, 0, -20, 0)")
使用平行處理
m_spel.TMove("P1 \!D50; On 1; D90; Off 1\!")
使用標籤指定的點
m_spel.TMove("pick")
```

// 使用點編號指定的點
m_spel.Tool(1);
m_spel.TMove(100);
// 使用SpelPoint指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.TMove(pt);
// 使用點運算式指定的點

m_spel.TMove ("PO");
m_spel.TMove ("XY(0, 0, -20, 0)");

```
// 使用平行處理
m_spel.TMove("P1 \!D50; On 1; D90; Off 1\!");
```

// 使用標籤指定的點 m_spel.TMove("pick");

15.3.199 Tool方法 · Spel類別

描述

選擇目前機器人工具。

語法

Sub Tool (ToolNumber As Integer)

參數

 ToolNumber 介於0-15的整數,代表16個工具定義中要搭配後續動作指令使用的工具定義。

另請參閱

TLSet, Arm, TGo, TMove

Tool範例

VB 例:

m_spel.Tool(1)
m_spel.TGo(100)

C# 例:

m_spel.Tool(1); m_spel.TGo(100);

15.3.200 TrapStop方法, Spel類別

描述

若目前機器人在上一個動作命令期間因trap而停止,會傳回True。

語法

Function TrapStop () As Boolean

傳回值

若機器人因trap而停止,會傳回True,否則會傳回False。

另請參閱

EstopOn, ErrorOn

TrapStop範例

VB 例:

```
If m_spel.TrapStop() Then
    MsgBox "Robot stopped by Trap"
End If
```

```
if (m_spel.TrapStop())
    MessageBox.Show("Robot stopped by trap");
```

15.3.201 TW方法, Spel類別

描述

傳回WAIT條件及WAIT定時器間隔的狀態。

語法

Function TW () As Boolean

傳回值

若發生超時,會傳回True,否則會傳回False。

另請參閱

WaitMem, WaitSw

TW範例

VB 例:

```
Const PartPresent = 1
m_spel.WaitSw(PartPresent, True, 5)
If m_spel.TW() Then
        MsgBox "Part present time out occurred"
End If
```

```
const int PartPresent = 1;
m_spel.WaitSw(PartPresent, True, 5);
if (m_spel.TW())
    MessageBox.Show("Part present time out occurred");
```

15.3.202 UserHasRight方法, Spel類別

描述

傳回目前登錄的使用者是否具有指定的權限。

語法

Function UserHasRight (SpelUserRights Right) As Boolean

參數

 Right 要針對目前登錄使用者進行檢查的權限。

傳回值

若使用者擁有指定權限,會傳回True,否則會傳回False。

另請參閱

Login, GetCurrentUser

UserHasRight範例

VB 例:

Dim hasRight As Boolean hasRight = m_spel.UserHasRight(SpelUserRights.EditPoints)

```
bool hasRight;
hasRight = m_spel.UserHasRight(SpelUserRights.EditPoints);
```

15.3.203 VCal方法 · Spel類別

描述

此命令可讓您執行視覺校準週期。

語法

Sub VCal (CalibName As String) Sub VCal (CalibName As String, ByRef Status As Integer) Sub VCal (CalibName As String, Parent As Form) Sub VCal (CalibName As String, Parent As Form, ByRef Status As Integer)

參數

- CalibName 評估目前專案中校正技術之名稱的字串運算式。
- Status 接收校正狀態的整數變數。
 0:失敗, 1:成功
- Parent 視窗的父.NET表單(選用)。

備註

執行VCal方法時,機器人將會移動。您應確認操作員是否已在執行VCal前完成準備。

VCal只會執行校準週期,不能讓您示教點。使用VCalPoints則可示教點。此外,您必須先在Epson RC+ 8.0中設定校準。如需詳細資訊,請參閱Vision Guide手冊。

使用Status參數檢查校準是否成功。如果ShowConfirmation校準屬性設為True,則會顯示確認對話方塊。當操作員按一下[OK]按鈕時,狀態傳回1:成功。

另請參閱

VCalPoints

VCal範例

VB 例:

```
Dim status As Integer
m_spel.VCal("CAMCAL1", status)
```

```
int status;
m spel.VCal("CAMCAL1", status);
```

15.3.204 VCalPoints方法, Spel類別

描述

此命令可讓您示教視覺校準點。

語法

Sub VCalPoints (CalibName As String) Sub VCalPoints (CalibName As String, ByRef Status As Integer) Sub VCalPoints (CalibName As String, Parent As Form) Sub VCalPoints (CalibName As String, Parent As Form, ByRef Status As Integer)

參數

- CalibName 字元運算式,代表目前專案的校正技術名稱。
- Status 選用。接收點示教狀態的整數變數。0:失敗,1:成功
- Parent
 視窗的父.NET表單(選用)。

備註

執行VCalPoints命令時,示教點校準點對話方塊會開啟。按一下[Finish]按鈕時,校準資料會自動保存。 您必須先從Epson RC+ 8.0建立校準設定。



使用Status參數檢查點示教是否成功。當所有點已示教並按一下[Finish]按鈕時, Status傳回1:成功。

另請參閱

VCal

VCalPoints範例

VB 例:

```
Dim status As Integer
m spel.VCalPoints("CAMCAL1", status)
```

```
int status;
m_spel.VCalPoints("CAMCAL1", out status);
```

15.3.205 VCIs方法 · Spel類別

描述

清除視覺圖形。

語法

Sub VCls ()

備註

使用VCls方法可清除視覺畫面。

另請參閱

VRun

VCIs範例

VB 例:

m_spel.VCls()

C# 例:

m_spel.VCls();

15.3.206 VCreateCalibration方法, Spel類別

描述

在目前專案中建立新建視覺校準。

語法

Sub VCreateCalibration (CameraNumber As Integer, CalibName As String) Sub VCreateCalibration (CameraNumber As Integer, CalibName As String,CopyCalibName As String)

參數

- CameraNumber
 包含要校正之攝影機編號的整數運算式。
- CalibName
 包含要建立之視覺校正名稱的字串運算式。
- CopyCalibName 字串運算式,代表要複製的視覺校準名稱(可省略)。

另請參閱

VcreateObject, VcreateSequence, VDeleteCalibration

VCreateCalibration範例

VB 例:

m_spel.VCreateCalibration(1, "mycal")

C# 例:

m_spel.VCreateCalibration(1, "mycal");

15.3.207 VCreateObject方法, Spel類別

描述

在目前專案中建立視覺物件。

語法

Sub VCreateObject (Sequence As String, ObjectName As String, ObjectType As SpelVisionObjectTypes)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- ObjectName
 包含要在Sequence序列中建立之物件名稱的字串運算式。
- ObjectType 指定視覺物件類型的SpelVisionObjectTypes。(以下所示的常數也可使用)

· · · · · · · · · · · · · · · · · · ·		1
物件類型	常數	數值
Correlation	Correlation	1
Blob	Blob	2
Edge	Edge	3
Polar	Polar	4
Line	Line	5
Point	Point	6
Frame	Frame	7
ImageOp	ImageOp	8
Ocr	Ocr	9
CodeReader	CodeReader	10
Geometric	Geometric	11
Color Match	ColorMatch	14
Line Finder	LineFinder	15
Arc Finder	ArcFinder	16
Defect Finder	DefectFinder	17
Line Inspector	LineInspector	18
Arc Inspector	ArcInspector	19
Box Finder	BoxFinder	20

物件類型	常數	數值
Corner Finder	CornerFinder	21
Contour	Contour	22
Text	Text	23
Decision	Decision	26
Coordinates	Coordinates	27

另請參閱

VcreateSequence, VdeleteObject, VDeleteSequence

VCreateObject範例

VB 例:

m_spel.VCreateObject("myseq", "myblob", SpelVisionObjectTypes.Blob)

C# 例:

m_spel.VCreateObject("myseq", "myblob", SpelVisionObjectTypes.Blob);

15.3.208 VCreateSequence方法 · Spel類別

描述

在目前專案中建立新建視覺序列。

語法

Sub VCreateSequence (CameraNumber As Integer, SequenceName As String) Sub VCreateSequence (CameraNumber As Integer, SequenceName As String, CopySequenceName As String)

參數

- CameraNumber
 包含要使用之攝影機編號的整數運算式。
- SequenceName
 包含要建立之視覺序列名稱的字串運算式。
- CopySequenceName 字串運算式,代表要複製的視覺校準名稱(可省略)。

另請參閱

VcreateObject, VdeleteObject, VDeleteSequence

VCreateSequence範例

VB 例:

m_spel.VCreateSequence(1, "myseq")

```
m_spel.VCreateSequence(1, "myseq");
```

15.3.209 VDefArm方法, Spel類別

描述

使用視覺系統可偵測到的特徵點,計算移動攝影機的手臂設定值。

注意:機器人會根據目標偵測結果自動運作。小心機器人與周邊設備發生干擾。此外,這可用於避開各軸延伸姿態附近的奇點,防止手臂設定期間發生錯誤。

語法

Sub VDefArm (ArmNumber As Integer, ArmDefType As SpelArmDefType, ArmDefMode As SpelArmDefMode, Sequence As String, Rotation As Double, TargetTolerance As Double)

Sub VDefArm (ArmNumber As Integer, ArmDefType As SpelArmDefType, ArmDefMode As SpelArmDefMode, Sequence As String, Rotation As Double, TargetTolerance As Double, Parent As Form)

Sub VDefArm (ArmNumber As Integer, ArmDefType As SpelArmDefType, ArmDefMode As SpelArmDefMode, Sequence As String, Rotation As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, ShowWarning As SpelVDefShowWarning)

Sub VDefArm (ArmNumber As Integer, ArmDefType As SpelArmDefType, ArmDefMode As SpelArmDefMode, Sequence As String, Rotation As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, ShowWarning As SpelVDefShowWarning,Parent As Form)

參數

- ArmNumber 整數運算式,包含用以執行手臂設定的手臂編號(1至15)。
- ArmDefType
 包含手臂類型的整數運算式。
 J2Camera:計算移動J2攝影機影像的中心。
- ArmDefMode

包含手臂設定模式的整數運算式。 執行rough手臂設定的模式。機器人將會以1公釐的設定準確度為目標進行移動。機器人動作會比較小。 Fine:執行fine手臂設定的模式。機器人將隨手臂方向改變大幅移動,提供更高準確度的手臂設定。

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Rotation rough手臂設定之旋轉角度(度)的實數運算式。 範圍介於0至45。
- TargetTolerance
 包含像素距離的實數運算式,將視覺偵測結果視為符合目標位置。
 範圍介於0至3。
- Parent 視窗的父.NET表單(選用)。
- RobotSpeed
 代表機器人速度(%)的整數變數(可省略)。數值範圍:0~100。若省略,將被設為「5」。
- RobotAccel
 代表機器人加速度(%)的整數變數(可省略)。數值範圍:0~99。若省略,將被設為「5」。
- ShowWarning 整數變數(可省略),用於設定是否在ArmSetMode為Fine時發出警告。Always:一定發出。DependsOnSpeed: RobotSpeed或RobotAccel大於5時發出。None:不發出。若省略,將被設為「DependsOnSpeed」。

另請參閱

VDefGetMotionRange, VDefLocal, VDefSetMotionRange, VDefTool, VGoCenter

VDefArm範例

VB 例:

m_spel.VDefArm(1, SpelArmDefType.J2Camera, SpelArmDefMode.Rough, "myseq", 5, 1)

```
m_spel.VDefArm(1, SpelArmDefType.J2Camera, SpelArmDefMode.Rough, "myseq", 5, 1);
```

15.3.210 VDefGetMotionRange方法, Spel類別

描述

取得VDefTool、VDefArm、VDefLocal及VGoCenter所限制之動作範圍的值。

語法

Sub VDefGetMotionRange(ByRef MaxMoveDist As Double, ByRef MaxPoseDiffAngle As Double, ByRef LjmMode As Integer)

參數

- MaxMoveDist 代表最大移動距離的實數變數。 如果指定0,角度不受限制。(0至500。預設:200) VDefTool、VDefArm、VDefLocal及VGoCenter用於限制範圍。
- MaxPoseDiffAngle 代表工具方向(UVW)最大位移角度(度)的實數變數。 如果指定0,範圍不受限制。 這只會影響VDefLocal。(0至180。預設:45度)
- LjmMode 代表LJM模式的整數變數。

另請參閱 VDefTool, VDefArm, VDefLocal, VGoCenter, VDefSetMotionRange

VDefGetMotionRange範例

VB 例:

```
Dim maxMoveDist As Double
Dim maxPoseDiffAngle As Double
Dim ljmMode As Integer
m_spel.VDefGetMotionRange(maxMoveDist, maxPoseDiffAngle, ljmMode)
```

```
double maxMoveDist, maxPoseDiffAngle;
int ljmMode;
m spel.VDefGetMotionRange(out maxMoveDist, out maxPoseDiffAngle, out ljmMode);
```
描述

透過移動攝影機偵測放在工作平面上的校準板,並定義與工作平面平行的本地座標。 這也會透過固定攝影機偵測工具末端的使用者工件,並定義與固定攝影機感測器平行的本地平面。 注意:機器人會根據目標偵測結果自動運作。小心機器人與周邊設備發生干擾。此外,這可用於避開各軸延伸姿態附 近的奇點,防止本地座標設定期間發生錯誤。

語法

Sub VDefLocal(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint) Sub VDefLocal(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint, Parent As Form) Sub VDefLocal(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint, Parent As Form) Sub VDefLocal(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint, RobotSpeed As Integer, RobotAccel As Integer)

Sub VDefLocal(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint, RobotSpeed As Integer, RobotAccel As Integer, Parent As Form)

參數

- LocalNumber
 代表用以設定本地座標之工具編號的整數。(1-15)
- LocalDefType 代表本地類型的整數。
- J5Camera:
 使用移動I5攝影機指定與校準板平行的本地座標。
- J6Camera:
 使用移動J6攝影機指定與校準板平行的本地座標。
- FixedUpwardCamera:
 使用上方固定攝影機指定與影像感測器平行的本地座標。
- FixedDownwardCamera:
 使用下方固定攝影機指定向下影像感測器平行的本地座標。
- CalPlateType 代表校正板類型的整數。
- Large: 大校準板
- Medium : 中校準板
- Small:
 小校準板
- XSmall: 超小校準板
- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
 使用移動攝影機時,此為拍攝校準板相片的視覺序列。
 使用固定攝影機時,此為偵測工具末端特徵點(例如使用者工件)的視覺序列。
- TargetTolerance
 代表刻度符合判定閾值的實際值。

- CameraTool 固定攝影機:指定保持偵測目標工具偏移的工具編號。若要執行自動校準,請指定-1。
 移動J6攝影機:如果已執行自動校準,請指定移動攝影機的工具編號。若要執行自動校準,請指定-1。
 移動J5攝影機:此選購件的設定會略過。
- RefPoint 平行於工作平面之本地平面通過的點編號。
 這個點用於指定本地平面高度。
- Parent
 視窗的父.NET表單(選用)。
- RobotSpeed
 代表機器人速度(%)的整數變數(可省略)。數值範圍:0~100。若省略,將被設為「5」。
- RobotAccel
 代表機器人加速度(%)的整數變數(可省略)。數值範圍:0~99。若省略,將被設為「5」。

另請參閱

VDefArm, VDefGetMotionRange, VDefSetMotionRange, VDefTool, VGoCenter

VDefLocal範例

VB 例:

```
Dim p2 = m_spel.GetPoint("P2")
m_spel.VDefLocal(1, SpelLocalDefType.J6Camera, SpelCalPlateType.Large, "myseq",
1.0, 1, p2)
```

```
SpelPoint p2;
p2 = m_spel.GetPoint("P2");
m_spel.VDefLocal(1, SpelLocalDefType.J6Camera,
SpelCalPlateType.Large, "myseq", 1.0, 1, p2);
```

描述

由VDefTool、VDefArm、VDefLocal及VGoCenter限制動作範圍。

語法

Sub VDefSetMotionRange(MaxMoveDist As Double, MaxPoseDiffAngle As Double, LjmMode As Integer)

參數

- MaxMoveDist 代表最大移動距離的實數 如果指定0,範圍不受限制。(0至500。預設:200) VDefTool、VDefArm、VDefLocal及VGoCenter用於限制範圍。
- MaxPoseDiffAngle 代表工具方向(UVW)最大位移角度(度)的實數 如果指定0,範圍不受限制。 這只會影響VDefLocal。(0至180。預設:45度)
- LjmMode 代表LJM模式的整數。

另請參閱 VDefTool, VDefArm, VDefLocal, VGoCenter, VDefGetMotionRange

VDefSetMotionRange範例

VB 例:

m_spel.VDefSetMotionRange(100, 30, 1)

C# 例:

m_spel.VDefSetMotionRange(100, 30, 1);

15.3.213 VDefTool方法 · Spel類別

描述

使用視覺偵測,計算TCP移動攝影機位置的工具偏移值。

注意:當工具為FixedCameraWithCal以外的類型時,機器人會根據目標偵測結果自動運作。小心機器人與周邊設備發生干擾。此外,這可用於避開各軸延伸姿態附近的奇點,防止工具設定期間發生錯誤。

語法

Sub VDefTool(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, Object As String) Sub VDefTool(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, Object As String, Parent As Form)

Sub VDefTool(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double)

Sub VDefTool(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, Parent As Form)

Sub VDefTool(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer) Sub VDefTool(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, Parent As Form)

參數

- ToolNumber 代表用以執行工具設定(1-15)之工具編號的整數
- ToolDefType

代表工具類型的整數。FixedCamera:透過使用未校正之固定攝影機,設定工具。J4Camera:計算移動J4攝影機的影像中心。J6Camera:計算移動J6攝影機的影像中心。FixedCameraWithCal:透過使用已校正之固定攝影機所設定的工具。

Sequence

包含目前專案中視覺序列之名稱的字串運算式。

Object

代表指定序列中視覺物件的字串運算式。當ToolDefType設為FixedCameraWithCal時,需要此參數。當ToolDefType不是FixedCameraWithCal時,物件應為空字串。

- FinalAngle 實數值,代表旋轉工具或攝影機工具的角度(度)。數值範圍:0、5~180、-5~-180。若省略,將被設為「90」。
- InitAngle

實數,代表設定暫時性工具時旋轉工具或攝影機工具的角度(度)。此值必須小於FinalAngle。數值範圍:-10~10。若省略,將被設為「5」。

TargetTolerance 代表像素距離的實際值,將視覺偵測結果視為符合目標位置。數值範圍:0至3像素

如果省略,請設為「1」。

- Parent 視窗的父.NET表單(選用)。
- RobotSpeed
 代表機器人速度(%)的整數變數(可省略)。數值範圍:0~100。若省略,將被設為「5」。
- RobotAccel
 代表機器人加速度(%)的整數變數(可省略)。數值範圍:0~99。若省略,將被設為「5」。

另請參閱

VDefArm, VDefGetMotionRange, VDefLocal, VDefSetMotionRange, VGoCenter

VDefTool範例

VB 例:

m_spel.VDefTool(1, SpelToolDefType.J6Camera, "myseq", 45, 5, 3.0)
m_spel.VDefTool(1, SpelToolDefType.FixedCameraWithCal, "myseq", "myobj")

```
m_spel.VDefTool(1, SpelToolDefType.J6Camera, "myseq", 45, 5, 3.0);
m_spel.VDefTool(1, SpelToolDefType.FixedCameraWithCal, "myseq", "myobj");
```

15.3.214 VDefToolXYZ 方法, Spel 類別

描述

VDefToolXYZ 使用視覺檢測來計算工具偏移值(XYZ)。

注意:

根據目標檢測結果,機器人會自動工作。注意機器人和週邊設備之間的干擾。此外,這可用於避開各軸延伸姿態附近 的奇點,防止工具設定期間發生錯誤。

語法

Sub VDefToolXYZ(ToolNumber As Integer, LocalNumber As Integer, PointNumber1 As Integer, PointNumber2 As Integer, Sequence1 As String, Sequence2 As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer)

Sub VDefToolXYZ(ToolNumber As Integer, LocalNumber As Integer, PointNumber1 As Integer, PointNumber2 As Integer, Sequence1 As String, Sequence2 As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, Parent As Form)

參數

- ToolNumber 代表用以執行工具設定(1-15)之工具編號的整數
- LocalNumber
 代表機器人移動位置之本地座標編號的整數。工具會移動到指定的本地座標的XY平面。
- PointNumber1
 代表Point Number第一姿勢的整數
- PointNumber2
 代表Point Number第二姿勢的整數
- Sequence1
 代表當前專案中第一姿勢的視覺序列名稱的字串運算式
- Sequence2
 代表當前專案中第二姿勢的視覺序列名稱的字串運算式
- FinalAngle
 代表工具與攝影機工具旋轉角度(度)的實數數值範圍:5~180,-5~-180
- InitAngle
 臨時工具設定時的旋轉角度(度)的實數該值必須小於FinalAngle
 數值範圍: 0.01 ~ 10, -0.01 ~ -10
- TargetTolerance
 代表視覺檢測結果與物件位置一致的閾值像素的實數數值範圍: 0.1 ~ 3.0 pixel
- RobotSpeed
 代表機器人速度(%)的整數變數,數值範圍:1~100
- RobotAccel
 代表機器人加速度(%)的整數變數,數值範圍:1~99
- Parent
 視窗的父.NET表單(選用)。

<mark>另請参閲</mark> VDefTool, VDefToolXYZUVW

VDefToolXYZ範例

VB 例:

m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar)

```
C# 例:
```

m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5, 5); m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5, 5); m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5, 5); m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar);

15.3.215 VDefToolXYZUVW 方法, Spel 類別

描述

用3个工具定义來計算工具UVW偏移值。 注意:計算出的U, V, W工具偏移值在指定的工具1中設置。工具1的X, Y, Z偏移不變。

語法

Sub VDefToolXYZUVW(ToolNumber1 As Integer, ToolNumber2 As Integer, ToolNumber3 As Integer, ToolDefType3D As SpelToolDefType3D)

參數

- ToolNumber1
 代表第1個工具定義的工具編號的整數變量(1~15)
 條形類型指定工具尖端的工具編號。平面類型指定工具中心的工具編號。
- ToolNumber2
 代表第2個工具定義的工具編號的整數變量(1~15)
 條形類型指定工具中央處的工具編號。平面類型指定工具中心以外,與ToolNumber3不同的工具編號。
- ToolNumber3
 代表第3個工具定義的工具編號的整數變量(1~15)。
 條形類型指定工具末端的工具編號。平面類型指定工具中心以外,與ToolNumber2不同的工具編號。
- ToolDefType3D
 代表工具定義的工具類型的整數變量Bar: 條形類型Plane: 平面類型

另請參閱

VDefToolXYZ

VDefToolXYZUVW範例

VB 例:

```
m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar)
```

```
m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5, 5);
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5, 5);
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5, 5);
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar);
```

15.3.216 VDeleteCalibration方法, Spel類別

描述

在目前專案中刪除視覺校準。

語法

Sub VDeleteCalibration (CalibName As String)

參數

CalibName
 包含目前專案中視覺校正之名稱的字串運算式。

另請參閱

VCreateCalibration, VDeleteObject, VDeleteSequence

VDeleteCalibration範例

VB 例:

m_spel.VDeleteCalibration("mycal")

C# 例:

m_spel.VDeleteCalibration("mycal");

15.3.217 VDeleteObject方法, Spel類別

描述

在目前專案中刪除視覺物件。

語法

Sub VDeleteObject (Sequence As String, ObjectName As String)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- ObjectName 字串運算式,代表目前專案的視覺物件名稱。

另請參閱

VcreateObject, VcreateSequence, VDeleteSequence

VDeleteObject範例

VB 例:

m_spel.VDeleteObject("myseq", "myobj")

C# 例:

m_spel.VDeleteObject("myseq", "myobj");

描述

在目前專案中刪除視覺序列。

語法

Sub VDeleteSequence (Sequence As String)

參數

Sequence
 包含目前專案中視覺序列之名稱的字串運算式。

另請參閱

VcreateObject, VcreateSequence, VDeleteObject

VDeleteSequence範例

VB 例:

m_spel.VDeleteSequence("myseq")

C# 例:

m_spel.VDeleteSequence("myseq");

15.3.219 VEditWindow 方法, Spel類別

描述

編輯搜尋視窗中的不要緊的圖元。更多詳細資訊,請參閱Vision Guide Properties and Results Reference手冊 的"EditWindow屬性"。

語法

Sub VEditWindow (Sequence As String, Object As String, Parent As Form)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- ObjectName 字串運算式,代表目前專案的視覺物件名稱。



另請參閱

VSave

VEditWindow範例

VB 例:

m_spel.VEditWindow("myseq", "myobj")

C# 例:

m_spel.VEditWindow("myseq", "myobj");

15.3.220 VGet方法, Spel類別

描述

取得視覺序列或物件屬性或結果的值。

語法

Sub VGet (Sequence As String, PropCode As SpelVisionProps, ByRef Value As Integer) Sub VGet (Sequence As String, PropCode As SpelVisionProps, ByRef Value As Boolean) Sub VGet (Sequence As String, PropCode As SpelVisionProps, ByRef Value As Double) Sub VGet (Sequence As String, PropCode As SpelVisionProps, ByRef Value As Single) Sub VGet (Sequence As String, PropCode As SpelVisionProps, ByRef Value As String) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Integer) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Boolean) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Boolean) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Color) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Color) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Double) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Single) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Single) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Single) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, ByRef Value As Single) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, Result As Integer, ByRef Value As Integer) Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, Result As Integer, ByRef Value As Integer)

Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, Result As Integer, ByRef Value As Boolean)

Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, Result As Integer, ByRef Value As Double)

Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, Result As Integer, ByRef Value As Single)

Sub VGet (Sequence As String, Object As String, PropCode As SpelVisionProps, Result As Integer, ByRef Value As String)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- PropCode 指定屬性代碼的SpelVisionProps值。
- Result 代表結果編號的整數運算式。
- Value
 代表屬性或結果值的字串。字串的類型必須符合屬性或結果類型。

另請參閱

VSet, VRun

VGet範例

VB 例:

```
Dim i As Integer
Redim score(10) As Integer
m_spel.VRun("testSeq")
For i = 1 to 10
    m_spel.VGet("testSeq", "corr" & Format$(i, "00"), _
```

```
SpelVisionProps.Score, score(i))
Next i
```

15.3.221 VGetCameraXYU方法, Spel類別

描述

擷取任何物件的攝影機X、Y及U實體座標。

語法

Sub VGetCameraXYU (Sequence As String, Object As String, Result As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Result 代表結果編號的整數運算式。
- Found
 包含是否找到物件的Boolean變數。
- X

包含x座標的實際變數(以公釐為單位)。

• Y

包含y座標的實際變數(以公釐為單位)。

U
 包含角度的實際變數(以度為單位)。

另請參閱

VGetPixelXYU, VGetRobotXYU

VGetCameraXYU範例

VB 例:

```
Dim found As Boolean
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String
seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGetCameraXYU(seq, blob, 1, found, x, y, u)
```

```
bool found;
float x, y, u;
string seq, blob;
seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m spel.VGetCameraXYU(seq, blob, 1, out found, out x, out y, out u);
```

15.3.222 VGetEdgeCameraXYU方法, Spel類別

描述

針對Line Finder、Arc Finder搜尋的每個邊緣, 擷取攝影機X、Y及U實體座標。

語法

Sub VGetEdgeCameraXYU (Sequence As String, Object As String, EdgeResultIndex As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- EdgeResultIndex
 代表邊緣結果索引的整數運算式。
- Found
 包含是否找到物件的Boolean變數。
- X

包含x座標的實際變數(以公釐為單位)。

• Y

包含y座標的實際變數(以公釐為單位)。

U
 包含角度的實際變數(以度為單位)。

另請參閱

VGetEdgePixelXYU, VGetEdgeRobotXYU, VGetPixelXYU, VGetRobotXYU

VGetEdgeCameraXYU範例

VB 例:

```
Dim found(10) As Boolean
Dim x(10) As Single, y(10) As Single, u(10) As Single
Dim seq As String, lineFinder As String
seq = "testSeq"
lineFinder = "LineFind01"
m_spel.VRun(seq)
' LineFinder的NumberOfEdges為10
For i = 1 To 10
m_spel.VGetEdgeCameraXYU(seq, lineFinder, i, found(i), x(i),
y(i), u(i))
Next i
```

```
bool[] found = new bool[11];
float[] x = new float[11];
float[] y = new float[11];
float[] u = new float[11];
string seq, lineFinder;
seq = "testSeq";
lineFinder = "LineFind01";
m_spel.VRun(seq);
// LineFinder的NumberOfEdges為10
```

15.3.223 VGetEdgePixelXYU方法, Spel類別

描述

針對Line Finder、Arc Finder搜尋的每個邊緣, 擷取X、Y及U像素座標。

語法

Sub VGetEdgePixelXYU (Sequence As String, Object As String, EdgeResultIndex As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- EdgeResultIndex
 代表邊緣結果索引的整數運算式。
- Found
 包含是否找到物件的Boolean變數。
- X

包含x座標的實際變數(以公釐為單位)。

• Y

包含y座標的實際變數(以公釐為單位)。

U
 包含角度的實際變數(以度為單位)。

另請參閱

VGetEdgeCameraXYU, VGetEdgeRobotXYU, VGetPixelXYU, VGetRobotXYU

VGetEdgePixelXYU範例

VB 例:

```
Dim found(10) As Boolean
Dim x(10) As Single, y(10) As Single, u(10) As Single
Dim seq As String, lineFinder As String
seq = "testSeq"
lineFinder = "LineFind01"
m_spel.VRun(seq)
' LineFinder的NumberOfEdges為10
For i = 1 To 10
m_spel.VGetEdgePixelXYU(seq, lineFinder, i, found(i), x(i),
    y(i), u(i))
Next i
```

```
bool[] found = new bool[11];
float[] x = new float[11];
float[] y = new float[11];
float[] u = new float[11];
string seq, lineFinder;
seq = "testSeq";
lineFinder = "LineFind01";
m_spel.VRun(seq);
// LineFinder的NumberOfEdges為10
```

15.3.224 VGetEdgeRobotXYU方法, Spel類別

描述

針對Line Finder、Arc Finder搜尋的每個邊緣, 擷取機器人X、Y及U實體座標。

語法

Sub VGetEdgeRobotXYU (Sequence As String, Object As String, EdgeResultIndex As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- EdgeResultIndex
 代表邊緣結果索引的整數運算式。
- Found
 包含是否找到物件的Boolean變數。
- X

包含x座標的實際變數(以公釐為單位)。

• Y

包含y座標的實際變數(以公釐為單位)。

U
 包含角度的實際變數(以度為單位)。

另請參閱

VGetEdgeCameraXYU, VGetEdgePixelXYU, VGetPixelXYU, VGetRobotXYU

VGetEdgeRobotXYU範例

VB 例:

```
Dim found(10) As Boolean
Dim x(10) As Single, y(10) As Single, u(10) As Single
Dim seq As String, lineFinder As String
seq = "testSeq"
lineFinder = "LineFind01"
m_spel.VRun(seq)
' LineFinder的NumberOfEdges為10
For i = 1 To 10
m_spel.VGetEdgeRobotXYU(seq, lineFinder, i, found(i), x(i),
    y(i), u(i))
Next i
```

```
bool[] found = new bool[11];
float[] x = new float[11];
float[] y = new float[11];
float[] u = new float[11];
string seq, lineFinder;
seq = "testSeq";
lineFinder = "LineFind01";
m_spel.VRun(seq);
// LineFinder的NumberOfEdges為10
```

15.3.225 VGetExtrema方法, Spel類別

描述

語法

Sub VGetExtrema (Sequence As String, Object As String, Result As Integer, ByRef MinX As Single, ByRef MaxX As Single, ByRef MinY As Single, ByRef MaxY As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Result
 代表結果編號的整數運算式。
- MinX
 包含最小x座標的實際變數(以像素為單位)。
- MaxX
 包含最大x座標的實際變數(以像素為單位)。
- MinY
 包含最小y座標的實際變數(以像素為單位)。
- MaxY
 包含最大y座標的實際變數(以像素為單位)。

另請參閱

VGet

VGetExtrema範例

VB 例:

```
float xMin, xMax, yMin, yMax;
bool found;
string seq, blob;
seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m_spel.VGet(seq, blob, "found", found);
if(found == true)
```

Rev.2

15.3.226 VGetModelWin方法, Spel類別

描述

擷取物件的模型視窗座標。

語法

Sub VGetModelWin (Sequence As String, Object As String, ByRef Left As Integer, ByRef Top As Integer, ByRef Width As Integer, ByRef Height As Integer)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Left
 包含左側座標的整數變數(以像素為單位)。
- Top
 包含頂端座標的整數變數(以像素為單位)。
- Width
 包含寬度的整數變數(以像素為單位)。
- Height 包含高度的整數變數(以像素為單位)。

另請參閱

VsetModelWin, VgetSearchWin, VSetSearchWin

VGetModelWin範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer
With m_spel
   .VGetModelWin("testSeq", "corr01", left, top, _
      width, height)
   .VSetModelWin("testSeq", "corr01", left + 20, top, _
      width, height)
   .VTeach("testSeq", "corr01")
End With
```

15.3.227 VGetPixelToCamera 方法, Spel 類別

描述

擷取指定圖元座標的攝影機座標。

語法

Sub VGetPixelToCamera (Calibration As String, PixelX As Single, PixelY As Single, Angle As Single, ByRef CameraX As Single, ByRef CameraY As Single, ByRef CameraU As Single)

參數

- Calibration
 代表目前專案中校正名稱的字串運算式
- PixelX
 包含y座標的實際變數(以像素為單位)。
- PixelY
 包含y座標的實際變數(以像素為單位)。
- Angle
 包含角度的實際變數(以度為單位)。
- CameraX
 包含x座標的實際變數(以公釐為單位)。
- CameraY
 包含y座標的實際變數(以公釐為單位)。
- CameraU
 包含角度的實際變數(以度為單位)。

另請參閱

VGetPixelXYU, VGetCameraXYU, VGetPixelToRobot

VGetPixelToCamera範例

VB 例:

```
Dim x As Single, y As Single, u As Single
Dim cal As String, seq As String
cal = "testCal"
seq = "testSeq"
m_spel.VRun(seq)
m_spel.VGetPixelToCamera(cal, 400, 300, 0, x, y, u)
```

```
float x, y, u
string cal, seq;

cal = "testCal";
seq = "testSeq";
m_spel.VRun(seq);
m_spel.VGetPixelToCamera(cal, 400, 300, 0, out x, out y, out u);
```

15.3.228 VGetPixelToRobot 方法, Spel 類別

描述

擷取指定圖元座標的機器人座標。

語法

Sub VGetPixelToRobot (Calibration As String, PixelX As Single, PixelY As Single, Angle As Single, ByRef RobotX As Single, ByRef RobotU As Single)

參數

- Calibration
 代表目前專案中校正名稱的字串運算式
- PixelX
 包含y座標的實際變數(以像素為單位)。
- PixelY
 包含y座標的實際變數(以像素為單位)。
- Angle
 包含角度的實際變數(以度為單位)。
- RobotX
 包含x座標的實際變數(以公釐為單位)。
- RobotY
 包含y座標的實際變數(以公釐為單位)。
- RobotU
 包含角度的實際變數(以度為單位)。

另請參閱

VGetPixelXYU, VGetRobotXYU, VGetPixelToCamera

VGetPixelToRobot範例

VB 例:

```
Dim x As Single, y As Single, u As Single
Dim cal As String, seq As String
cal = "testCal"
seq = "testSeq"
m_spel.VRun(seq)
m_spel.VGetPixelToRobot(cal, 400, 300, 0, x, y, u)
```

```
float x, y, u
string cal, seq;
cal = "testCal";
seq = "testSeq";
m_spel.VRun(seq);
m_spel.VGetPixelToRobot(cal, 400, 300, 0, out x, out y, out u);
```

15.3.229 VGetPixelXYU方法, Spel類別

描述

擷取任何物件的像素X、Y及U座標。

語法

Sub VGetPixelXYU (Sequence As String, Object As String, Result As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Result
 代表結果編號的整數運算式。
- Found
 包含是否找到物件的Boolean變數。
- X

包含x座標的實際變數(以像素為單位)。

• Y

包含y座標的實際變數(以像素為單位)。

U
 包含角度的實際變數(以度為單位)。

另請參閱

VGetCameraXYU, VGetRobotXYU

VGetPixelXYU範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String
seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGetPixelXYU(seq, blob, 1, found, x, y, u)
```

```
int found;
float x, y, u;
string seq, blob;
seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m spel.VGetPixelXYU(seq, blob,1, out found, out x, out y, out u);
```

15.3.230 VGetRobotPlacePos方法, Spel類別

描述

擷取機器人放置位置。

語法

Sub VGetRobotPlacePos (Sequence As String, Object As String, Result As Integer, ByRef Found As Boolean, ByRef PlacePointAs SpelPoint)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Result
 代表結果編號的整數運算式。
- Found
 代表布林類型偵測狀態的整數變數。若檢測為False,則未定義W、Y及U。
- PlacePoint
 包含放置位置的SpelPoint變數

另請參閱

VGetRobotPlaceTargetPos, VSetRobotPlaceTargetPos

VGetRobotPlacePos範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String
Dim placePoint As SpelPoint
seq = "testSeq"
blob = "blob01"
' 移動高於上方攝影機的工件
m_spel.Jump("camPos")
m_spel.VRun(seq)
m_spel.VGetRobotPlacePos(seq, blob, 1, found, placePoint)
' 使用SCARA以使用+TLZ接近
m_spel.Jump(placePoint, "+TLZ(10)")
m_spel.Go(placePoint)
```

```
bool found;
float x, y, u;
string seq, blob;
SpelPoint placePoint = new SpelPoint();
seq = "testSeq";
blob = "blob01";
// 移動高於上方攝影機的工件
m_spel.Jump("camPos");
m_spel.VRun(seq);
m spel.VGetRobotPlacePos(seq,blob,1,out found, out placePoint);
```

```
// 使用scara以使用+TLZ接近
m_spel.Jump(placePoint, "+TLZ(10)");
m_spel.Go(placePoint);
```

15.3.231 VGetRobotPlaceTargetPos方法, Spel類別

描述

擷取工件放置位置。

語法

Sub VGetRobotPlaceTargetPos (Sequence As String, Object As String, ByRef Point As SpelPoint)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Point
 包含放置位置的SpelPoint變數

另請參閱

VGetRobotPlacePos, VSetRobotPlaceTargetPos

VGetRobotPlaceTargetPos範例

VB 例:

```
Dim seq As String, blob As String
Dim targetPoint As SpelPoint
seq = "testSeq"
blob = "blob01"
m_spel.VGetRobotPlaceTargetPos(seq, blob, targetPoint)
```

```
' Adjust the place position
targetPoint.X = targetPoint.X + 10
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint)
```

```
Spe
lPoint targetPoint = new SpelPoint();
seq = "testSeq";
blob = "blob01";
m_spel.VGetRobotPlaceTargetPos(seq, blob, out targetPoint);
```

```
// 調整放置位置
targetPoint.X = targetPoint.X + 10;
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint);
```

15.3.232 VGetRobotXYU方法, Spel類別

描述

擷取任何物件的機器人預設X、Y及U座標。

語法

Sub VGetRobotXYU (Sequence As String, Object As String, Result As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Result
 代表結果編號的整數運算式。
- Found
 代表布林類型偵測狀態的整數變數。若檢測為False,則未定義W、Y及U。
- X 代表x座標的實數變數(以公釐為單位)。
- Y

代表y座標的實數變數(以公釐為單位)。

U
 包含角度的實際變數(以度為單位)。

另請參閱

VGetCameraXYU, VGetPixelXYU

VGetRobotXYU範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String
seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGetRobotXYU(seq, blob, 1, found, x, y, u)
```

```
bool found;
float x, y, u;
string seq, blob;
seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m spel.VGetRobotXYU(seq, blob, 1, out found, out x, out y, out u);
```

15.3.233 VGetRobotToolXYU方法, Spel類別

描述

擷取工具定義的機器人預設X、Y及U數值。

語法

Sub VGetRobotToolXYU (Sequence As String, Object As String, Result As Integer, ByRef Found As Boolean, ByRef X As Single, ByRef Y As Single, ByRef U As Single)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Result
 代表結果編號的整數運算式。
- Found
 代表布林類型偵測狀態的整數變數。若檢測為False,則未定義W、Y及U。
- X 代表x座標的實數變數(以公釐為單位)。
- Y

代表y座標的實數變數(以公釐為單位)。

U
 包含角度的實際變數(以度為單位)。

備註

使用VGetRobotToolXYU可針對透過上方攝影機檢視的工件輕鬆定義工具。這可讓您拾取工件、在上方攝影機FOV 中進行搜尋、定義工件的工具,然後放置工件。

另請參閱

VGetCameraXYU, VGetPixelXYU, VGetRobotXYU

VGetRobotToolXYU範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String
seq = "testSeq"
```

```
blob = "blob01"
' 移動高於上方攝影機的工件
m_spel.Jump("camPos")
m_spel.VRun(seq)
m_spel.VGetRobotToolXYU(seq, blob, 1, found, x, y, u)
m_spel.TLSet(1, x, y, u)
```

C# 例:

bool fnd; float x, y, u; string seq, blob; seq = "testSeq"; Rev.2

Rev.2

15.3.234 VGetSearchWIn方法, Spel類別

描述

擷取搜尋視窗座標。

語法

Sub VGetSearchWin (Sequence As String, Object As String, ByRef Left As Integer, ByRef Top As Integer, ByRef Width As Integer, ByRef Height As Integer)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Left
 包含左側座標的整數變數(以像素為單位)。
- Top
 包含頂端座標的整數變數(以像素為單位)。
- Width
 包含寬度的整數變數(以像素為單位)。
- Height 包含高度的整數變數(以像素為單位)。

另請參閱

VgetModelWin, VsetModelWin, VSetSearchWin

VGetSearchWin範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer
With m_spel
   .VGetSearchWin("testSeq", "corr01", left, top, _
      width, height)
   .VSetSearchWin("testSeq", "corr01", newLeft, top, _
      width, height)
   .VRun("testSeq")
End With
```

15.3.235 VGoCenter方法 · Spel類別

描述

使用可由視覺系統偵測的特徵點時,將機器人移至特徵點位於攝影機影像中央的位置。

語法

Sub VGoCenter(Sequence As String, LocalNumber As Integer, TargetTolerance As Double)

Sub VGoCenter(Sequence As String, LocalNumber As Integer, TargetTolerance As Double, Parent As Form) Sub VGoCenter(Sequence As String, LocalNumber As Integer, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer)

Sub VGoCenter(Sequence As String, LocalNumber As Integer, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, Parent As Form)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- LocalNumber
 代表機器人移動位置之本地座標編號的整數。
 如果指定-1,機器人會在工具旋轉的XY平面中移動。
- TargetTolerance
 代表像素距離的實數,將視覺偵測結果視為符合目標位置。
 數值範圍:0~3像素
- Form 視窗的父.NET表單(選用)。
- RobotSpeed
 代表機器人速度(%)的整數變數(可省略)。數值範圍:0~100。若省略,將被設為「5」。
- RobotAccel
 代表機器人加速度(%)的整數變數(可省略)。數值範圍:0~99。若省略,將被設為「5」。

另請參閱

VDefArm, VDefGetMotionRange, VDefLocal, VDefSetMotionRange, VDefTool

VGoCenter範例

VB 例:

m_spel.VGoCenter("myseq", 1, 1.0)

C# 例:

m_spel.VGoCenter("myseq", 1, 1.0);

15.3.236 VLoad方法 · Spel類別

描述

從目前專案載入視覺屬性。

語法

Sub VLoad ()

備註

要在程式啟動期間將視覺屬性設定、模型及字型恢復為原始設定時,請使用VLoad方法。

另請參閱

VSave

VLoad範例

VB 例:

m_spel.VLoad()

C# 例:

m_spel.VLoad();
15.3.237 VLoadModel方法, Spel類別

描述

從驅動盤檔案載入視覺模型。

語法

Sub VLoadModel (Sequence As String, Object As String, Path As String)

參數

- Sequence
 包含目前專案中序列之名稱的字串。
- Object 代表物件名稱的字串。物件請使用Correlation、Geometric或Polar。
- Path
 要載入模型之檔案的完整路徑名稱,不包括副檔名。

備註

若檔案的模型資料為錯誤的類型,將會發生錯誤。例如,如果嘗試將polar模型載入correlation,將會發生錯誤。如果 提供副檔名,則會予以忽略。有兩個檔案與fileName相關聯。 對於correlation和geometric模型,ModelOrgX和ModelOrgY值會與模型視窗寬度和高度一起恢復。 對於polar模型,則會恢復Radius、Thickness及AngleOffset。

另請參閱

VSaveModel

VLoadModel範例

VB 例:

```
m_spel.VLoadModel("seq01", "corr01", "d:\models\part1")
```

```
m_spel.VLoadModel("seq01", "corr01", @"d:\models\part1");
```

Rev.2

15.3.238 VRun方法, Spel類別

描述

在目前專案中執行視覺序列。

語法

Sub VRun (Sequence As String)

參數

Sequence
 包含目前專案中序列之名稱的字串。

備註

VRun支援任何攝影機校準或無校準類型的序列。

若要顯示圖形,您必須使用SPELVideo控制項,並將Spel類別執行個體的SpelVideoControl屬性設為SPELVideo控制項。

執行VRun後,請使用VGet擷取結果。

另請參閱

Vget, VSet

VRun範例

VB 例:

```
Function FindPart(x As Single, y As Single, angle As Single)As Boolean
Dim found As Boolean
Dim x, y, angle As Single
With m_spel
.VRun("seq01")
.VGet("seq01", "corr01", "found", found)
If found Then
.VGet("seq01", "corr01", "cameraX", x)
.VGet("seq01", "corr01", "cameraY", y)
.VGet("seq01", "corr01", "angle", angle)
FindPart = True
End If
End With
End Function
```

```
bool FindPart(float x, float y, float angle)
{
    bool found;
    m_spel.VRun("seq01");
    m_spel.VGet("seq01", "corr01", "found", found);
    if (found) {
        m_spel.VGet("seq01", "corr01", "cameraX", out x);
        m_spel.VGet("seq01", "corr01", "cameraY", out y);
        m_spel.VGet("seq01", "corr01", "angle", out angle);
    }
    return found;
}
```

15.3.239 VSave方法, Spel類別

描述

在目前專案中保存所有視覺資料。

語法

Sub VSave ()

備註

使用VSave可對視覺屬性作任何永久改變。

另請參閱

Vset

VSaveImage範例

VB 例:

```
With m_spel
    .VSet("seq01", "blob01", "SearchWinLeft", 100)
    .VSet("seq01", "corr01", "Accept", userAccept)
    .VSave()
End With
```

```
m_spel.VSet("seq01", "blob01", "SearchWinLeft", 100);
m_spel.VSet("seq01", "corr01", "Accept", userAccept);
m_spel.VSave();
```

15.3.240 VSaveImage方法, Spel類別

描述

將視覺視訊視窗保存至PC驅動盤檔案。

語法

Sub VSaveImage (Sequence As String, Path As String) Sub VSaveImage (Sequence As String, Path As String, WithGraphics As Boolean)

參數

- Sequence
 包含目前專案中序列之名稱的字串。
- Path
 要保存影像之檔案的完整路徑名稱,包括副檔名。
- WithGraphics 設定是否要在影像檔中保存序列結果圖形的Boolean運算式。

備註

使用VSaveImage可將視訊顯示器上的影像保存至驅動盤。副檔名: .bmp .tif .jpg **另請參閱**

LoadImage(SPELVideo控制項)

VSaveImage範例

VB 例:

```
Dim found As Boolean
m_spel.VRun("Seq")
m_spel.VGet("Seq", SpelVisionProps.AllFound, found)
If Not found Then
    m_spel.VSaveImage("Seq", "d:\reject.mim")
End If
```

```
bool found;
m_spel.VRun("Seq");
m_spel.VGet("Seq", SpelVisionProps.AllFound, out found);
if (\!found)
    m spel.VSaveImage("Seq", @"d:\reject.mim");
```

Rev.2

15.3.241 VSaveModel方法, Spel類別

描述

將視覺物件搜尋模型保存至PC驅動盤檔案。

語法

Sub VSaveModel (Sequence As String, Object As String, Path As String)

參數

Sequence
 包含目前專案中序列之名稱的字串。

- Object 代表物件名稱的字串。物件請使用Correlation、Geometric或Polar。
- Path
 要保存模型之檔案的完整路徑名稱,不包括副檔名。

備註

執行VSaveModel時, Epson RC+ 8.0會建立兩個檔案(Path + 副檔名): Path.VOB, Path.MDL 對於correlation和geometric模型, ModelOrgX和ModelOrgY值會與模型視窗一起保存。對於Polar模型, 則會保存 Radius、Thickness及AngleOffset。

另請參閱

VLoadModel

VSaveModel範例

VB 例:

```
m_spel.VSaveModel("seq01", "corr01", "d:\models\part1")
```

```
m spel.VSaveModel("seq01", "corr01", @"d:\models\part1");
```

15.3.242 VSet方法, Spel類別

描述

設定視覺序列或物件屬性的值。

語法

Sub VSet (Sequence As String, PropCode As SpelVisionProps, Value As Integer) Sub VSet (Sequence As String, PropCode As SpelVisionProps, Value As Boolean) Sub VSet (Sequence As String, PropCode As SpelVisionProps, Value As Double) Sub VSet (Sequence As String, PropCode As SpelVisionProps, Value As Single) Sub VSet (Sequence As String, PropCode As SpelVisionProps, Value As String) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Integer) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Integer) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Boolean) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Color) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Color) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Double) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As String) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As String) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As Single) Sub VSet (Sequence As String, Object As String, PropCode As SpelVisionProps, Value As String)

參數

Sequence 包含目前專案中視覺序列之名稱的字串運算式。

- Object
 包含Sequence序列中物件之名稱的字串運算式。
- propName
 包含要為其設定值的屬性之名稱的字串運算式。
- PropCode 指定屬性代碼的SpelVisionProps值。
- value
 包含新建值的運算式。運算式類型必須符合屬性類型。

另請參閱

Vget, VRun

VSet範例

VB 例:

```
m_spel.VSet("seq01", "corr01", SpelVisionProps.Accept, 250)
```

C# 例:

m spel.VSet("seq01", "corr01", SpelVisionProps.Accept, 250);

15.3.243 VSetModelWin方法, Spel類別

描述

設定模型視窗座標。

語法

Sub VSetModelWin (Sequence As String, Object As String, Left As Integer, Top As Integer, Width As Integer, Height As Integer)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Left 代表左側座標的整數運算式(以像素為單位)。
- Top
 代表頂端座標的整數運算式(以像素為單位)。
- Width 代表寬度的整數運算式(以像素為單位)。
- Height 代表高度的整數運算式(以像素為單位)。

另請參閱

VgetModelWin, VgetSearchWin, VSetSearchWin

VSetModelWin範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer
With m_spel
   .VGetModelWin("testSeq", "corr01", left, top, _
    width, height)
   .VSetModelWin("testSeq", "corr01", left + 20, top, _
    width, height)
   .VTeach("testSeq", "corr01")
End With
```

15.3.244 VSetRobotPlaceTargetPos方法, Spel類別

描述

設定工件放置位置。

語法

Sub VSetRobotPlaceTargetPos (Sequence As String, Object As String, Point As SpelPoint)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Point
 包含放置位置的SpelPoint變數

另請參閱

VGetRobotPlacePos, VGetRobotPlaceTargetPos

VSetRobotPlaceTargetPos Example

VB 例:

```
Dim seq As String, blob As String
Dim targetPoint As SpelPoint
```

```
seq = "testSeq"
blob = "blob01"
m_spel.VGetRobotPlaceTargetPos(seq, blob, targetPoint)
```

• 調整放置位置

```
targetPoint.X = targetPoint.X + 10
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint)
```

```
string seq, blob;
SpelPoint targetPoint = new SpelPoint();
seq = "testSeq";
blob = "blob01";
m_spel.VGetRobotPlaceTargetPos(seq, blob, out targetPoint);
// 調整放置位置
```

```
targetPoint.X = targetPoint.X + 10;
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint);
```

15.3.245 VSetSearchWin方法, Spel類別

描述

設定搜尋視窗座標。

語法

Sub VSetSearchWin (Sequence As String, Object As String, Left As Integer, Top As Integer, Width As Integer, Height As Integer)

參數

- Sequence
 包含目前專案中視覺序列之名稱的字串運算式。
- Object
 包含Sequence序列中物件之名稱的字串運算式。
- Left 代表左側座標的整數運算式(以像素為單位)。
- Top
 代表頂端座標的整數運算式(以像素為單位)。
- Width 代表寬度的整數運算式(以像素為單位)。
- Height 代表高度的整數運算式(以像素為單位)。

另請參閱

VgetModelWin, VsetModelWin, VGetSearchWin

VSetSearchWin範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer
With m_spel
.VGetSearchWin("testSeq", "corr01", left, top, _____
width, height)
.VSetSearchWin("testSeq", "corr01", newLeft, top, _____
width, height)
.VRun("testSeq")
End With
```

```
int left, top, width, height;
m_spel.VGetSearchWin("testSeq", "corr01", out left, out top,
    out width, out height);
m_spel. .VSetSearchWin("testSeq", "corr01", left + 50,
    top, width, height);
m spel.VRun("testSeq");
```

15.3.246 VShowModel方法, Spel類別

描述

顯示物件模型。如需詳細資訊,請參閱Vision Guide屬性參考中的ShowModel屬性。

語法

Sub VShowModel (Sequence As String, Object As String, Parent As Form)

參數

Sequence
 包含目前專案中視覺序列之名稱的字串運算式。

ObjectName
 代表當前專案的視覺物件名稱的字串運算式

💽 占 🔳 🗢 🔗 🐹 🗗 🚥 🔳 Zoom: 100%	~ .				
			✓ B Sequences		
		Sequence test Step 1 © Correlation Corrol	v (g) test ≅g: 1: Corr01 v (g) Calibrations gi≩ test		
			Step 1: Corr01		
			Property	Value	
			CurrentResult	1	•
			Description		
			Enabled	True	•
			Graphics	All	•
			> ModelOrg		
			> ModelWin		
			Name	Corr01	
	0		NumberToFind	1	•
			> SearchWin		
			ShowModel	Click to show ->	
			Result	Value	
			Angle		
			Found		
			NumberFound		
			Passed		
			PixelX		
			PixelY		
			RobotX		
			ShowModel Shows the model for the o	urrent object for viewing and edit	ing.
ОК	Cancel				

另請參閱

VshowSequence, VTrain

VShowModel範例

VB 例:

```
m_spel.VShowModel("myseq", "myobj")
```

C# 例:

m_spel.VShowModel("myseq", "myobj");

15.3.247 VShowSequence方法, Spel類別

描述

顯示序列中的所有物件。

語法

Sub VShowSequence (Sequence As String)

參數

Sequence
 包含要建立之視覺序列名稱的字串運算式。

備註

使用VShowSequence可在不執行序列的情況下顯示序列中的物件。使用中物件的顏色(洋紅色)會用於所有物件,以便輕易查看。

其中一個用途是,當機器人攝影機移至某工件的特定部分上時,會利用多個序列進行掃描。機器人處於定位後,可調用VShowSequence以顯示序列。

另請參閱

VShowModel

VShowSequence範例

VB 例:

m_spel.VShowSequence("myseq")

C# 例:

m_spel.VShowSequence("myseq");

15.3.248 VStatsReset方法, Spel類別

描述

重置目前專案中指定序列的視覺統計。

語法

Sub VStatsReset (Sequence As String)

參數

Sequence
 包含目前專案中視覺序列之名稱的字串運算式。

備註

VStatsReset只會針對目前Epson RC+ 8.0 會話, 重置記憶體中指定序列的統計。若要永久改變, 您應執行 VStatsSave。否則, 若您重啟Epson RC+ 8.0, 統計就會從驅動盤恢復。

另請參閱

VstatsResetAll, VstatsShow, VStatsSave

VStatsReset範例

VB 例:

```
Sub btnResetStats_Click()
    m_spel.VStatsReset("seq01")
End Sub
```

```
void btnResetStats_Click(object sender, EventArgs e)
{
m spel.VStatsReset("seq01");}
```

15.3.249 VStatsResetAll方法, Spel類別

描述

重置所有序列的視覺統計。

語法

Sub VStatsResetAll

備註

VStatsResetAll只會針對目前Epson RC+ 8.0會話,重置記憶體中的統計。若要永久改變,您應執行VStatsSave。

另請參閱

VstatsReset, VstatsShow, VStatsSave

VStatsResetAll範例

VB 例:

```
Sub btnResetStats_Click()
    m_spel.VStatsResetAll()
End Sub
```

```
void btnResetStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsResetAll();
}
```

15.3.250 VStatsSave方法 · Spel類別

描述

保存目前專案中所有序列的視覺統計。

語法

Sub VStatsSave ()

備註

若要保存對視覺統計所作的改變,必須在關閉Epson RC+ 8.0之前執行VStatsSave。

另請參閱

VstatsReset, VstatsResetAll, VStatsShow

VStatsSave範例

VB 例:

```
Sub btnResetStats_Click()
    m_spel.VStatsSave()
End Sub
```

```
void btnResetStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsSave();
}
```

15.3.251 VStatsShow方法 · Spel類別

描述

顯示目前專案中指定序列的視覺統計對話方塊。

語法

Sub VStatsShow (Sequence As String, Parent As Form)

參數

Sequence

包含目前專案中視覺序列之名稱的字串運算式。

Statistics - [test]					?	×	
Object: Results Index:	Geom01 1	~]	Found] :0 of 0		
Result	Units	Mean	StdDev	Range	Min	Max	
PixelX	pixel						
PixelY	pixel						
Angle	deg						
CameraX	mm						- 11
CameraY	mm						
							- 11
RobotX	pixel						
RobotY	pixel						- 11
RobotU	deg						
							- 11
Time	ms						
Score							
	_	_	_	_	_		

另請參閱

VstatsReset, VstatsResetAll, VStatsSave

VStatsShow範例

VB 例:

```
Sub btnShowStats_Click()
    m_spel.VStatsShow("seq01")
End Sub
```

```
void btnShowStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsShow("seq01");
}
```

Rev.2

15.3.252 VTeach方法 · Spel類別

描述

示教correlation、geometric或polar模型。

語法

Sub VTeach (Sequence As String, Object As String, ByRef Status as Integer) Sub VTeach (Sequence As String, Object As String, AddSample as Boolean, KeepDontCares As Boolean, ByRef Status as Integer)

參數

- Sequence
 包含目前專案中之視覺序列的名稱。
- Object 物件名稱。您可示教Correlation物件、Geometric物件或Polar物件。
- AddSample
 新增範例時為True,新增新模型時為False。
- KeepDontCares 使用舊偵測遮罩時為True,處置時為False。

 Status 傳回狀態。成功會傳回1,否則會傳回0。

備註

在調用VTeach之前,您必須確定模型視窗位於正確位置。

對於polar物件, 搜尋視窗和thickness必須正確設定。使用VSet搜尋視窗位置及thickness。

對於correlation和geometric物件, 搜尋視窗及模型視窗必須正確設定。使用SearchWin和ModelWin的VSet來設定搜 尋及模型視窗位置。或者, 您也可以使用VTrain命令, 讓操作員以互動方式改變視窗。

示教模型之後,您可以使用VSaveModel方法將其保存至PC驅動盤檔案。

另請參閱

Vtrain, VSaveModel

VTeach範例

VB 例:

Dim status As Integer

```
' 先讓操作員改變視窗位置m_spel.VTrain("seq01", "corr01", status)
```

```
現在示教模型m spel.VTeach("seq01", "corr01", status)
```

C# 例:

int status;

// 先讓操作員改變視窗位置
m_spel.VTrain("seq01", "corr01", status);

// 現在示教模型
m_spel.VTeach("seq01", "corr01", out status);

15.3.253 VTrain方法 · Spel類別

描述

此命令可讓您示教整個序列中的物件, 或示教個別物件。

語法

Function VTrain (Sequence As String [, Object As String] [, Flags as Integer] [, Parent as Form]) As Boolean

參數

Sequence
 包含目前專案中之視覺序列的名稱。

- Object
 Sequence的物件名稱。您可註冊任何物件類型。若Object是空白字串,則可示教整個序列。
- Flags
 組成[VTrain]對話方塊(可省略)。1 顯示[示教]按鈕。2 不顯示模型視窗。
- Parent 視窗的父.NET表單(選用)。

傳回值

若操作員按一下OK按鈕, VTrain會傳回True, 否則會傳回False。

備註

執行VTrain時,會開啟顯示即時視訊的對話方塊,並顯示指定的物件。操作員可調整搜尋視窗大小或移動該視窗, 並示教模型視窗(correlation和geometric物件適用)。操作員完成操作時,可按一下OK以保存改變,或按一下取消以 忽略改變。若按一下OK,新建資訊會自動保存至目前專案。

若flags位元1已設定,將會顯示示教按鈕。對於Correlation、Geometric及Polar物件,若按一下示教按鈕,將會示教 模型。您可在執行VTrain之後擷取ModelOK屬性,以確認模型是否完成示教。對於Blob物件,該按鈕將會開啟條形 圖,且操作員可調整高低閾值,然後查看改變的結果。

若flags位元2已設定,將不會顯示模型視窗。操作員僅能改變搜尋視窗。

對於correlation和geometric物件,如果不顯示示教按鈕,您可在調用VTrain之後調用VTeach以示教模型。

Train Corr01 \times 1: Corr01 OK Cancel

另請參閱 VTeach, VSaveModel

VTrain範例

VB 例:

```
Dim status As Integer
Dim trainOK As Boolean
' 先讓操作員改變視窗位置
trainOK = m_spel.VTrain("seq01", "corr01")
' 現在示教模型
If trainOK Then
        m_spel.VTeach("seq01", "corr01", status)
EndIf
```

```
C# 例:
```

int status; bool trainOK; // 先讓操作員改變視窗位置 trainOK = m_spel.VTrain("seq01", "corr01");

- // 現在示教模型
- if (trainOK)
 - m_spel.VTeach("seq01", "corr01", out status);

15.3.254 WaitCommandComplete方法, Spel類別

描述

此命令會等待以AsyncMode = True開頭的命令完成。

語法

Sub WaitCommandComplete ()

另請參閱

A syncMode

WaitCommandComplete範例

VB 例:

```
With m_spel
    .AsyncMode = True
    .Jump("pick")
    .Delay(500)
    .On(1)
    .WaitCommandComplete()
End With
```

```
m_spel.AsyncMode = true;
m_spel.Jump("pick");
m_spel.Delay(500);
m_spel.On(1);
m_spel.WaitCommandComplete();
```

15.3.255 WaitMem方法, Spel類別

描述

等待記憶體位元狀態改變。

語法

Sub WaitMem (BitNumber As Integer, Condition As Boolean, Timeout As Single) Sub WaitMem (Label As String, Condition As Boolean, Timeout As Single)

參數

- BitNumber 代表記憶體位元編號的整數運算式。
- Label 代表記憶體位元標籤的字串。
- Condition 代表記憶體位元狀態的Boolean運算式。
- Timeout
 代表最長等待時間的單一運算式(以秒為單位)。

備註

請務必使用TW方法檢查是否發生超時。請參閱以下範例。

另請參閱

WaitSw

WaitMem範例

VB 例:

C# 例:

// 等待記憶體位元1變成1 (True)
// 最長時間為5秒
m_spel.WaitMem(1, True, 5);
// WaitMem是否超時?
if (m_spel.TW())
 MessageBox.Show("memory bit time out occurred");

15.3.256 WaitSw方法 · Spel類別

描述

等待輸入位元狀態改變。

語法

Sub WaitSw (BitNumber As Integer, Condition As Boolean, Timeoutl As Single) Sub WaitSw (Label As String, Condition As Boolean, Timeoutl As Single)

參數

- BitNumber 代表輸入位元編號的整數運算式。
- Label 代表輸入位元標籤的字串。
- Condition 代表輸入位元狀態的Boolean運算式。
- Timeout
 代表最長等待時間的單一運算式(以秒為單位)。

備註

請務必使用TW方法檢查是否發生超時。請參閱以下範例。

另請參閱

WaitMem

WaitSw範例

VB 例:

```
Const PartPresent = 1
m_spel.WaitSw(PartPresent, True, 5)
If m_spel.TW() Then
        MsgBox "Part present time out occurred"
End If
```

```
const int PartPresent = 1;
m_spel.WaitSw(PartPresent, True, 5);
if (m_spel.TW())
    MessageBox.Show("Part Present time out occurred");
```

15.3.257 WaitTaskDone方法, Spel類別

描述

等待任務完成並傳回狀態。

語法

Function WaitTaskDone (TaskNumber As Integer) As SpelTaskState Function WaitTaskDone (TaskName As String) As SpelTaskState

參數

- TaskNumber 傳回執行狀態的任務#。
- TaskName
 包含任務名稱的字串運算式。

傳回值

SpelTaskState值。

另請參閱

SpelTaskState, TasksExecuting, TaskState, Xqt

WaitTaskDone範例

VB 例:

```
Dim taskState As SpelTaskState
m_spel.Xqt 2, "mytask"
'
' 其他處理
'
taskState = m_spel.WaitTaskDone(2)
```

```
SpelTaskState taskState;
m_spel.Xqt(2, "mytask");
//
// 其他處理
//
taskState = m_spel.WaitTaskDone(2);
```

描述

指定目前機器人的重量參數。

語法

Sub Weight (PayloadWeight As Single, ArmLength As Single, [Axis])

參數

- PayloadWeight
 待運送之夾具末端的重量(公斤)。
- ArmLength 從第二手臂旋轉中心至夾具末端重心的距離(公釐)。
- Axis 指定指派負載重量的附加軸(S或T)。

\Lambda 注意

請勿將整數值輸入至PayLoadWeight和ArmLength參數。使用單一變數或直接輸入單一類型值。

另請參閱

Inertia, JRange, Tool

Weight範例

VB 例:

m_spel.Weight(2.0F, 2.5F)

C# 例:

m_spel.Weight(2.0F, 2.5F);

15.3.259 Xqt方法, Spel類別

描述

啟動一個SPEL+任務。

語法

Sub Xqt (FuncName As String [, TaskType As SpelTaskType]) Sub Xqt (TaskNumber As Integer, FuncName As String [, TaskType As SpelTaskType])

參數

- TaskNumber 要執行之任務的任務#。任務#的範圍介於1至32。
- FuncName 要執行之函數的名稱。您也可以為函數提供引數。引數必須加上括號,並以逗號分隔。如需詳細資訊,請參閱 SPEL+ Xqt陳述式。另請參閱範例。
- TaskType 將任務類型指定為Normal、NoPause或NoEmgAbort(可省略)。

備註

執行Xqt時,控制將會立即回到調用程式。請使用Call方法等待任務完成,或者,亦可使用含任務狀態事件的 EventReceived等待任務結束。

另請參閱

Call, EnableEvent, EventReceived

Xqt範例

VB 例:

```
m spel.Xqt(2, "conveyor")
```

```
' 提供引數給RunPart函數
m_spel.Xqt(3, "RunPart(3)")
```

```
Dim funcToExec As String
funcToExec = "RunPart(" & partNum & ")"
m_spel.Xqt(3, funcCall)
```

```
m_spel.Xqt(2, "conveyor");
```

```
// 提供引數給RunPart函數
m spel.Xqt(3, "RunPart(3)");
```

```
string funcToExec;
funcToExec = string.Format("RunPart({0})", partNum);
m spel.Xqt(3, funcToExec);
```

15.3.260 XYLim方法 · Spel類別

描述

設定機器人的允許動作範圍限制。

語法

Sub XYLim (XLowerLimit As Single, XUpperLimit As Single, YLowerLimit As Single, YUpperLimit As Single [, ZLowerLimit As Single] [, ZUpperLimit As Single])

參數

- XLowerLimit 機器人可移動的最小X座標位置。(機器人無法移至小於minX的X座標位置。)
- XUpperLimit 機器人可移動的最大X座標位置。(機器人無法移至大於maxX的X座標位置。)
- YLowerLimit 機器人可移動的最小Y座標位置。(機器人無法移至小於min Y的Y座標位置。)
- YUpperLimit 機器人可移動的最大Y座標位置。(機器人無法移至大於max Y的Y座標位置。)
- ZLowerLimit 機器人可移動的最小Z座標位置。(機器人無法移至小於min Z的Z座標位置。)(可省略)
- ZUpperLimit 機器人可移動的最大Z座標位置。(機器人無法移至大於max Z的Z座標位置。)(可省略)

備註

XYLim係用來定義動作範圍限制。許多機器人系統都可讓使用者定義關節限制,而SPEL+語言則可允許同時定義關節限制與動作範圍限制。此可讓使用者有效地為其應用建立工作空間。 使用XYLim值建立的動作範圍只會套用至動作命令目標位置,不會套用到起始位置至目標位置的動作路徑。因此, 手臂可能會在動作期間超出XYLim範圍以外。(亦即XYLim範圍不會影響脈衝。) 若要關閉動作範圍限制,請將範圍限制參數指定為0。

另請參閱

JRange

XYLim範例

VB 例:

m_spel.XYLim(0, 0, 0, 0)

C# 例:

m_spel.XYLim(0, 0, 0, 0);

15.3.261 XYLimClr方法 · Spel類別

描述

清除(取消定義)XYLim定義。

語法

Sub XYLimClr ()

另請參閱

XYLim, XYLimDef

XYLimClr範例

VB 例:

m_spel.XYLimClr()

C# 例:

m_spel.XYLimClr();

15.3.262 XYLimDef方法, Spel類別

描述

傳回XYLim是否定義。

語法

Function XYLimDef () As Boolean

傳回值

若XYLim已定義, 會傳回True, 否則會傳回False。

另請参閱 XYLim, XYLimClr

XYLimDef範例

VB 例:

Dim xyLimDefined As Boolean
xyLimDefined = m_spel.XYLimDef()

C# 例:

```
bool xyLimDefined;
xyLimDefined = m_spel.XYLimDef();
```

15.4 Spel類別事件

15.4.1 EventReceived事件 · Spel類別

描述

發生於Epson RC+ 8.0傳送系統事件時;或發生於使用SPELCom_Event陳述式,以SPEL+執行中的程式傳送事件時。

語法

EventReceived (ByVal sender As Object, ByVal e As RCAPINet.SpelEventArgs)

參數

- e.Event
 代表特用使用者定義事件的編號。
- e.Message
 包含事件訊息的字串。

備註

Epson RC+ 8.0會發出多種系統事件。下表提供詳細說明。

系統事件

依據預設,部分事件為停用狀態。若要使用這些事件,您必須先使用EnableEvent方法將他們啟用。

事件編 號	事件訊息	常數	描述
1	"PAUSE"	SpelEvents.Pause	於任務暫停時發生。 預設為啟用。

事件編 號	事件訊息	常數	描述
2	"SAFE GUARD OPEN"	SpelEvents.SafeGuardOpen	於安全防護打開時發生。 預設為啟用。
3	"SAFE GUARD CLOSE"	SpelEvents.SafeGuardClose	於安全防護關閉時發生。 預設為啟用。
4	Project build status text	SpelEvents.ProjectBuildStatus	各建置狀態訊息會在BuildProject方法期間傳送。 CRLF會視需要添加。 這些訊息與在Epson RC+ 8.0 GUI的Project Build Status視窗上顯示的訊息相同。 此事件必須以EnableEvent方法啟用。 預設為停用。
5	"Error xxx!: mmm in task at line yyy"	SpelEvents.Error	於因未處理的錯誤或產生系統錯誤而導致任 務終止時發生。 預設為啟用。
6	Text from print statement	SpelEvents.Print	於一Print陳述式從SPEL+任務執行時發生。 預設為停用。
7	"ESTOP ON"	SpelEvents.EStopOn	於緊急停止情況變為ON時發生。 預設為啟用。
8	"ESTOP OFF"	SpelEvents.EStopOff	於緊急停止情況變為OFF時發生。 預設為啟用。
9	"CONT"	SpelEvents.Continue	於執行Cont後發生。 預設為啟用。
10	<robot #="">,"MOTOR ON"</robot>	SpelEvents.MotorOn	於所指示機器人的馬達變為ON時發生。 預設為停用。
11	<robot #="">,"MOTOR OFF"</robot>	SpelEvents.MotorOff	於所指示機器人的馬達變為OFF時發生。 預設為停用。
12	<robot #="">,"POWER HIGH"</robot>	SpelEvents.PowerHigh	於所指示機器人的運行功率變為HIGH時發 生。 預設為停用。
13	<robot #="">,"POWER LOW"</robot>	SpelEvents.PowerLow	於所指示機器人的運行功率變為LOW時發 生。 預設為停用。
14	"TEACH MODE"	SpelEvents.TeachMode	於示教模式啟動時發生。 預設為啟用。
15	"AUTO MODE"	SpelEvents.AutoMode	於自動模式啟動時發生。 預設為啟用。

事件編 號	事件訊息	常數	描述
16	" <taskid>,<status>, <funcname>" Status: "RUN", "HALT", "PAUSE", "FINISHED", "ABORTED"</funcname></status></taskid>	SpelEvents.TaskState	於任務狀態改變時發生。 預設為無效。
17	"SHUTDOWN"	SpelEvents.Shutdown	於RC+正在關閉時發生。 預設為無效。
18	"ALL TASKS STOPPED"	SpelEvents.AllTasksStopped	於所有任務已停止時發生。 預設為無效。
19	"DISCONNECTED"	SpelEvents.Disconnected	當控制器通信與PC中斷時發生。 啟用時,RC+不會顯示表示連線中斷的訊息 方塊。 預設為無效。
20	"MOTION Started"	SpelEvents.MotionStarted	控制命令已經開始。 如果啟用,將在執行以下Spel類方法時顯示。 Arc、Arc3、BGo、BMove、CVMove、Go、 Home、JTran、Jump、Jump3、Jump3CP、 MCal、Move、Pass、PTran、Pulse、TGo、 TMove 預設為無效。
21	"MOTION COMPLETE"	SpelEvents.MotionComplete	控制命令已經結束。 如果啟用,將在執行以下Spel類方法時顯示。 Arc、Arc3、BGo、BMove、CVMove、Go、 Home、JTran、Jump、Jump3、Jump3CP、 MCal、Move、Pass、PTran、Pulse、TGo、 TMove 預設為無效。

使用者事件

您可透過SPEL+程式,使用「SPELCom_Event」命令,將事件傳送至Visual Basic應用程式。

Spelcom_Event 3000, cycNum, lotNum, cycTime

此陳述式執行時, EventReceived常式將會連同事件編號及訊息一起調用。請參閱Epson RC+ 8.0線上說明或SPEL+ 語言參考, 瞭解「SPELCom_Event」的詳細資訊。

使用範例

VB 例:

```
Sub m_spel_EventReceived (
        ByVal sender As Object,
        ByVal e As RCAPINet.SpelEventArgs)
        Handles m_spel.EventReceived
    Redim tokens(0) As String
    Select Case e.Event
        Case 3000
```

C# 例:

```
public void m_spel_EventReceived(object sender, SpelEventArgs e)
{
    string[] tokens = new string[3];
    switch ((int) e.Event) {
        case 3000:
            tokens = e.Message.Split(' ');
            lblCycCount.Text = tokens(0);
            lblLotNumber.Text = tokens(1);
            lblLotVumber.Text = tokens(2);
            break;
        default:
            break;
    }
}
```

處理事件

當EventReceived從Spel類別執行個體調用時, Epson RC+ 8.0處理伺服器將需等待事件處理常式結束。因此, 您切 勿嘗試在EventReceived常式期間執行任何RC+ API命令或執行長執行時間的處理作業。若要根據發生的事件執行命 令,請在EventReceived中設定旗標,並從事件處理函數範圍外的應用程式main迴圈處理旗標。

例如,在Load程序的Visual Basic主要表單中,您可從SPEL+建立接收事件的事件迴圈。在「spel_EventReceived」 常式中,設定「全域旗標」以指示已接收的事件後,即可從「Load」程序所建立的事件迴圈中,執行實際的事件處 理。

顯示事件訊息

將TextBox控制項添加至表單。每次收到事件時,您可以在TextBox控制項的Text屬性中顯示事件訊息。

VB 例:

```
Private Sub m_spel_EventReceived(ByVal sender As Object,
    ByVal e As SpelEventArgs) Handles m_spel.EventReceived
    txtEvents.AppendText(e.Event & ": " & e.Message & vbCrLf)
End Sub
```

C# 例:

```
private void m_spel_EventReceived(object sender, SpelEventArgs e)
{
    txtEvents.AppendText(e.Event + ": " + e.Message);
}
```

另請參閱 EnableEvent(Spel類別)

15.5 SPELVideo控制項

描述

此控制項可讓您顯示視覺系統的視訊。如需此控制項使用方式的資訊,請參閱以下內容。

顯示視訊 檔案名稱

RCAPINt2.dll

15.6 SPELVideo控制項屬性

除了標準.NET元件屬性外(例如Left、Top、Width及Height),此控制項亦支援下列屬性。如需標準屬性的說明文件,請參閱Visual Basic線上說明。

- Camera
- GraphicsEnabled
- VideoEnabled

15.6.1 Camera屬性, SPELVideo控制項

描述

設定/取得要顯示視訊的攝影機編號。當您想在步進操作、即時視訊監控等情況下顯示視訊時,這會相當實用。若您 目前使用控制項來顯示視覺序列的圖形,則序列執行時,將會使用該序列的攝影機編號來取代此屬性值。

語法

Property Camera As Integer

預設值

0-顯示任何攝影機

傳回值

包含目前攝影機編號的整數值

另請參閱

VideoEnabled, GraphicsEnabled

Camera範例

VB 例:

SpelVideo1.Camera = 1

C# 例:

SpelVideo1.Camera = 1;

15.6.2 GraphicsEnabled屬性, SPELVideo控制項

描述

設定/傳回是否要在執行序列後顯示視覺圖形。為能看見圖形,您必須使用SPELVideo控制項屬性將控制項附加至 Spel類別執行個體。此屬性可「即時」設定,能在執行序列時開啟/關閉圖形。

語法

Property GraphicsEnabled As Boolean

預設值

False

傳回值

若顯示視覺圖形,會傳回True,否則會傳回False。

另請參閱 Camera, VideoEnabled

GraphicsEnabled範例 VB例:

SpelVideo1.GraphicsEnabled = True

C# 例:

SpelVideo1.GraphicsEnabled = true;

15.6.3 VideoEnabled屬性, SPELVideo控制項

描述

決定是否要顯示視訊。

語法

Property VideoEnabled As Boolean

預設值

False

傳回值

若顯示視訊,會傳回True,否則會傳回False。

另請參閱

Camera, GraphicsEnabled

VideoEnabled範例

VB 例:

SpelVideo1.VideoEnabled = True

C# 例:

SpelVideo1.VideoEnabled = true;

15.7 SPELVideo控制項方法

15.7.1 LoadImage方法, SPELVideo控制項

描述

從檔案載入要顯示的影像。

語法

Sub LoadImage (Path As String)

參數

 Path 要載入影像之檔案的完整路徑名稱。

備註

使用LoadImage可載入並顯示先前保存的影像。副檔名必須為BMP、TIF或JPG。

另請參閱

VSaveImage(Spel類別)

LoadImage範例

VB 例:

m_spelVideo.LoadImage("c:\RejectImages\reject001.bmp")

C# 例:

m_spelVideo.LoadImage(@"c:\RejectImages\reject001.bmp");
15.8 SPELVideo控制項事件

此控制項的所有事件都是標準.NET事件。如需詳細資訊,請參閱Visual Basic線上說明。

15.9 SpelConnectionInfo類別

成員名稱	類型	描述
ConnectionIPAddress	String	在Epson RC+中設定的IP位址。 *若以USB或虛擬控制器連接, ConnectionIPAddress為空白。
ConnectionNumber	Integer	在Epson RC+中設定的連線數量。
ConnectionName	String	在Epson RC+中設定的連線名稱。
ConnectionType	SpelConnectionType	在Epson RC+中設定的連線類型。

以下是範例。 VB 例:

Dim connectionInfo() As SpelConnectionInfo
connectionInfo = m_spel.GetConnectionInfo()

C# 例:

SpelConnectionInfo[] info = m_spel.GetConnectionInfo();

成員名稱	類型	描述
ProjectName	String	控制器中專案的名稱。
ProjectID	String	控制器中專案的唯一專案ID。
Options	List[SpelOptionInfo]	控制器中選配件的清單。

以下是範例。

VB 例:

```
Dim info As RCAPINet.SpelControllerInfo
info = m_spel.GetControllerInfo()
Label1.Text = info.ProjectID + " " + info.ProjectName
```

C# 例:

```
SpelControllerInfo info;
info = m_spel.GetControllerInfo();
Label1.text = info.ProjectID + " " + info.ProjectName;
```

15.11 SpelException類別

SpelException類別係衍生自ApplicationException類別。它會添加一ErrorNumber屬性及一些建構函式。 以下範例顯示如何擷取錯誤編號及錯誤訊息。

VB 例:

```
Try
  m_spel.Go(1)
Catch (ex As RCAPINet.SpelException)
  MsgBox(ex.ErrorNumber & ": " & ex.Message)
End Try
```

C# 例:

```
try{
   m_spel.Go(1);
}
catch(RCAPINet.SpelException ex){
   MessageBox.Show(string.Format("{0}: {1}", ex.ErrorNumber, ex.Message));
}
```

15.11.1 SpelException屬性

ErrorNumber As Integer

15.11.2 SpelException方法

Sub New ()

預設的建構函式。

Sub New (Message As String)

指定錯誤訊息的選用建構函式。

Sub New (ErrorNumber As Integer, Message As String)

指定錯誤編號及相關訊息的選用建構函式。

Sub New (Message As String, Inner As Exception)

指定錯誤訊息及內部例外的選用建構函式。

Sub New (ErrorNumber As Integer, Message As String, Inner As Exception)

指定錯誤編號、錯誤訊息及內部例外的選用建構函式。

15.12 SpelOptionInfo類別

成員名稱	類型	描述
Name	String	選件名稱
Status	SpelOptionStatus	選件設置狀態

以下是範例。

VB 例:

```
Dim info As SpelControllerInfo
info = m_spel.GetControllerInfo()
Label1.Text = info.Options(1).Name + " " + info.Options(1).Status.ToString()
```

C# 例:

```
SpelControllerInfo info;
info = m_spel.GetControllerInfo();
Label1.Text = info.Options[1].Name + " " + info.Options[1].Status;
```

15.13 SpelPoint類別

SpelPoint類別可用於多種動作方法,以及Spel類別的GetPoint與SetPoint方法。 Visual Basic範例如下:

1:

```
Dim pt As New RCAPINet.SpelPoint(25.5, 100.3, -21, 0)
m_spel.Go(pt)
```

2:

```
Dim pt As New RCAPINet.SpelPoint
pt.X = 25.5
pt.Y = 100.3
pt.Z = -21
m spel.Go(pt)
```

3:

```
Dim pt As New RCAPINet.SpelPoint
pt = m_spel.GetPoint("P*")
pt.Y = 222
m_spel.Go(pt)
```

Visual C# 範例如下:

1:

```
SpelPoint pt = new SpelPoint(25.5, 100.3, -21, 0);
m_spel.Go(pt);
```

2:

```
SpelPoint pt = new SpelPoint();
pt.X = 25.5;
pt.Y = 100.3;
pt.Z = -21;
m spel.Go(pt);
```

3:

```
SpelPoint pt = new SpelPoint();
pt = m_spel.GetPoint("P0");
pt.Y = 222;
m spel.Go(pt);
```

VB 例: X As Single Y As Single Z As Single U As Single V As Single W As Single R As Single S As Single T As Single Hand As SpelHand Elbow As SpelElbow Wrist As SpelWrist Local As Integer J1Flag As Integer J2Flag As Integer J4Flag As Integer J6Flag As Integer J1Angle As Single J4Angle As Single C# 例: float x float y float z float u float v float w float r float s float t SpelHand Hand SpelElbow Elbow SpelWrist Wrist int Local int J1Flag int J2Flag int J4Flag

int J6Flag float J1Angle float J4Angle

15.13.2 SpelPoint方法

Sub Clear ()

清除所有點資料。

Sub New ()

預設的建構函式。建立空的點(所有資料會清除)。

Sub New (X As Single, Y As Single, Z As Single, U As Single [, V As Single] [, W As Single]) 指定座標之新建點的選用建構函式。

Function ToString([Format As String]) As String

允許指定Format的ToString覆寫。此會傳回在SPEL+中所定義的點。

參數

 Empty: 傳回整個點(含所有座標及屬性)。
 "XY":傳回"XY(...)"
 "XYST":傳回"XY(...):ST(...)"

15.14 SpelRobotInfo類別

成員名稱	類型	描述
RobotModel	String	機器人型號
RobotName	String	機器人名稱
RobotNumber	Integer	機器人編號
RobotSerial	String	機器人序列號
RobotType	SpelRobotType	機器人類型

以下是範例。

VB 例:

```
Dim info As SpelRobotInfo
info = m_spel.GetRobotInfo()
Label1.Text = info.RobotNumber + " " + info.RobotModel
```

C# 例:

```
SpelRobotInfo info;
info = m_spel.GetRobotInfo();
Label1.Text = info.RobotNumber + " " + info.RobotModel;
```

15.15 SpelTaskInfo類別

成員名稱	類型	描述
CPU	Integer	SPEL任務的CPU負載率
ErrorCode	Integer	錯誤代碼
StartTime	DateTime	任務開始日期
TaskName	String	任務名稱
TaskNumber	Integer	任務編號
State	SpelTaskState	任務狀態

以下是範例。

VB 例:

```
Dim info As SpelTaskInfo
info = m_spel.GetTaskInfo(1)
Label1.Text = info.TaskNumber.ToString() + " " + info.State.ToString()
```

C# 例:

```
SpelTaskInfo info;
info = m_spel.GetTaskInfo(1);
Label1.Text = info.TaskNumber + " " + info.State;
```

15.16 列舉

15.16.1 SpelArmDefMode列舉

成員名稱	數值	描述
Rough :	1	使用一個姿態定義手臂。
Fine :	1	使用兩個姿態定義手臂。

15.16.2 SpelArmDefType列舉

成員名稱	數值	描述
J2Camera	1	定義J2安裝攝影機的手臂。

15.16.3 SpelAxis列舉

成員名稱	數值	描述
Х	1	X軸
Y	2	Y軸
Ζ	3	Z軸
U	4	U軸
V	5	V軸
W	6	W軸
R	7	R軸
S	8	S軸
Т	9	T軸

15.16.4 SpelBaseAlignment列舉

成員名稱	數值	描述
XAxis	0	對齊X軸。
YAxis	1	對齊Y軸。

15.16.5 SpelCalPlateType列舉

成員名稱	數值	描述
None	0	無校準板。
Large	1	大校準板。
Medium	2	中校準板。
Small	3	小校準板。
XSmall	4	超小校準板。

15.16.6 SpelConnectionType列舉

成員名稱	數值	描述
USB	1	USB連線。
Ethernet	2	乙太網連線。
Virtual	3	與虛擬控制器的連線。

15.16.7 SpelDialogs列舉

成員名稱	數值	描述
RobotManager	1	工具 機器人管理器對話方塊的ID
ControllerTools	2	工具 控制器對話方塊的ID
VisionGuide	3	工具 Vision Guide對話方塊的ID
ForceGuide	4	Force Guide對話方塊的ID
PartFeeding	5	Part Feeding對話方塊的ID

15.16.8 SpelElbow列舉

成員名稱	數值	描述
Above	1	肘部方向為上方。
Below	2	肘部方向為下方。

15.16.9 SpelEvents列舉

成員名稱	數值	描述
Pause	1	Pause事件的ID
SafeguardOpen	2	安全防護打開事件的ID
SafeguardClose	3	安全防護關閉事件的ID
ProjectBuildStatus	4	專案建置狀態事件的ID
Error	5	錯誤事件的ID
Print	6	列印事件的ID
EstopOn	7	緊急停止on事件的ID
EstopOff	8	緊急停止off事件的ID
Continue	9	Continue事件的ID
MotorOn	10	馬達on事件的ID
MotorOff	11	馬達off事件的ID
PowerHigh	12	運行功率high事件的ID
PowerLow	13	運行功率low事件的ID
TeachMode	14	示教模式事件的ID
AutoMode	15	自動模式事件的ID
TaskState	16	任務狀態事件的ID
Shutdown	17	Shutdown事件的ID
AllTasksStopped	18	所有任務停止事件的ID
Disconnected	19	斷開事件的ID
MotionStarted	20	控制命令開始事件的ID
MotionComplete	21	控制命令停止事件的ID

15.16.10 SpelForceCompareType列舉

成員名稱	數值	描述	
LessOrEqual	0	當力小於或等於指定閾值時,會觸	發Till。
GreaterOrEqual	1	當力大於或等於指定閾值時,會觸	發Till。

成員名稱	數值	描述	
EndStatus	28340	力覺引導序列或力覺引導物件的結束狀態	
ForceCondOK	28440	力覺引導物件的力的結束條件的達成狀態	
IOCondOK	28590	力覺引導物件的I/O結束條件的達成狀態	
PosCondOK	28860	力覺引導物件的位置結束條件的達成狀態	
Time	29070	力覺引導序列或力覺引導物件的執行時間	
TimedOut	29080	力覺引導物件超時的達成狀態	
LastExecObject	30100	最後被執行的力覺引導物件的名稱	
MeasuredHeight	30500	高度檢查序列測量出的高度	
FailedStatus	30510	力覺引導序列的失敗原因	
AvgForcesFx	100211	力覺引導物件執行時,Fx力的平均值	
AvgForcesFy	100212	力覺引導物件執行時,Fy力的平均值	
AvgForcesFz	100213	力覺引導物件執行時,Fz力的平均值	
AvgForcesTx	100214	力覺引導物件執行時,Tx轉矩的平均值	
AvgForcesTy	100215	力覺引導物件執行時, Ty轉矩的平均值	
AvgForcesTz	100216	力覺引導物件執行時,Tz轉矩的平均值	
EndForcesFx	102411	力覺引導序列或力覺引導物件結束時Fx的力	
EndForcesFy	102412	力覺引導序列或力覺引導物件結束時Fy的力	
EndForcesFz	102413	力覺引導序列或力覺引導物件結束時Fz的力	
EndForcesTx	102414	力覺引導序列或力覺引導物件結束時Tx的轉矩	
EndForcesTy	102415	力覺引導序列或力覺引導物件結束時Ty的轉矩	
EndForcesTz	102416	力覺引導序列或力覺引導物件結束時Tz的轉矩	
EndPosX	102421	力覺引導物件結束時的位置 (X座標)	
EndPosY	102422	力覺引導物件結束時的位置 (Y座標)	
EndPosZ	102423	力覺引導物件結束時的位置 (Z座標)	
EndPosU	102424	力覺引導物件結束時的位置(U座標)	
EndPosV	102425	力覺引導物件結束時的位置(V座標)	

Rev.	2
	_

EndPosW	102426	力覺引導物件結束時的位置 (W座標)
PeakForcesFx	105711	力覺引導序列或力覺引導物件執行時,Fx力的峰值
PeakForcesFy	105712	力覺引導序列或力覺引導物件執行時,Fy力的峰值
PeakForcesFz	105713	力覺引導序列或力覺引導物件執行時,Fz力的峰值
PeakForcesTx	105714	力覺引導序列或力覺引導物件執行時,Tx轉矩的峰值
PeakForcesTy	105715	力覺引導序列或力覺引導物件執行時,Ty轉矩的峰值
PeakForcesTz	105716	力覺引導序列或力覺引導物件執行時,Tz轉矩的峰值
TriggeredForcesFx	109411	力覺引導物件的力達成結束條件時Fx的力
TriggeredForcesFy	109412	力覺引導物件的力達成結束條件時Fy的力
TriggeredForcesFz	109413	力覺引導物件的力達成結束條件時Fz的力
TriggeredForcesTx	109414	力覺引導物件的力達成結束條件時Tx的轉矩
TriggeredForcesTy	109415	力覺引導物件的力達成結束條件時Ty的轉矩
TriggeredForcesTz	109416	力覺引導物件的力達成結束條件時Tz的轉矩
TriggeredPosX	109421	力覺引導物件的力達成結束條件時的位置(X座標)
TriggeredPosY	109422	力覺引導物件的力達成結束條件時的位置(Y座標)
TriggeredPosZ	109423	力覺引導物件的力達成結束條件時的位置(Z座標)
TriggeredPosU	109424	力覺引導物件的力達成結束條件時的位置(U座標)
TriggeredPosV	109425	力覺引導物件的力達成結束條件時的位置(V座標)
TriggeredPosW	109426	力覺引導物件的力達成結束條件時的位置(Z座標)

15.16.12 SpelHand列舉

成員名稱	數值	描述
Righty	1	右手姿態。
Lefty	2	左手姿態。

15.16.13 SpellOLabelTypes列舉

成員名稱	數值	描述
InputBit	1	指定輸入位元。

成員名稱	數值	描述
InputByte	2	指定輸入位元組。
InputWord	3	指定輸入字元。
OutputBit	4	指定輸出位元。
OutputByte	5	指定輸出位元組。
OutputWord	6	指定輸出字元。
MemoryBit	7	指定記憶體位元。
MemoryByte	8	指定記憶體位元組。
MemoryWord	9	指定記憶體字元。
InputReal	10	指定實數輸入。
OutputReal	11	指定實數輸出。

15.16.14 SpelLocalDefType列舉

成員名稱	數值	描述
J5Camera	1	定義J5設定攝像機的本地
J6Camera	2	定義J6設定攝像機的本地
FixedUpwardCamera	3	使用固定向上攝像機,定義本地
FixedDownwardCamera	4	使用固定向下攝像機,定義本地

15.16.15 SpelOperationMode列舉

成員名稱	數值	描述
Auto	1	Epson RC+ 8.0處於auto模式。
Program	2	Epson RC+ 8.0處於program模式。

15.16.16 SpelOptions列舉

成員名稱	數值	描述
ECP	1	ECP選配件
API	2	RC+ API選配件

成員名稱	數值	描述
PCVision	3	PC視覺選配件
ConveyorTracking	5	輸送帶追蹤選配件
GUIBuilder	6	GUI Builder選配件
OCR	7	OCR選配件
FieldbusMaster	8	現場匯流排主站選配件
LegacyForceSensing	9	Force Sensing選配件
PartFeeding	11	Part Feeding選配件
ThirdPartyForceSensors	13	第三方力覺感應器選配 件
OPCUA	4	OPCUA選配件
VRT	10	VRT選配件
SafetyFunction	17	SafetyFunction配件

15.16.17 SpelOptionStatus列舉

成員名稱	數值	描述
Inactive	0	無効
Active	1	有効

15.16.18 SpelRobotPosType列舉

成員名稱	數值	描述
World	0	指定世界座標。
Joint	1	指定關節座標。
Pulse	2	指定脈衝。

15.16.19 SpelRobotType列舉

成員名稱	數值	描述
Joint	1	機器人類型為joint。
Cartesian	2	機器人類型為Cartesian。

成員名稱	數值	描述
Scara	3	機器人類型為SCARA。
Cylindrical	4	機器人類型為Cylindrical。
SixAxis	5	機器人類型為6軸。
RS	6	機器人類型為SCARA RS系 列。
N	7	機器人類型為N系列。

15.16.20 SpelShutdownMode列舉

成員名稱	數值	描述
ShutdownWindows	0	Windows將會關閉。
RebootWindows	1	Windows將會重啟。

15.16.21 SpelSimObjectType列舉

成員名稱	數值	描述
Unknown	-1	未設定物件類型
Layout	0	佈局物件
Part	1	零件物件
MountedDevice	3	機械臂安裝設備

15.16.22 SpelSimProps列舉

成員名稱	數值	描述
PositionX	100	X座標位置
PositionY	200	Y座標位置
PositionZ	300	Z座標位置
RotationX	400	X軸旋轉角度
RotationY	500	Y軸旋轉角度
RotationZ	600	Z軸旋轉角度
CollisionCheck	700	啟用/禁止碰撞偵測

-		
成員名稱	數值	描述
CollisionCheckSelf	800	啟用/禁止自碰撞偵測
Visible	900	顯示/不顯示狀態
Туре	1000	物件的類型
HalfSizeX	1500	Box物件X方向的長度
HalfSizeY	1600	Box物件Y方向的長度
HalfSizeZ	1700	Box物件Z方向的長度
HalfSizeHeight	1800	Plane物件的高度
HalfSizeWidth	1900	Plane物件的寬度
PlaneType	2000	Plane物件的類型
Radius	2100	Sphere或Cylinder物件的半 徑
Height	2200	Cylinder物件的高度
Name	2300	物件名稱
Color	2400	物件顯示顔色

15.16.23 SpelStopType列舉

成員名稱	數值	描述
StopNormalTasks	0	僅停止正常任務(非背景任務)。
StopAllTasks	1	停止所有任務,包含背景任務。

15.16.24 SpelTaskState列舉

成員名稱	數值	描述
Quit	0	任務處於quit狀態。
Run	1	任務處於run狀態。
Aborted	2	任務已終止。
Finished	3	任務已完成。
Breakpoint	4	任務位於中斷點。
Halt	5	任務處於halt狀態。

成員名稱	數值	描述
Pause	6	任務處於pause狀 態。
Step	7	任務處於step狀態。
Walk	8	任務處於walk狀態。
Error	9	任務處於error狀態。
Waiting	10	任務處於wait狀態。

15.16.25 SpelTaskType列舉

成員名稱	數值	描述
Normal	0	標準任務。
NoPause	1	任務沒有受pause影響。
NoEmgAbort	2	任務沒有受緊急停止影響。

15.16.26 SpelToolDefType列舉

成員名稱	數值	描述
J4Camera	1	定義J4安裝攝影機的工具。
J6Camera	2	定義J6安裝攝影機的工具。
FixedCamera	3	透過使用未校準的固定攝影機定義工具。
FixedCameraWithCal	4	透過使用已校準的上方攝影機定義工具。

15.16.27 SpelToolDefType3D列舉

成員名稱	數值	描述	
Bar	1	為條形類型定義3D工具。	
Plane	2	為平面類型定義3D工具。	

15.16.28 SpelUserRights列舉

成員名稱	數值	描述
All	-1	使用者擁有所有權限。
None	0	使用者沒有權限。

EditSecurity	1	使用者可設定安全功能。	
SysConfig	2	使用者可改變系統設定。	
EditPrograms	4	使用者可編輯程式。	
EditPoints	8	使用者可編輯點。	
EditVision	16	使用者可改變視覺屬性。	
JogAndTeach	32	使用者可步進示教。	
CommandWindow	64	使用者可使用命令視窗。	
EditRobotParameters	128	使用者可編輯機器人參數。	
ConfigureOptions	256	使用者可設定選購件。	
ViewAudit	512	使用者可查看安全審計日誌。	
EditProject	1024	使用者可編輯專案設定。	
DeleteAudit	2048	使用者可刪除安全審計日誌項目。	
TeachPoints	4096	使用者可示教點。	
ChangeOutputs	8192	使用者可改變輸出狀態。	
ChangeMemIO	16384	使用者可改變記憶體I/O狀態。	
EditGUIBuilder	32768	使用者可在GUI Builder中進行改變。	
EditForce	65536	使用者可在Force Guide與Force Control中進行改 變。	
EditPartFeeding	131072	使用者可在工件供給中進行改變。	

15.16.29 SpelVDefShowWarning列舉

成員名稱	數值	描述	
None	-1	不顯示警告。	
Always	0	永遠顯示警告。	
DependsOnSpeed	1	RobotSpeed或RobotAccel大於5時顯示。	

15.16.30 SpelVisionImageSize列舉

成員名稱	數值	描述
Size320x240	1	320 x 240影像大小。

成員名稱	數值	描述
Size640x480	2	640 x 480影像大小。
Size800x600	3	800 x 600影像大小。
Size1024x768	4	1024 x 768影像大小。
Size1280x1024	5	1280 x 1024影像大 小。
Size1600x1200	6	1600 x 1200影像大 小。
Size2048x1536	7	2048 x 1536影像大 小。
Size2560x1920	8	2560 x 1920影像大 小。
Size3664x2748	9	3664 x 2748影像大 小。
Size5472x3648	10	5472 x 3648影像大 小。
Size4024x3036	11	4024 x 3036影像大 小。
Size2448x2048	12	2448 x 2048影像大 小。

15.16.31 SpelVisionObjectTypes列舉

成員名稱	數值	描述
Correlation	1	Correlation物件
Blob	2	Blob物件
Edge	3	Edge物件
Polar	4	Polar物件
Line	5	Line物件
Point	6	Point物件
Frame	7	Frame物件
ImageOp	8	ImageOp物件
OCR	9	OCR物件
CodeReader	10	CodeReader物件
Geometric	11	Geometric物件

成員名稱	數值	描述
ArcFinder	16	Arc Finder物件
DefectFinder	17	Defect Finder物件
LineInspector	18	Line Inspector物 件
ArcInspector	19	Arc Inspector物件
BoxFinder	20	Box Finder物件
CornerFinder	21	Corner Finder物件
Contour	22	Contour物件
Text	23	Text物件
Decision	26	Decision物件
Coordinates	27	Coordinates物件

15.16.32 SpelVisionProps列舉

此列舉用於所有視覺屬性與結果。如需詳細資訊,請參閱Vision Guide Reference手冊。

15.16.33	Spel	Wind	lows列	J舉

成員名稱	數值	描述
IOMonitor	1	I/O監視器視窗的ID
TaskManager	2	任務管理器視窗的ID
ForceMonitor	3	力監視器視窗的ID
Simulator	4	模擬器視窗的ID

15.16.34 SpelWrist列舉

成員名稱	數值	描述
NoFlip	1	腕部方向為不翻轉。
Flip	2	腕部方向為翻轉。

15.17 Spel錯誤編號及訊息

如需錯誤編號及錯誤訊息的詳細資訊,請參閱狀態碼和錯誤碼手册。

16.32位元與64位元應用程式

RCAPINet程式庫自動支援32位元與64位元應用程式。

17. 使用LabVIEW VI程式庫

17.1 概述

LabVIEW VI新程式庫具有以下功能:

- 使用VI(虛擬儀器)的Epson RC+ 8.0高階層介面。
- 使用者無須處理Epson RC+ 8.0的.NET介面-此為自動處理。
- 每個Spel命令都包含在個別的VI中。
- VI是在多個工具面板中組織。
- 同時支援32位元與64位元LabVIEW應用程式。
- 支援LabVIEW 2009及更新版本。

若要使用LabVIEW VI程式庫,您必須針對所連接的每個控制器購買RC+ API軟體授權。

17.2 安裝

若要使用Epson RC+ 8.0 LabVIEW VI程式庫,您必須使用PC上的¥EpsonRC80¥API¥LabVIEW資料夾中的安裝程式進行安裝。

- 1. 安裝LabVIEW 2009或更新版本。
- ii. 瀏覽至電腦上的¥EpsonRC80¥API¥LabVIEW資料夾,並執行EpsonRC80_vxxx_LabVIEW.exe安裝程式,其 中xxx表示Epson RC+ 8.0的版本編號。例如: EpsonRC80_v800_LabVIEW.exe。

3. 啟動安裝程式並選擇您正在使用的LabVIEW應用程式的位,它將顯示您PC上安裝的版本。默認情況下,選擇最新版本。選擇要使用Epson RC+ 8.0 LabVIEW VI程式庫的版本。

🔀 Setup - Epson RC+ 8.0 API LabVIEW Library		-		×
Select LabVIEW 32 Bit or 64 Bit				
	Back	<u>N</u> ext	Cance	1
😼 Setup - Epson RC+ 8.0 API LabVIEW Library		-		×
Select LabVIEW 32 Bit Version Select the LabVIEW version to install the library for. Version 19.0 32 Bit				×

17.3 工具面板

安裝Epson RC+ 8.0 LabVIEW VI程式庫後,您可從[Epson Robots 8.0]工具面板和[Epson Robots 8.0]控制面板存取可用的VI及控制項。

工具面板

工具面板是從方塊圖存取。Epson Robots 8.0工具面板內有多個子面板,如下表所示:

面板	描述
System	用於初始化及關閉API。
Robot Settings	變更機器人參數。
Points	載入、儲存、變更機器人點。
Motion	執行機器人動作。
Inputs & Outputs	控制及監控控制器輸入與輸出。
Tasks	管理機器人控制器中的任務。
Variables	讀取及寫入控制器中的變數。
Vision	執行Vision命令。
GUI	顯示GUI功能。

若要存取[Epson Robots 8.0]工具面板,請開啟VI中的方塊圖、用滑鼠右鍵按一下空白區域,然後選擇[Epson Robots 8.0]查看上述子面板。

- Controls Express		Q Search		
Num Ctrls	Buttons	Text Ctrls		
User Ctrls	Num Inds	LEDs		
Text Inds	Graph		- Epson Robots 8.0	
Control & Simulation	Indicators	•	6	6
Epson Robots 8.0	*		SPELVideo.ct I	Vision

控制面板

控制面板是從前面板存取。

面板	描述
Vision	包含用於顯示視訊的SPELVideo控制項。



17.4 開始使用

若要使用LabVIEW VI程式庫,您的應用程式必須先針對要使用的每個控制器調用Spel Initialize VI。Initialize VI會 啟動Epson RC+ 8.0伺服器程序,將會連接至機器人控制器並處理後續的Spel命令VI。Initialize VI具有一個Spel Ref Out輸出。這必須連接至下一個Spel VI Spel Ref In輸入。對於每個後續VI,上一個Spel VI的Spel Ref Out輸出必須 連接至下一個Spel VI的Spel Ref In輸入。

例如:以下流程圖顯示每個Spel節點之間的Spel Ref Out及Spel Ref In連接。



當應用程式關閉時,您必須調用Spel Shutdown VI。這將會中斷與機器人控制器的連接,並關閉關聯的Epson RC+8.0伺服器程序。

依照下列步驟開始使用。首先,您將會從LabVIEW預設Spel+專案的Epson RC+ 8.0 GUI中建立兩個安全機器人點。 接著,您將會在LabVIEW中建置一個小應用程式,用以在兩點之間移動機器人。

- 確認Epson RC+ 8.0及Epson RC+ 8.0 LabVIEW VI程式庫已安裝在PC上。如需安裝LabVIEW VI程式庫的詳細資 訊,請參閱以下內容。
 安裝
- 2. 啟動Epson RC+ 8.0。
- 3. 從[Project]功能表選擇[Open],移動至LabVIEW資料夾並選擇LabVIEW_Default專案。按一下Open。
- 4. 從[Tools]功能表中, 選擇[Robot Manager]。按一下Motor On按鈕。
- 5. 在機器人管理器上, 選擇[Jog & Teach]頁面。將機器人步進至某個安全位置。
- 6. 按一下[Teach]示教點0。
- 7. 將機器人步進至其他安全位置。
- 8. 從Point清單選擇P1, 然後按一下[Teach]示教點1。
- 9. 按一下主要工具條上的[Save]按鈕儲存這些點。
- 10. 關閉Epson RC+ 8.0。
- 11. 啟動LabVIEW並建立新的VI。
- 12. 開啟新VI的方塊圖。
- 13.xiii. 從[Epson Robots 8.0] [System]工具面板,將Init VI拖曳至方塊圖。您所連接的每個控制器都需要Initialize VI。
- 14.xiv. 從[Epson Robots 8.0] [Robot Settings]工具面板,將MotorOn VI拖曳至方塊圖。將Inititalize VI的Spel Ref Out輸出連接至MotorOn VI的Spel Ref In輸入。
- 15. xv. 從[Epson Robots 8.0] [Motion],將Go VI拖曳至方塊圖。將MotorOn VI的Spel Ref Out輸出連接至Go VI 的Spel Ref In輸入。新增Point Number輸入的常數並將數值設為0。
- 16.xvi. 從[Epson Robots 8.0] [Motion]工具面板,將其他Go VI拖曳至方塊圖。將上一個Go VI的Spel Ref Out輸出 連接至第二個Go VI的Spel Ref In輸入。新增Point Number輸入的常數並將數值設為1。
- 17xvii. 從[Epson Robots 8.0] [Robot Settings]工具面板,將MotorOff VI拖曳至方塊圖。將Go VI的Spel Ref Out輸出連接至MotorOff VI的Spel Ref In輸入。
- 1&viii. 從[Epson Robots 8.0] [System]工具面板,將Shutdown VI拖曳至方塊圖。每個Init VI都必須使用Shutdown VI。
 - 方塊圖應看起來如下:



19. 執行應用程式。機器人馬達應該開啟,機器人應移至點0,再移至點1,然後機器人馬達將會關閉。

17.5 使用Spel+專案

您可選擇以LabVIEW應用程式使用Spel+專案。不過,如果您要儲存點資料或想使用點標籤及/或I/O標籤、任務或 視覺序列,則您需要使用Spel+專案。

根據預設,專案為¥EpsonRC80¥Projects¥LabVIEW資料夾中的LabVIEW_Default。

如有需要,您可使用Epson RC+ 8.0建立專屬的專案,然後使用Initialize VI Project輸入參數指定您要使用的專案,如下所示:

SPEL INIT

若要使用Epson RC+ 8.0專案,請啟動Epson RC+ 8.0應用程式。使用Project功能表建立、開啟及編輯專案。如需詳細資訊,請參閱Epson RC+ 使用指南。

17.6 顯示視訊

您可使用SPELVideo控制項及VideoControl VI顯示Vision Guide序列的視訊。 若要顯示視訊:

- 1. 將SPELVideo控制項添加至前面板。
- 2. 將VideoControl.vi添加至對應的方塊圖。
- 3. 在VideoControl VI上,將SPELVideo控制項的輸出連接至SPELVideo Ref In輸入。
- 4. 連接VideoControl VI和VideoControl VI的Spel Ref Out參數。
- 5. 在VideoControl VI上,添加Camera、Graphics Enabled及Video Enabled參數的常數或控制項。Video Enabled必 須設為true才能顯示視訊。

下方流程圖顯示SPELVideo控制項與SPEL VideoControl VI的連接。



當Video Enabled為true, 且VRun從VRun VI或在控制器任務中執行時, 您將會看見產生的視訊(視Camera設定而 定)。

根據預設, Camera輸入參數為零, 這會允許顯示任何攝影機的視訊。如果將Camera設為零以外的數值, 將會針對使 用指定攝影機的序列顯示視訊。 當Graphics Enabled為true且VRun執行時,序列結果圖形會顯示在視訊影像之上。 您在應用程式中一次只能使用一個SPEL Video控制項。

17.7 VI參考

本節包含Epson RC+ 8.0 LabVIEW VI程式庫中所有VI的資訊。 以下為每個VI的資訊:

工具面板	包含VI的工具面板。
描述	函數的簡短描述。
輸入	輸入參數
輸出	輸出參數
備註	其他詳細資訊。
17.7.1 Accel VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

設定目前機器人的PTP加速和減速。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Accel	PTP加速的整數值。
Decel	PTP減速的整數值。
Depart Accel	選用。Jump起始加速的整數值。
Depart Decel	選用。Jump起始減速的整數值。
Appro Accel	選用。Jump接近加速的整數值。
Appro Decel	選用。Jump接近減速的整數值。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

使用Accel設定目前機器人的PTP加速和減速值。所有值可介於1至100%。如果指定Depart Accel,則必須同時指定剩餘輸入。

正常 Accel 設定的最大值是 100, 但也有可設定 100 以上的機種。正常使用時, Accel 設定值 100 是在加減速度與定 位時的震動間取得平衡時的最佳設定。但依據動作條件而定, 也有可能想優先透過減少定位時的震動來提升加減速 度, 以提高週期時間。在這種情況下, 請將 Accel 設定值設定在 100 以上。但依據動作條件而定, 也有可能即使將值 設定在 100 以上, 仍看不到週期時間的變化。

另請參閱

AccelS, Speed, SpeedS

Accel範例



17.7.2 AccelS VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

設定目前機器人的線性加速和減速。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Accel	線性加速的雙精度值。
Decel	線性減速的雙精度值。
Depart Accel	選用。Jump起始加速的雙精度值。
Depart Decel	選用。Jump起始減速的雙精度值。
Appro Accel	選用。Jump接近加速的雙精度值。
Appro Decel	選用。Jump接近減速的雙精度值。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

使用AccelS設定目前機器人的線性加速和減速值。所有值以mm/sec2為單位。如果指定Depart Accel,則必須同時指 定剩餘輸入。

另請參閱

Accel, Speed, SpeedS

AccelS範例



17.7.3 Arc VI

工具面板

Epson Robots 8.0 | Motion

描述

Arc利用XY平面中的圓形插補將手臂移至指定點。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Mid Point Number	使用整數指定中點。
Mid Point Expr	透過使用字串運算式來指定中點。如果使用此輸入,您必須同時使用字串運算式指定結束點。
End Point Number	使用整數指定結束點。
End Point Expr	透過使用字串運算式來指定結束點。使用字串運算式時,您可加入ROT, CP, SYNC, Till, Find,以及平行處理陳述式。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc3, BMove, Go, Jump, Jump3, Move, Speed, TGo, TMove, BGo, Find, Till

Arc範例



17.7.4 Arc3 VI

工具面板

Epson Robots 8.0 | Motion

描述

Arc3在3D中利用圓形插補將手臂移至指定點。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Mid Point Number	使用整數指定中點。
Mid Point Expr	透過使用字串運算式來指定中點。如果使用此輸入,您必須同時使用字串運算式指定結束點。
End Point Number	使用整數指定結束點。
End Point Expr	透過使用字串運算式來指定結束點。使用字串運算式時,您可加入ROT, CP, SYNC, Till, Find,以及平行處理陳述式。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

AccelS, Arc, BMove, Go, Jump, Jump3, Move, SpeedS, TGo, TMove, BGo, Find, Till

Arc3範例



17.7.5 Arch VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

定義要搭配JUMP指令使用的ARCH參數(開始水平動作之前要移動的Z高度)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Arch Number	在開始水平移動之前, Jump指令開始時移動的起始距離(公釐)。
Depart Dist	在開始水平移動之前, Jump指令開始時移動的起始距離(公釐)。
Appro Dist	Jump指令目標位置以上的結束距離(公釐)。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Jump, Jump3

Arch範例



17.7.6 Arm VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

選擇目前機器人手臂。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Arm Number	介於0-15的整數。使用者最多可選擇16種不同的手臂。手臂0是標準(預設)機器人手臂。手臂1-15是透過ArmSet指令定義的輔助手臂。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Armset, GetArm, Tool

Arm範例



17.7.7 Armset VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

Defines an auxiliary robot arm.

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
ArmNumber	整數:有效範圍介於1-15。
Param1	(SCARA機器人適用)肘部中心線到新方向軸中心線的水平距離。(亦即新輔助手臂方向軸中心線所 在的位置。)(Cartesian機器人適用)原始X位置的X軸方向位置偏移(公釐)。
Param2	(SCARA機器人適用)從標準肘部中心線與標準方向軸中心線所形成的直線,到新輔助手臂肘部中 心線與新方向軸中心線所形成的直線,兩者之間的偏移(度)。(這兩條直線應該在肘部中心線相交, 形成的角度即為Param2。)(Cartesian機器人適用)原始Y位置的Y軸方向位置偏移(公釐)。
Param3	(SCARA & Cartesian機器人適用)新方向軸中心與舊方向軸中心之間的Z高度偏移差距。(此為距離。)
Param4	(SCARA機器人適用)肩部中心線到新輔助軸肘部方向之肘部中心線的距離。(Cartesian機器人適用)此為虛擬參數(指定0)
Param5	(SCARA & Cartesian機器人適用)新方向軸與舊方向軸的角度偏移(度)。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

ArmSet範例



17.7.8 AtHome VI

工具面板

Epson Robots 8.0 | Motion

描述

若目前機器人位於起始點位置,會傳回True。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
At Home	表示目前機器人是否位於起始點位置的布林值。

AtHome範例



17.7.9 AvoidSing VI

工具面板

Epson Robots 8.0 | Motion

描述

啟動/停用Move、Arc及Arc3動作方法的奇點避開功能。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Enable	True可啟用奇點避開, False則可停用。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

AvoidSing範例



17.7.10 BGo VI

工具面板

Epson Robots 8.0 | Motion

描述

在所選本地座標系統中執行PTP相對動作。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。如果使用字串表示式,則可以包括SYNC、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BMove, Go, Jump, Jump3, Move, Speed, TGo, TMove, Find, Till

BGo範例



17.7.11 BMove VI

工具面板

Epson Robots 8.0 | Motion

描述

在所選本地座標系統中執行線性插補相對動作。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。如果使用字串表示式,則可以包括ROT、SYNC、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

AccelS, Arc, Arc3, BGo, Go, Jump, Jump3, Move, SpeedS, TGo, TMove, Find, Till

BMove範例



17.7.12 Box VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

指定在盒體內定義的接近檢查區域。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
AreaNumber	介於1-15的整數, 代表15個方塊中要定義的方塊。
Min X	接近檢查區域的最小X座標位置。
Max X	接近檢查區域的最大X座標位置。
Min Y	接近檢查區域的最小Y座標位置。
Max Y	接近檢查區域的最大Y座標位置。
Min Z	接近檢查區域的最小Z座標位置。
Max Z	接近檢查區域的最大Z座標位置。
Polarity On	設定在使用對應遠端輸出時的遠端輸出邏輯。若要在夾具末端位於盒體區域中時將I/O輸出設為開 啟,請使用True。若要在夾具末端位於盒體區域中時將I/O輸出設為關閉,請使用False。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

XYLim

Box範例



17.7.13 Continue VI

工具面板

Epson Robots 8.0 | Tasks

描述

使控制器中暫停的所有任務繼續。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

Continue範例



17.7.14 Delay VI

工具面板

Epson Robots 8.0 | System

描述

指定的毫秒數的延遲處理。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Milliseconds	延遲的毫秒數。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

Delay範例



17.7.15 ECP VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

選擇目前的ECP定義。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
ECPNumber	介於0-15的整數,代表16個ECP定義中要搭配下一個動作指令使用的定義。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

ECPSet, GetECP

ECP範例

10	SPEL	
	201	

17.7.16 ECPset VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

定義ECP(外部控制點)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
ECPNumber	介於1-15的整數,代表15個外部控制點中要定義的外部控制點。
Х	外部控制點X座標。
Y	外部控制點Y座標。
Z	外部控制點Z座標。
U	外部控制點U座標。
V	外部控制點V座標。
W	外部控制點W座標。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

ECP, GetECP

ECPSet範例



448

17.7.17 EStopOn VI

工具面板

Epson Robots 8.0 | System

描述

返回緊急停止狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
EStopOn	緊急停止情況為True,否則為False

另請參閱

SafetyOn

EStopOn範例



17.7.18 Find VI

工具面板

Epson Robots 8.0 | Motion

描述

指定動作期間的座標儲存條件。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Condition	包含函數和運算符的字串運算式。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

使用Find指定在動作期間儲存位置的時間。符合條件時,目前位置會儲存至FindPos。

另請參閱

FindPos

Find範例



17.7.19 Fine VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

指定並顯示目標點的定位準確度。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
J1MaxErr - J9MaxErr	介於0-32767的整數,代表各點允許的定位誤差。關節7、8及9的值為選用。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

Fine範例



17.7.20 GetArm VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

傳回目前機器人的目前手臂編號。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Arm Number	目前手臂編號。

GetArm範例



17.7.21 GetAvoidSing VI

工具面板

Epson Robots 8.0 | Motion

描述

傳回AvoidSingularity傳回值

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
EStopOn	緊急停止情況為True,否則為False

另請參閱

AvoidSing

GetAvoidSing範例



17.7.22 GetECP VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

傳回目前機器人的目前ECP編號。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
ECP Number	目前ECP編號。

GetECP範例

SPEL	ECP Number	
GET ECP	▶ 1.23 TR	

17.7.23 GetLimZ VI

工具面板

Epson Robots 8.0 | Motion

描述

傳回目前的LimZ設定

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Z Limit	目前的LimZ設定。

另請參閱

LimZ

GetLimZ範例



17.7.24 GetMotor VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

傳回目前機器人的馬達開啟狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Motors On	True表示馬達開啟, false表示未開啟。

另請參閱

GetPower, MotorOn, MotorOff

GetMotor範例



17.7.25 GetOprMode VI

工具面板

Epson Robots 8.0 | System

描述

讀取Epson RC+ 8.0操作模式。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。	
Error Out	後續Spel節點的錯誤條件輸出。	
Operation Mode	相關Epson RC+ 8.0伺服器程序的操作模式。	
模式	ID	描述
Auto	1	Epson RC+ 8.0處於auto模式。
Program	2	Epson RC+ 8.0處於program模式。

備註

當Operation Mode設為Program時,相關伺服器程序的Epson RC+ 8.0 GUI會開啟,而控制器操作模式會設為 Program。若使用者關閉RC+ GUI, Operation Mode會設為Auto。如果Operation Mode設為Auto,則RC+ GUI會 同時關閉。

另請參閱

OprMode

GetOprMode範例



17.7.26 GetPoint VI

工具面板

Epson Robots 8.0 | Points

描述

擷取某一機器人點的座標資料。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
X - W	指定點的X、Y、Z、U、V、W座標。

另請參閱

LoadPoints, SavePoints, SetPoint

GetPoint範例



17.7.27 GetPower VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

傳回目前機器人的高運行功率。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Power High	True表示功率高,否則會傳回false。

另請參閱

PowerHigh, PowerLow

GetPower範例



17.7.28 GetRobot VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

傳回目前機器人編號。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Robot Number	目前機器人編號。

另請參閱

Robot

GetRobot範例



17.7.29 GetTool VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

傳回目前機器人的目前工具編號。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Tool Number	目前工具編號。

GetTool範例



17.7.30 GetVar VI

工具面板

Epson Robots 8.0 | Variables

描述

傳回控制器中SPEL+全域保留變數的值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Var Name	SPEL+全域保留變數的名稱。對於陣列,可以傳回整個陣列或只傳回一個元素。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	包含數值的變數。

另請參閱

SetVar

GetVar範例



17.7.31 Go VI

工具面板

Epson Robots 8.0 | Motion

描述

以PTP的形式將手臂從目前位置移至指定點或XY位置。GO指令可同時移動任何組合的機器人軸。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指 定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。如果使用字串表示式,則可以包括SYNC、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Jump, Jump3, Move, Speed, TGo, TMove, Find, Till

Go範例



17.7.32 Halt VI

工具面板

Epson Robots 8.0 | Tasks

描述

暫停執行指定的任務。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Task Number	選用。要暫停之任務的任務編號。任務編號的範圍介於1至32。如果指定Task Name,則會略過 Task Number。
Task Name	選用。指定要暫停之任務的名稱。如果未指定Task Name,將會使用Task Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Quit, Resume

Halt範例



17.7.33 Here VI

工具面板

Epson Robots 8.0 | Points

描述

在目前位置示教一個點。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指 定Point Expression,則會略過Point Number。
Point Name	選用。指定點名稱。如果未指定Point Name,將會使用Point Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

Here範例



17.7.34 HideWindow VI

工具面板

Epson Robots 8.0 | GUI

描述

隱藏先前與ShowWindow一起顯示的Epson RC+ 8.0視窗。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Window ID	要顯示的Epson RC+ 8.0視窗的ID。

視窗名稱	ID	描述
IOMonitor	1	I/O監視器視窗的ID
TaskManager	2	任務管理器視窗的ID
ForceMonitor	3	力監視器視窗的ID
Simulator	4	模擬器視窗的ID

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

RunDialog, ShowWindow

HideWindow範例



17.7.35 In VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定輸入埠的狀態。每個連接埠包含8個輸入位元(一位元組)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含8個輸入位元(一位元組)。如果未指定Label,則會使用Port Number。
Label	選用。包含輸入位元組標籤的字串。如果指定Label,則會略過Port Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。	
Error Out	後續Spel節點的錯誤條件輸出。	
Value	介於0至255的整數運算式,代表輸入埠的狀態。	

另請參閱

InBCD, InW, Sw

In範例



17.7.36 InBCD VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回8個輸入的輸入狀態(使用BCD格式)。(二進位十進碼)

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含8個輸入位元(一位元組)。如果未指定Label,則會使用Port Number。	
Label	選用。包含輸入位元組標籤的字串。如果指定Label,則會略過Port Number。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。	
Error Out	後續Spel節點的錯誤條件輸出。	
Value	介於0至9的整數運算式,代表輸入埠的狀態。	

另請參閱

In, InW, Sw

InBCD範例


17.7.37 InsideBox VI

工具面板

Epson Robots 8.0 | Motion

描述

傳回目前機器人夾具末端是否位於指定盒體區域內。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Area Number	介於1-15的整數,代表15個方塊中要勾選的方塊。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。	
Error Out	後續Spel節點的錯誤條件輸出。	
Status	如果機器人夾具末端位於盒體內,則布林值為True。	

另請參閱

Box, InsidePlane, Plane

InsideBox範例



17.7.38 InsidePlane VI

工具面板

Epson Robots 8.0 | Motion

描述

傳回目前機器人夾具末端是否位於指定平面內。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
AreaNumber	介於1-15的整數,代表15個方塊中要勾選的方塊。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。	
Error Out	後續Spel節點的錯誤條件輸出。	
Status	如果機器人夾具末端位於平面內,則布林值為True。	

另請參閱

Box, InsideBox, Plane

InsidePlane範例



17.7.39 InW VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定輸入字元埠的狀態。每個字元埠包含16個輸入位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含8個輸入位元(一位元組)。如果未指定Label,則會使用Port Number。
Label	選用。包含輸入位元組標籤的字串。如果指定Label,則會略過Port Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	介於0至65535的整數運算式,代表輸入埠的狀態。

另請參閱

In, InBCD, Sw

InW範例



17.7.40 Inertia VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

指定目前機器人的裝載慣性和離心率。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
LoadInertia	雙精度值,用以指定夾具末端關節(包含夾具末端和工件)中心周圍的總慣性力矩(kgm2)。	
Eccentricity	雙精度值,用以指定夾具末端關節(包含夾具末端和工件)中心周圍的離心率(公釐)。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Weight Inertia範例



17.7.41 Initialize VI

工具面板

Epson Robots 8.0 | System

描述

初始化LabVIEW VI程式庫所使用的Spel執行個體。

輸入

Server Product Type	選用。指定要連接的Epson RC+產品。
Connection Number	選用。指定要使用的控制器連線。
Project	選用。指定要使用的Epson RC+專案。
ServerInstance	選用。指定使用Epson RC+伺服器的哪個實例。
ConnectionPassword	選用。連線密碼的字串。如果控制器具有用與控制器連接的密碼,並且密碼未在Epson RC+中[配置] - [電腦與控制器連接]對話方塊中設置,則必須設置連接到控制器的密碼。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

針對要使用之程式庫的每個執行個體,必須調用Initialize VI。

Server Product Type用於指定要使用的Epson RC+產品。預設值為RC80 (Epson RC+ 8.0)。

當未指定Connection Number時,將會使用Epson RC+ 8.0中上次使用的連線。

當未指定Project時,將會使用預設的LabVIEW Epson RC+ 8.0專案。專案必須在以Server Product Type指定的 Epson RC+產品中使用。

另請參閱

Shutdown

Initialize範例



17.7.42 JRange VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

定義指定機器人關節的允許工作範圍(以脈衝為單位)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
JointNumber	介於1-9的整數,代表要指定之JRange的關節。
LowerLimitPulses	代表指定關節下限範圍之編碼器脈衝計數位置的整數。
UpperLimitPulses	代表指定關節上限範圍之編碼器脈衝計數位置的整數。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

JRange範例



17.7.43 JS VI

工具面板

Epson Robots 8.0 | Motion

描述

Jump Sense會偵測手臂是否在完成JUMP指令之前停止(使用SENSE輸入),或手臂是否完成JUMP移動。JS會傳回 Jump Sense狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
JS	若動作期間偵測到SENSE輸入,會傳回True,否則會傳回False。

另請參閱

Jump, Sense

JS範例



17.7.44 JTran VI

工具面板

Epson Robots 8.0 | Motion

描述

執行相對關節移動。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
JointNumber	要移動的特定關節。	
Distance	要移動的距離。旋轉關節的單位為度,線性關節的單位為公釐。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

JTran範例



17.7.45 Jump VI

工具面板

Epson Robots 8.0 | Motion

描述

利用PTP動作將手臂從目前位置移至指定點(先垂直向上移動,接著水平移動,最後向下垂直移至最終目標點)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指 定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。如果使用字串表示式,則可以包括SYNC、Sense、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump3, Move, Speed, TGo, TMove, Find, Sence, Till

Jump範例



17.7.46 Jump3 VI

工具面板

Epson Robots 8.0 | Motion

描述

使用兩個CP動作與一個PTP動作之組合的3D閘道動作。機器人會移到起始點、接近點和目標點。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Depart Point Number	使用整數指定起始點。
Depart Point Expr	使用字串運算式指定起始點。如果使用此輸入,您必須同時使用字串運算式指定接近和目 標點。
Appro Point Number	使用整數指定接近點。
Appro Point Expr	使用字串運算式指定接近點。如果使用此輸入,您必須同時使用字串運算式指定起始和目 標點。
Dest Point Number	使用整數指定目標點。
Dest Point Expr	使用字串運算式指定目標點。如果使用此輸入,您必須同時使用字串運算式指定起始和接 近點。

如果使用字串表示式,则可以包括ROT、SYNC、Sense、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump, Move, Speed, TGo, TMove, Find, Sence, Till

Jump3範例



17.7.47 Jump3CP VI

工具面板

Epson Robots 8.0 | Motion

描述

使用三個CP動作之組合的3D閘道動作。機器人會移到起始點、接近點和目標點。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Depart Point Number	使用整數指定起始點。
Depart Point Expr	使用字串運算式指定起始點。如果使用此輸入,您必須同時使用字串運算式指定接近和目 標點。
Appro Point Number	使用整數指定接近點。
Appro Point Expr	使用字串運算式指定接近點。如果使用此輸入,您必須同時使用字串運算式指定起始和目 標點。
Dest Point Number	使用整數指定目標點。
Dest Point Expr	使用字串運算式指定目標點。如果使用此輸入,您必須同時使用字串運算式指定起始和接 近點。

如果使用字串表示式,则可以包括ROT、SYNC、Sense、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, Move, Speed, TGo, TMove, Find, Sence, Till

Jump3CP範例



17.7.48 LimZ VI

工具面板

Epson Robots 8.0 | Motion

描述

設定JUMP命令之Z軸高度的預設值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Z Limit	Z軸可移動範圍內的座標值。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Jump, Jump3

LimZ範例



17.7.49 LoadPoints VI

工具面板

Epson Robots 8.0 | Points

描述

將SPEL+點文件載入至目前機器人的控制器點記憶體。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
File Name	目前Spel專案中或先前使用SavePoints VI儲存的有效點檔案。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

GetPoint, Robot, SavePoints, SetPoint

LoadPoints範例



17.7.50 MemIn VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定記憶體I/O位元組埠的狀態。每個連接埠包含8個記憶體I/O位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含8個輸入位元(一位元組)。如果未指定Label,則會使用Port Number。
Label	選用。包含輸入位元組標籤的字串。如果指定Label,則會略過Port Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	介於0至255的整數,代表埠的狀態。

另請參閱

MemInW, MemOut, MemOutW

MemIn範例



17.7.51 MemInW VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定記憶體I/O字元埠的狀態。每個字元埠包含16個記憶體I/O位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個字元埠包含16個輸入位元。如果未指定Label,則會使用Port Number。
Label	選用。包含輸入位元組標籤的字串。如果指定Label,則會略過Port Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	介於0至255的整數,代表埠的狀態。

另請參閱

MemIn, MemOut, MemOutW

MemInW範例



17.7.52 MemOut VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

根據使用者指定的8位元值,同時設定8個記憶體I/O位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含8個輸入位元(一位元組)。如果未指定Label,則會使用Port Number。
Label	選用。包含輸入位元組標籤的字串。如果指定Label,則會略過Port Number。
Value	包含指定位元組之輸出模式的整數。有效值介於0-255。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MemIn, MemInW, MemOutW

MemOut範例



17.7.53 MemOff VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

關閉記憶體I/O的指定位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Bit Number	選用。代表任一個記憶體I/O位元的整數。如果未指定Label,則會使用Bit Number。	
Label	選用。包含輸入位元標籤的字串。如果指定Label,則會略過Bit Number。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MemOn, MemOut, MemOutW MemOff範例**



17.7.54 MemOn VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

開啟記憶體I/O的指定位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Bit Number	選用。代表任一個記憶體I/O位元的整數。如果未指定Label,則會使用Bit Number。	
Label	選用。包含輸入位元標籤的字串。如果指定Label,則會略過Bit Number。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MemOff, MemOut, MemOutW

MemOn範例



17.7.55 MemOut VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

根據使用者指定的8位元值,同時設定8個記憶體I/O位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個記憶體I/O埠的整數。每個連接埠包含8個記憶體I/O位元(一位元組)。如果未指定 Label,則會使用Port Number。
Label	選用。包含記憶體I/O位元組標籤的字串。如果指定Label,則會略過Port Number。
Value	包含指定位元組之輸出模式的整數。有效值介於0-255。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MemOn, MemOff, MemOutW

MemOut範例



17.7.56 MemOutW VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

根據使用者指定的16位元值,同時設定16個記憶體I/O位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	選用。代表任一個記憶體I/O埠的整數。每個字元埠包含16個輸入位元。如果未指定Label,則會使用Port Number。
Label	選用。包含記憶體I/O位元組標籤的字串。如果指定Label,則會略過Port Number。
Value	包含指定字元之輸出模式的整數。有效值介於0-65535。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MemOn, MemOff, MemOut

MemOutW範例



17.7.57 MemSw VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定記憶體I/O位元的狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	選用。代表任一個記憶體I/O位元的整數。如果未指定Label,則會使用Bit Number。
Label	選用。包含記憶體I/O位元標籤的字串。如果指定Label,則會略過Bit Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	記憶體I/O位元開啟時為True的布林值。

另請參閱

MemIn, MemInW

MemSW範例



17.7.58 MotorOff VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

關閉目前機器人的馬達。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MotorOn, PowerHigh, PowerLow, Robot

MotorOff範例



17.7.59 MotorOn VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

開啟目前機器人的馬達。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MotorOff, PowerHigh, PowerLow, Robot

MotorOn範例



17.7.60 Move VI

工具面板

Epson Robots 8.0 | Motion

描述

使用線性插補法(即以直線移動)將手臂從目前位置移至指定點。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。

如果使用字串表示式,则可以包括ROT、ECP、SYNC、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

AccelS, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, SpeedS, TGo, TMove, ECP, Find, Till

Move範例



17.7.61 Off VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

關閉指定輸出位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	選用。代表任一個輸出位元的整數。如果未指定Label,則會使用Bit Number。
Label	選用。包含輸入位元標籤的字串。如果指定Label,則會略過Bit Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

Off範例



17.7.62 On VI

工具面板

EpsonRobots 8.0 | Inputs & outputs

描述

開啟指定輸出位元。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	選用。代表任一個輸出位元的整數。如果未指定Label,則會使用Bit Number。
Label	選用。包含輸入位元標籤的字串。如果指定Label,則會略過Bit Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

On範例



17.7.63 OPort VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定輸出位元的狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	選用。代表任一個輸出位元的整數。如果未指定Label,則會使用Bit Number。
Label	選用。包含輸出位元標籤的字串。如果指定Label,則會略過Bit Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	輸出位元開啟時為True的布林值。

另請參閱

In, InW, On, Off, Out, Sw

Oport範例



17.7.64 OprMode VI

工具面板

Epson Robots 8.0 | System

描述

設定Epson RC+ 8.0操作模式。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Operation Mode	相關Epson RC+ 8.0伺服器程序的操作模式。

Mode	ID	描述
Auto	1	Epson RC+ 8.0處於auto模式。
Program	2	Epson RC+ 8.0處於program模式。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

當Operation Mode設為Program時,相關伺服器程序的Epson RC+ 8.0 GUI會開啟,而控制器操作模式會設為 Program。若使用者關閉RC+ GUI, Operation Mode會設為Auto。如果Operation Mode設為Auto,則RC+ GUI會 同時關閉。

另請參閱

GetOprMode

OprMode範例



17.7.65 Out VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

同時設定8個輸出位元(1位元組)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	代表輸入埠的整數(可省略)。 每個連接埠包含8個輸出位元(一位元組)。如果未指定Label,則會使用Port Number。
Label	代表輸入位元組標籤的字串(可省略)。 如果指定Label,則會略過Port Number。
Value	包含指定位元組之輸出模式的整數。 有效範圍介於0至255。

傳回值

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

On, Off, OutW

Out範例



17.7.66 OutW VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

同時設定16個輸出位元(1位元組)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Port Number	代表輸入埠的整數(可省略)。 每個字元埠由16個輸出位元(1字元)組成。如果未指定Label,則會使用Port Number。
Label	代表輸出字元標籤的字串(可省略)。 如果指定Label,則會略過Port Number。
Value	代表指定字元之輸出模式的整數。 有效範圍介於0至65535。

傳回值

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

On, Off, Out

Out範例



17.7.67 Pause VI

工具面板

Epson Robots 8.0 | Tasks

描述

使控制器中的所有正常SPEL+任務暫停。若機器人正在移動,將會立即減速至停止。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Continue, Stop

Pause範例



17.7.68 Plane VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

定義平面。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Plane Number	介於1-15的整數運算式,代表15個平面中要定義的平面。
Х	平面座標系統原點X座標。
Y	平面座標系統原點Y座標。
Z	平面座標系統原點Z座標。
U	繞Z軸的平面座標系統旋轉。
V	繞Y軸的平面座標系統旋轉。
W	繞X軸的平面座標系統旋轉。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Box, InsideBox, InsidePlane

Plane範例



17.7.69 PowerHigh VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

將目前機器人的馬達運行功率設為高。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MotorOff, MotorOn, PowerLow, Robot

PowerHigh範例



17.7.70 PowerLow VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

將目前機器人的馬達運行功率設為低。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

MotorOff, MotorOn, PowerHigh, Robot

PowerLow範例



17.7.71 Quit VI

工具面板

Epson Robots 8.0 | Tasks

描述

終止執行指定的任務。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Task Number	選用。要終止之任務的任務編號。任務編號的範圍介於1至32。如果指定Task Name,則會略過 Task Number。
Task Name	選用。指定要終止之任務的名稱。如果未指定Task Name,將會使用Task Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Halt, Resume

Quit範例



17.7.72 Reset VI

工具面板

Epson Robots 8.0 | System

描述

將控制器重置為初始狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

Reset範例


17.7.73 Resume VI

工具面板

Epson Robots 8.0 | Tasks

描述

使遭到Halt VI暫停的任務重新開始。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Task Number	選用。要恢復之任務的任務編號。任務編號的範圍介於1至32。如果指定Task Name,則會略過 Task Number。
Task Name	選用。指定要恢復之任務的名稱。如果未指定Task Name,將會使用Task Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Halt, Quit

Resume範例



17.7.74 Robot VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

選擇目前機器人。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Robot Number	介於1-16的整數。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

GetRobot, MotorOff, MotorOn, PowerHigh, PowerLow

Robot範例



17.7.75 RunDialog VI

工具面板

Epson Robots 8.0 | GUI

描述

執行Epson RC+ 8.0對話方塊。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Dialog ID	要執行之Epson RC+ 8.0對話方塊的ID。

對話方塊名稱	ID	描述
RobotManager	1	工具 機器人管理器對話方塊的ID
ControllerTools	2	工具 控制器對話方塊的ID
VisionGuide	3	工具 Vision Guide對話方塊的ID

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

RunDialog範例



17.7.76 SafetyOn VI

工具面板

Epson Robots 8.0 | System

描述

傳回安全門的狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
SafetyOn	安全門打開時為True, 否則為False

另請參閱

EStopOn

SafetyOn範例



17.7.77 SavePoints VI

工具面板

Epson Robots 8.0 | Points

描述

保存目前機器人的點至檔案。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
File Name	在目前Spel專案中的點檔案名稱,或要儲存至控制器的新檔案名稱。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

GetPoint, LoadPoints, Robot, SetPoint

SavePoints範例



17.7.78 Sense VI

工具面板

Epson Robots 8.0 | Motion

描述

指定輸入條件,符合條件時會停止目標位置上方的機器人,以完成進行中的Jump。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Condition	使用字串運算式來指定I/O條件。如需詳細資訊,請參閱SPEL+語言參考手冊中的Sense陳述式。

輸出

sSpel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

JS, Jump

Sense範例

	Г		1
MemSW(0) = On	4	SPEL	
memori(o) = on	I	SENSE	

17.7.79 SetPoint VI

工具面板

Epson Robots 8.0 | Points

描述

設定目前機器人某個點的座標資料。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指 定Point Name,則會略過Point Number。
Point Name	選用。透過使用點名稱的字串運算式指定點。如果未指定Point Name,將會使用Point Number輸入。
X - W	指定點的X、Y、Z、U、V、W座標。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

GetPoint, LoadPoints, Robot, SavePoints

SetPoint範例



17.7.80 SetVar VI

工具面板

Epson Robots 8.0 | Variables

描述

設定控制器中SPEL+全域保留變數的值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Var Name	SPEL+全域保留變數的名稱。
Value	包含數值的變數。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

GetVar

SetVar範例



17.7.81 SFree VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

從伺服系統控制,釋放指定的機器人軸。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Axes	選用。指定要釋放之軸的整數陣列。如果略過,所有軸會釋放。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如果略過Axes, 則所有軸會釋放。

另請參閱

MotorOff, MotorOn, Slock

SFree範例



17.7.82 ShowWindow VI

工具面板

Epson Robots 8.0 | GUI

描述

顯示Epson RC+ 8.0視窗。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Window ID	要顯示的Epson RC+ 8.0視窗的ID。

視窗名稱	ID	描述
IOMonitor	1	I/O監視器視窗的ID
TaskManager	2	任務管理器視窗的ID
ForceMonitor	3	力監視器視窗的ID
Simulator	4	模擬器視窗的ID

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

HideWindow, RunDialog

ShowWindow範例



17.7.83 Shutdown VI

工具面板

Epson Robots 8.0 | System

描述

關閉在Initialize VI調用時啟動的Epson RC+ 8.0伺服器程序。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Error Out	後續Spel節點的錯誤條件輸出。
-----------	------------------

備註

針對程式庫的每個執行個體,必須調用Shutdown VI。將會關閉相關的Epson RC+ 8.0伺服器程序。

另請參閱

Initialize

Shutdown範例



17.7.84 SLock VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

將指定的機器人軸傳回伺服系統控制。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Axes	選用。指定要鎖定之軸的整數陣列。如果略過,所有軸會鎖定。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如果略過Axes,則所有軸會鎖定。

另請參閱

MotorOff, MotorOn, SFree

SLock範例



17.7.85 Speed VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

指定要搭配PTP指令Go、Jump及Pulse使用的手臂速度。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
PointToPoint Speed	指定要搭配PTP指令Go、Jump及Pulse使用的手臂速度。
Depart Speed	介於1-100的整數, 代表Jump指令的Z軸向上動作速度。
Appro Speed	介於1-100的整數, 代表Jump指令的Z軸向下動作速度。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

使用Speed設定目前機器人的PTP速度。所有值可介於1至100%。如果指定Depart Speed,则必須同時指定Appro Speed。

另請參閱

Accel, AccelS, SpeedS

Speed範例



17.7.86 SpeedS VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

指定要搭配連續路徑指令Jump3CP、Move、Arc及CVMove使用的手臂速度。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Linear Speed	指定要搭配連續路徑指令Jump3CP、Move、Arc及CVMove使用的手臂速度。
Depart Speed	介於1-5000的雙精度值,代表Jump3CP指令的Z軸向上動作速度。
Appro Speed	介於1-5000的雙精度值,代表Jump3CP指令的Z軸向下動作速度。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

使用Speed設定目前機器人的線性速度(mm / sec)。如果指定Depart Speed,則必須同時指定Appro Speed。

另請參閱

Accel, AccelS, Speed

SpeedS範例



17.7.87 Start VI

工具面板

Epson Robots 8.0 | Tasks

描述

啟動要在控制器中執行的程式。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
ProgramNumber	要啟動的程式編號,對應至SPEL+中的8個main函數,如下表所示。範圍介於0至63。

程式編號	SPEL+函數名稱
0	main
1	main1
2	main2
3	main3
4	main4
5	main5
6	main6
63	main63

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

執行Start時,控制將會立即回到調用VI。您無法啟動正在執行的程式。請注意,Start會清除控制器中的全域變數且 會載入預設的機器人點。

另請參閱

Continue, Pause, Stop, Xqt

Start範例



17.7.88 Stop VI

工具面板

Epson Robots 8.0 | Tasks

描述

停止控制器中執行的所有一般SPEL+任務,以及選擇性停止所有背景任務。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Stop Type	選用。指定StopNormalTasks(預設)或StopAllTasks(也會停止背景任務)。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Continue, Pause, Start, Xqt

Stop範例



17.7.89 Sw VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回指定輸入位元的狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	選用。代表任一個輸入位元的整數。如果未指定Label,則會使用Bit Number。
Label	選用。包含輸入位元標籤的字串。如果指定Label,則會略過Bit Number。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	輸出位元開啟時為True的布林值。

另請參閱

In, InW, On, Off, OPort, Out

SW範例



17.7.90 TargetOK VI

工具面板

Epson Robots 8.0 | Motion

描述

傳回指示目前位置到目標位置的PTP(點至點)動作是否可行的狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Target OK	機器人可移至目標位置。
At Home	表示目前機器人是否位於起始點位置的布林值。

另請參閱

BGo, Go, Jump, TGo

TargetOK範例



17.7.91 TGo VI

工具面板

Epson Robots 8.0 | Motion

描述

在目前工具座標系統中執行PTP相對動作。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指 定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。

如果使用字串表示式,则可以包括SYNC、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, Move, Speed, TMove, Find, Till

TGo範例



17.7.92 Till VI

工具面板

Epson Robots 8.0 | Motion

描述

指定事件條件,符合條件時會減速並停止中間位置的機器人,以完成進行中的動作命令(Jump、Go、Move等)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Condition	使用字串運算式來指定I/O條件。如需詳細資訊,請參閱SPEL+語言參考手冊中的Till陳述式。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Jump, Jump3, Move, Speed, TGo, TillOn, TMove

Till範例



17.7.93 TillOn VI

工具面板

Epson Robots 8.0 | Motion

描述

若till條件在最後Go/Jump/Move陳述式期間發生停止狀況, 會傳回True。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Till On	若動作期間偵測到till條件,會傳回True。否則會傳回False。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Jump, Jump3, Move, Speed, TGo, Till, TMove

TillOn範例



17.7.94 TLSet VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

定義ECP(外部控制點)。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
ToolNumber	介於1-15的整數運算式,代表15個工具中要定義的工具。(Tool 0為預設工具,無法改變。)
Х	工具座標系統原點X座標。
Y	工具座標系統原點Y座標。
Z	工具座標系統原點Z座標。
U	繞Z軸的工具座標系統旋轉。
V	繞Y軸的工具座標系統旋轉。
W	繞X軸的工具座標系統旋轉。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

TLSet範例



17.7.95 TMove VI

工具面板

Epson Robots 8.0 | Motion

描述

在所選工具座標系統中執行線性插補相對動作。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指 定Point Expression,則會略過Point Number。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定Point Expression,將會使用Point Number輸入。

如果使用字串表示式,則可以包括ROT、SYNC、Till、Find和並行處理語句。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

AccelS, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, Move, SpeedS, Tgo, Find, Till

TMove範例



17.7.96 Tool VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

選擇目前機器人工具。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Tool Number	介於0-15的整數,代表16個工具定義中要搭配後續動作指令使用的工具定義。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Arm, Armset, GetTool, TLSet

Tool範例



17.7.97 TW VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

傳回Wait命令的條件是否成立命令。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Status	傳回Wait命令的條件是否成立命令。

TW範例



17.7.98 VGetBool VI

工具面板

Epson Robots 8.0 | Vision

描述

擷取傳回布林值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	布林值。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

Vrun, VgetDbl, VgetInt, VgetStr, VsetBool, VsetDbl, VsetInt, VsetStr

VGetBool範例



17.7.99 VGetDbl VI

工具面板

Epson Robots 8.0 | Vision

描述

擷取傳回雙精度值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	雙精度值。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

Vrun, VgetBool, VgetInt, VgetStr, VsetBool, VsetDbl, VsetInt, VsetStr

VGetDbl範例



17.7.100 VGetInt VI

工具面板

Epson Robots 8.0 | Vision

描述

擷取傳回的整數值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	整數值。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

Vrun, VgetBool, VgetDbl, VgetStr, VsetBool, VsetDbl, VsetInt, VSetStr

VGetInt範例



17.7.101 VGetStr VI

工具面板

Epson Robots 8.0 | Vision

描述

擷取傳回字串值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Value	字串值。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

Vrun, VgetBool, VgetDbl, VgetInt, VsetBool, VsetDbl, VsetInt, VSetStr

VGetStr範例



17.7.102 VideoControl VI

工具面板

Epson Robots 8.0 | Vision

描述

為SPEL Video控制項進行設定,以顯示視覺系統的影像。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
VideoRef In	來自SPEL Video控制項的參考。
Camera	設定要顯示的攝影機視訊。預設值為0, 會顯示任何攝影機。
Graphics Enabled	設定是否要顯示圖形。
Video Enabled	設定是否要顯示視訊。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Displaying Video, VGet, Vrun

VideoControl範例



17.7.103 VRun VI

工具面板

Epson Robots 8.0 | Vision

描述

在目前專案中執行視覺序列。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如需執行視覺序列的資訊,請參閱Vision Guide 8.0軟體手冊。

另請參閱

VgetBool, VgetDbl, VgetInt, VgetStr, VsetBool, VsetDbl, VsetInt, VSetStr

VRun範例



17.7.104 VSetBool VI

工具面板

Epson Robots 8.0 | Vision

描述

設定資料類型為布林值的視覺屬性值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。
Property Code	屬性程式碼。
Value	屬性的新布林值。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

VgetBool, VgetDbl, VgetInt, VgetStr, Vrun, VsetDbl, VsetInt, VSetStr

VSetBool範例



17.7.105 VSetDbl VI

工具面板

Epson Robots 8.0 | Vision

描述

設定資料類型為實數或雙精度值的視覺屬性值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Sequence	包含目前專案中之視覺序列的名稱。	
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。	
Property Code	屬性程式碼。	
Value	屬性的新雙精度值。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

VgetBool, VgetDbl, VgetInt, VgetStr, Vrun, VsetBool, VsetInt, VSetStr

VSetDbl範例



17.7.106 VSetInt VI

工具面板

Epson Robots 8.0 | Vision

描述

設定資料類型為整數的視覺屬性值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。	
Error In	來自上一個Spel節點的錯誤條件。	
Sequence	包含目前專案中之視覺序列的名稱。	
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。	
Property Code	屬性程式碼。	
Value	屬性的新整數值。	

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

VgetBool, VgetDbl, VgetInt, VgetStr, Vrun, VsetBool, VsetDbl, VSetStr

VSetInt範例



17.7.107 VSetStr VI

工具面板

Epson Robots 8.0 | Vision

描述

設定資料類型為字串的視覺屬性值。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。
Property Code	屬性程式碼。
Value	屬性的新字串值。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

備註

如需Vision Guide屬性與結果的詳細資訊,請參閱Vision Guide 8.0屬性與結果參考手冊。

另請參閱

VgetBool, VgetDbl, VgetInt, VgetStr, Vrun, VsetBool, VsetDbl, VsetInt

VSetStr範例


17.7.108 WaitMem VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

等待記憶體位元狀態改變。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	代表記憶體I/O位元的整數(可省略)。 如果未指定Label,則會使用Bit Number。
Label	代表記憶體位元標籤的字串(可省略)。 如果指定Label,則會略過Bit Number。
Condition	代表記憶體位元狀態的Boolean運算式。
Timeout	代表最長等待時間的整數(以秒為單位)。

傳回值

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

WaitSw

WaitMem範例



17.7.109 WaitSw VI

工具面板

Epson Robots 8.0 | Inputs & Outputs

描述

等待輸入位元狀態改變。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Bit Number	代表輸入位元的整數(可省略)。 如果未指定Label,則會使用Bit Number。
Label	代表輸入位元標籤的字串(可省略)。 如果指定Label,則會略過Bit Number。
Condition	代表輸入位元狀態的Boolean運算式。 Timeout 代表最長等待時間的整數(以秒為單位)。

傳回值

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

WaitMem

WaitSw範例



17.7.110 WaitTaskDone VI

工具面板

Epson Robots 8.0 | Tasks

描述

等待任務完成並傳回狀態。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Task Number	選用。要暫停之任務的任務編號。任務編號的範圍介於1至32。如果指定Task Name,則會略過 Task Number。
Task Name	選用。指定要暫停之任務的名稱。如果未指定Task Name,將會使用Task Number輸入。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。
Task State	代表任務的最終狀態(Quit、Aborted、Finished)。

另請參閱

Xqt

WaitTaskDone範例



17.7.111 Weight VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

指定目前機器人的重量參數。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Payload Weight	待運送之夾具末端的重量(公斤)。
Arm Length	從第二手臂旋轉中心至夾具末端重心的距離(公釐)。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Inertia

Weight範例



17.7.112 Xqt VI

工具面板

Epson Robots 8.0 | Tasks

描述

啟動一個SPEL+任務。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Task Number	選用。要執行之任務的任務編號。任務編號的範圍介於1至32。如果略過Task Number,將會自動指派任務編號。
Func Name	指定要執行之函數的名稱。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Halt, Quit, Resume, WaitForTaskDone

XQT範例



17.7.113 XYLim VI

工具面板

Epson Robots 8.0 | Robot Settings

描述

設定機械手的允許動作範圍限制。

輸入

Spel Ref In	來自上一個Spel Ref Out的Spel參考。
Error In	來自上一個Spel節點的錯誤條件。
Min X	機器人可移動的最小X座標位置。(機器人無法移至小於min X的X座標位置。)
Max X	機器人可移動的最大X座標位置。(機器人無法移至大於max X的X座標位置。)
Min Y	機器人可移動的最小Y座標位置。(機器人無法移至小於min Y的Y座標位置。)
Max Y	機器人可移動的最大Y座標位置。(機器人無法移至大於max Y的Y座標位置。)
Min Z	機器人可移動的最小Z座標位置。(機器人無法移至小於min Z的Z座標位置。)
Max Z	機器人可移動的最大Z座標位置。(機器人無法移至大於max Z的Z座標位置。)

備註

XYLim係用來定義動作範圍限制。許多機器人系統都可讓使用者定義關節限制,而SPEL+語言則可允許同時定義關節限制與動作範圍限制。此可讓使用者有效地為其應用建立工作空間。(請記住,關節範圍限制也可以使用SPEL定義。)

使用XYLim值建立的動作範圍只會套用至動作命令目標位置,不會套用到起始位置至目標位置的動作路徑。因此, 手臂可能會在動作期間超出XYLim範圍以外。(亦即XYLim範圍不會影響脈衝。)

若要關閉動作範圍限制,請將範圍限制參數指定為0。

輸出

Spel Ref Out	下一個要使用的VI的Spel參考輸出。
Error Out	後續Spel節點的錯誤條件輸出。

另請參閱

Box

XYLim範例



18. LabVIEW與RCAPINt2.dll一起使用

在「使用LabVIEW VI程式庫」章節所述的LabVIEW VI程式庫,是使用了RCAPINt2.dll的高階層介面。某些使用者可能想直接連接RCAPINt2.dll,而不使用高階層程式庫。本章包含LabVIEW與RCAPINt2.dll的搭配使用資訊。將針對下列主題進行說明。

- 初始化
- 在應用程式中使用Spel屬性及方法
- 關閉
- 使用對話方塊與視窗

18.2 初始化

18.2.1 添加Spel類別的建構函式節點

從Spel類別調用方法或使用屬性之前,您必須使用建構函式節點建立Spel類別的執行個體。您應在應用程式中使用一個Spel類別執行個體。

在將會包含Spel類別執行個體之VI的方塊圖中,從[RC+ API] – [.NET pallete]中添加建構函式節點。此時會顯示 [Select .NET Constructor]對話方塊。

在[Assembly]列表中選擇「RCAPINt2(1.0.0.0)」,再從[Objects]列表中選擇「Spel」,如下所示。

RCAPINet(2.0.0.0)	▼ Browse
Objects	
🗖 RCAPINet	-
 RCProductType 	
• Spel	
Spel+EventReceivedEventHandler	
SpelAxis	
SpelBaseAlignment SpelConnectionInfo	
 SpeiConnectionType 	
SpeiControllerInfo	
obercontrollerino	
Constructors	
Spel()	· · · · · · · · · · · · · · · · · · ·

按一下[OK],在方塊圖中建立Spel的建構函式節點。

_개 Gool Spel 권

18.2.2 初始化Spel類別執行個體

添加Spel類別Initialize方法的Invoke節點。

連接ServerOutOfProcess節點的參考輸出和Initialize節點的參考輸入。執行Initialize時,將會在背景中設定並啟動

RC+作為伺服器。 ServerOutOfProcess為True時, RC+將在單獨的進程中啟動。

18.2.3 連接至控制器並設定專案

添加Spel類別Connect方法的Invoke節點。

針對您要使用的控制器連線,設定ConnectionNumber參數。

若要查看連線數量,請啟動Epson RC+ 8.0,然後選擇[設定]-[電腦與控制器通信]。

添加Spel類別Project屬性的Property節點。

將Project參數設定到所需的專案檔。



18.3 在應用程式中使用Spel屬性及方法

對您的應用程式添加更多節點,以使用Spel屬性與方法。

您必須將先前節點的參考輸出連接到目前節點的參考輸入。此可讓每個屬性或方法使用您在先前步驟所建立及初始化的Spel類別執行個體。

有關可使用的屬性與方法,請參閱以下內容。

RCAPINet參考

18.4 關閉

使用Spel類別執行個體後,您必須叫用Dispose方法。此將會關閉與Spel類別執行個體相關的Epson RC+ 8.0伺服器。 一般而言,您應該在應用程式結束時調用Dispose。

作為跨進程伺服器使用Epson RC+時:

若在未叫用Dispose的情況下中斷應用程式, RC+會繼續進行處理。若重新啟動應用程式, 則正在執行的RC+程序會 重啟。如您嘗試執行RC+ GUI, 系統將會詢問您是否要執行其他RC+執行個體。在此情況下, 您可從Windows任務 管理器中先終止RC+ 程序(erc80.exe), 然後執行Epson RC+ 8.0 GUI。

18.5 使用對話方塊與視窗

.NET父表單搭配.NET應用程式使用時,通常會作為從Spel類別執行個體顯示之對話方塊與視窗的父項。 但LabVIEW並不會使用.NET表單,因此,若要從LabVIEW顯示視窗與對話方塊,請使用ParentWindowHandle屬 性。

請將其設為VI的視窗控制代碼。您可調用Windows API FindWindow方法來取得視窗控制代碼。 使用ParentWindowHandle時,您必須直接調用Spel.ShowWindow(無父參數)。

19. 如何從一台PC控制多個控制器

19.1 概述

使用RC+ API時,您可從一台PC控制最多六個機器人控制器。 若要控制多個控制器,必須針對每個控制器具現化RCAPINet Spel類別。

下圖顯示使用RC+ API控制多個控制器的基本系統設定圖。



應用程式透過為每個控制器所準備的伺服器(RCAPINet Spel類別)控制多個控制器。

19.1.1 系統需求

PC的推薦性能,請參閱以下手冊。 "Epson RC+ 使用指南 - 系統需求"

⚠ 注意

使用低於推薦性能的PC時,請客戶充分確認動作后使用。如果使用的 PC 性能低於上述性能,則可能無法穩定地控制。

19.1.2 電腦與控制器連接

第一個控制器的連線類型可以為USB或乙太網。剩餘控制器的連線類型必須為乙太網。 下圖顯示PC和多個控制器的基本設定圖。



機器人控制器支援連接到Epson RC+ 8.0的控制器。



\Lambda 注意

如果PC有安裝防毒軟體·則執行防毒掃描時·與控制器的通信可能異常中斷。若要執行防毒掃描·請先中斷與控 制器的通信。

19.2 控制多個控制器的限制

多個控制器的控制限制詳見下列章節的說明。

19.2.1 控制器選購件的限制

由每個控制器所控制的下列控制器選購件具有相關限制。

- PC視覺
- 現場匯流排主控端

當上述三個選購件的其中一個已連接到作用中的控制器時,這些選購件無法用於其他(第二個或後者)控制器。

19.2.2 模擬器的限制

模擬器視窗顯示

Epson RC+8.0模擬器視窗可從.NET應用程式使用。有關詳細資訊,請參閱以下內容。

視窗

當連接多個控制器時,若針對每個控制器開啟模擬器視窗,則與不顯示模擬器視窗相比,循環時間可能增加100至 200 msec。

此外,如果在模擬器視窗開啟的情況下執行程式,CPU使用率會提高將近100%,可能對PC造成巨大負載。

除非是進行程式偵錯,否則建議在模擬器視窗關閉的情況下使用系統。

碰撞偵測

若要使用模擬器避免與周邊設備發生碰撞,請在模擬器物件周圍設定15公釐以上的邊緣,以避免碰撞偵測。 模擬器的碰撞偵測無法保證準確度。應用於實際系統時,務必設定邊緣並確保正常運作。 如需各種限制的詳細資訊,請參閱Epson RC+使用指南 中的 模擬器規格與限制。

19.3 連接多個控制器的範例程式

下節說明使用.NET應用程式將PC連接到控制器1和控制器2的範例程式。

▶ 提示

有關可使用的屬性與方法,請參閱以下內容。

RCAPINet參考

19.3.1 控制連線設定

同時連接至多個控制器時,使用Spel類別的Connect方法指定連線。

m_spel.Connect(1)

Connect方法中的參數代表連線編號。此數字與下方對話方塊中「Number」中所示的數字相同(Epson RC+ 8.0功能表-[Setup]-[PC to Controller Communications])。如果數值使用-1,表示使用最近的連線。

C to Co	ontroller Communications				?	×	
Current Connection: 1 Connection Status: Disconnected							
Number	Name	Туре	IP A	Address	Connect		
1	USB	USB	N/A		connect		
2	C4	Virtual	N/A		Disconne	ct	
3	G6	Virtual	N/A		Disconne		
6	RS	Virtual	N/A		Add		
					Delete		
					Password		
🗌 Work Offline 🗹 Auto Connect							
		Rest	ore	Apply	Close		

19.3.2 專案設定

若要連接多個控制器,請使用Spel類別的Project屬性指定專案。每個控制器都必須使用獨立的專案。

```
m_spel.Project = "c:\EpsonRC80\projects\API_Demos\Demo1\Demo1.sprj"
```

19.3.3 使用Visual Basic的範例程式

```
1. 在Visual Studio .NET中, 選擇功能表-[File]-[New]-[Project]。
```

- 2. 建立Visual Basic專案。
- 3. 選擇功能表-[Project]-[Add Reference]。

4. 選擇[瀏覽]標籤, 參照「(Program folder)¥API¥Assembly¥net462」目錄, 然後選擇「RCAPINt2.dll」檔案。

5. 添加兩個按鈕(btnController1、btnController2)至Form1類別。

6. 添加各按鈕的快速事件, 並建立執行緒以控制各個機器人控制器。

```
Private trd1 As System. Threading. Thread
                                            ·機器人控制器1

    機器人控制器2

Private trd2 As System. Threading. Thread
Private Sub btnController1 Click(ByVal sender As System.Object,
ByVal e As System. EventArgs) Handles btnController1. Click
      ' 啟動機器人控制器1的執行緒
      trd1 = New System.Threading.Thread(
New System. Threading. ThreadStart (AddressOf StartController1))
      trd1.Start()
End Sub
Private Sub StartController1()

    控制機器人控制器1

    Try
       Dim frm1 As New frmDemo1
       frm1.ShowDialog()
       frm1.Dispose()
    Catch ex As Exception
      MsgBox(ex.Message)
```

```
End Try
End Sub
Private Sub btnController2 Click(ByVal sender As System.Object,
ByVal e As System. EventArgs) Handles btnController2. Click

        ·        啟動機器人控制器2的執行緒

     trd2 = New System.Threading.Thread(
New System.Threading.ThreadStart(AddressOf StartController2))
     trd2.Start()
End Sub
Private Sub StartController2()

    控制機器人控制器2

    Try
       Dim frm2 As New frmDemo2
       frm2.ShowDialog()
       frm2.Dispose()
    Catch ex As Exception
       MsgBox(ex.Message)
    End Try
End Sub
```

7. 添加控制器1的表單(frmDemo1)。

```
Private WithEvents m spell As New Spel
Private Sub frmDemol Load (ByVal sender As System.Object, _
ByVal e As System. EventArgs) Handles MyBase. Load
     Try
        m spel1.ServerInstance=1
        m spell.Initialize()
        m spell.Connect(5)
        m spell.Project = " c:\\EpsonRC80\\Projects\\Demo1\\Demo1.sprj "
     Catch ex As SpelException
       MsqBox (ex.Message)
     End Try
End Sub
Private Sub m spell EventReceived (ByVal sender As Object, ByVal e As
 SpelEventArgs) Handles m spel1.EventReceived

    機器人控制器1

End Sub
Private Sub frmDemol FormClosed(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.FormClosedEventArgs)
Handles MyBase.FormClosed
   m spell.Dispose()
End Sub
```

8. 添加控制器2的表單(frmDemo2)。

```
Private WithEvents m_spel2 As New Spel
Private Sub frmDemo2_Load(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles MyBase.Load

Try
    m_spel2.ServerInstance=2
    m_spel2.Initialize()
    m_spel2.Connect(6)
    m_spel2.Project = " c:\\EpsonRC80\\Projects\\Demo2\\Demo2.sprj "
    Catch ex As SpelException
    MsgBox(ex.Message)
    End Try
End Sub
Private Sub m_spel2_EventReceived(ByVal sender As Object, ByVal e As
SpelEventArgs) Handles m spel2.EventReceived
```

```
    機器人控制器2
    End Sub
    Private Sub frmDemo2_FormClosed(ByVal sender As System.Object, _
    ByVal e As System.Windows.Forms.FormClosedEventArgs) _
    Handles MyBase.FormClosed
    m_spel2.Dispose()
    End Sub
```

19.3.4 使用Visual C#的範例程式

```
1. 在Visual Studio .NET中, 選擇功能表-[File]-[New]-[Project]。
2. 建立Visual C#專案。
3. 選擇功能表-[Project]-[Add Reference]。
4. 選擇[瀏覽]標籤, 參照「(Program folder)¥API¥Assembly¥net462」目錄, 然後選擇「RCAPINt2.dll」檔案。
5. 添加兩個按鈕(btnController1、btnController2)至Form1類別。
6. 添加各按鈕的快速事件, 並建立執行緒以控制各個機器人控制器。
private System.Threading.Thread trd1; // 機器人控制器1
private System.Threading.Thread trd2; // 機器人控制器2
private void btnController1 Click(object sender, EventArgs e)
{
    // 啟動機器人控制器1的執行緒
   trd1 = new System.Threading.Thread(new
                       System.Threading.ThreadStart(StartController1));
   trd1.Start();
}
private void StartController1()
{
   // 控制機器人控制器1
   try
    {
      frmDemo1 frm1 = new frmDemo1();
      frm1.ShowDialog();
      frm1.Dispose();
   }
   catch (System.Exception ex)
       MessageBox.Show(ex.Message);
private void btnController2 Click(object sender, EventArgs e)
{
   // 啟動機器人控制器2的執行緒
   trd2 = new System.Threading.Thread(new
System.Threading.ThreadStart(StartController2));
   trd2.Start();
}
private void StartController2()
   // 控制機器人控制器2
   try
    {
      frmDemo2 frm2 = new frmDemo2();
      frm2.ShowDialog();
      frm2.Dispose();
    }
   catch (System.Exception ex)
```

```
{
    MessageBox.Show(ex.Message);
}
```

```
7. 添加控制器1的表單(frmDemo1)。
```

```
private Spel m spel1;
private void frmDemol Load(object sender, EventArgs e)
{
    m spel1 = new Spel();
    try
    {
       m spel1.ServerInstance = 1;
       m spel1.Initialize();
       m spel1.Connect(5);
       m spel1.Project = "c:\\EpsonRC80\\Projects\\Demo1\\Demo1.sprj";
       m spel1.EventReceived += new
Spel.EventReceivedEventHandler(m spel1 EventReceived);
    }
    catch (SpelException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
public void m spell EventReceived (object sender, SpelEventArgs e)
{
    // 機器人控制器1
}
private void frmDemol FormClosed(object sender, FormClosedEventArgs e)
    m spell.Dispose();
```

```
8. 添加控制器2的表單(frmDemo2)。
```

```
private Spel m spel2;
private void frmDemo2 Load(object sender, EventArgs e)
{
    m spel2 = new Spel();
    try
    {
       m spel2.ServerInstance = 2;
       m spel2.Initialize();
       m spel2.Connect(6);
       m spel2.Project = "c:\\EpsonRC80\\Projects\\Demo2\\Demo2.sprj";
       m spel2.EventReceived += new
Spel.EventReceivedEventHandler(m spel2 EventReceived);
   }
    catch (SpelException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
public void m spel2 EventReceived (object sender, SpelEventArgs e)
{
    // 機器人控制器2
}
private void frmDemo2 FormClosed(object sender, FormClosedEventArgs e)
{
    m spel2.Dispose();
```