

リファレンスマニュアル

A C a t R S L
V e r . 1 . 2 0

目次

第1章 関数一覧

第2章 関数仕様

2-1 ライブラリ使用方法	2-1
2-1-1 アプリケーション開始	2-1
2-1-2 アプリケーション開始 (DC モード有効のスレーブを使用する場合)	2-2
2-1-3 ネットワーク情報ファイル (config.xml) 生成	2-3
2-1-4 アプリケーション終了	2-4
2-2 ライブラリ関数	2-5
2-3 マスタアクセス関数	2-13
2-4 マスタ通知	2-63
2-5 汎用 PDO/SDO アクセス関数	2-99
2-6 CiA402 ドライブプロファイル制御関数	2-106
2-7 デジタル入出力ユニット関数	2-142
2-8 アナログ入出力ユニット関数	2-153
2-9 モーションコントローラユニット関数	2-170
2-10 エンコーダユニット関数	2-179
2-11 SIO ゲートウェイユニット関数	2-194
2-12 A-Link ゲートウェイユニット関数	2-213
2-13 Modbus ゲートウェイユニット関数	2-232
2-14 CUnet ゲートウェイユニット関数	2-269
2-14-1 Utility	2-269
2-14-2 Open/Close	2-271
2-14-3 メール	2-283
2-14-4 レジスタアクセス	2-288
2-14-5 レジスタアクセス	2-294

2-15 付録	2-303
2-15-1 通知コード	2-303
2-15-2 EtherCAT エラー通知コード	2-304
2-15-3 エラーコード	2-305
2-15-4 EtherCAT マスタエラーコード	2-306
2-15-5 EtherCAT 用語説明	2-313

第1章 関数一覧

1) ライブラリ関数

関 数	機 能
ACat_Create()	ライブラリを初期化します
ACat_Destroy()	ライブラリを終了します
ACat_Make_ENI()	ネットワーク情報ファイル(config.xml)を生成します
ACat_Start()	通信を開始します
ACat_StartEx()	DCを有効にし、通信を開始します
ACat_Stop()	通信を停止します
ACat_Dllname()	ライブラリ名称を取得します
ACat_Version()	バージョンを取得します

2) マスタ制御関数

関 数	機 能
ACat_Mst_Start()	通信を開始します
ACat_Mst_Stop()	通信を停止します
ACat_Mst_Get_State()	マスタの状態を取得します
ACat_Mst_Set_State()	マスタの状態を設定します
ACat_Mst_Get_Slave_Count()	マスタのスレーブ数を取得します
ACat_Mst_Get_Slave_Info()	マスタのスレーブ情報を取得します
ACat_Mst_Dc_Start()	マスタの DC モードを有効にします
ACat_Mst_Dc_Stop()	マスタの DC モードを無効にします
ACat_Mst_Dc_Config()	マスタの DC コンフィグを設定します
ACat_Mst_RxPdo_Read()	マスタの入力 PDO を読み込みます
ACat_Mst_TxPdo_Read()	マスタの出力 PDO を読み込みます
ACat_Mst_TxPdo_Write()	マスタの出力 PDO を書き込みます
ACat_Mst_Set_Notification_Hook()	マスタ通知関数の登録を行います
ACat_Mst_Set_CycleEvent_Hook()	マスタ周期関数の登録を行います
ACat_Mst_Get_Link_Connected()	ポートのリンク状態を取得します
ACat_Mst_Get_Link_Connected_Main()	ポート(メイン)のリンク状態を取得します
ACat_Mst_Get_Link_Connected_Sub()	ポート(サブ)のリンク状態を取得します
ACat_Mst_Get_Connected_Slave_Count()	通信しているスレーブ数を取得します
ACat_Mst_Get_Connected_Slave_Count_Main()	ポート(メイン)の通信スレーブ数を取得します
ACat_Mst_Get_Connected_Slave_Count_Sub()	ポート(サブ)の通信スレーブ数を取得します
ACat_Mst_Get_Configured_Slave_Count()	コンフィグレーションしているスレーブ数を取得します
ACat_Slv_Read_Input()	スレーブの Input データを取得します
ACat_Slv_Read_Output()	スレーブの Output データを取得します
ACat_Slv_Write_Output()	スレーブの Output データを出力します
ACat_Slv_Get_State()	スレーブの State を取得します
ACat_Slv_Set_State()	スレーブの State を設定します
ACat_Slv_Get_Info()	スレーブ情報を取得します
ACat_Slv_Get_Address()	フィジカルアドレスを取得します
ACat_Slv_Get_SlaveID()	スレーブアドレスを取得します
ACat_Slv_Reset()	スレーブのソフトウェアリセットを行います
ACat_Slv_Sdo_Download()	スレーブの SDO ダウンロードします
ACat_Slv_Sdo_DownloadReq()	スレーブの SDO ダウンロードを開始し、すぐに戻ります
ACat_Slv_Sdo_Upload()	スレーブの SDO アップロードします
ACat_Slv_Sdo_UploadReq()	スレーブの SDO アップロードを開始し、すぐに戻ります
ACat_Slv_Foe_Download()	スレーブの FoE ダウンロードします
ACat_Slv_Foe_DownloadReq()	スレーブの FoE ダウンロードを開始し、すぐに戻ります
ACat_Slv_Foe_Upload()	スレーブの FoE アップロードします
ACat_Slv_Foe_UploadReq()	スレーブの FoE アップロードを開始し、すぐに戻ります
ACat_Slv_Read_Register()	スレーブの ASIC のレジスタを読み込みます
ACat_Slv_Write_Register()	スレーブの ASIC のレジスタを書き込みます
ACat_Slv_Read_Eeprom()	スレーブの EEPROM を読み出します
ACat_Slv_Write_Eeprom()	スレーブの EEPROM を書き込みます
ACat_Slv_Reload_Eeprom()	スレーブの EEPROM をリロードします
ACat_Slv_Assign_Eeprom()	スレーブの EEPROM アクセスを設定します
ACat_Slv_Active_Eeprom()	スレーブの EEPROM アクセス設定を読み出します

3) 汎用 PDO/SDO 制御関数

関 数	機 能
ACat_Gene_Create()	初期化します
ACat_Gene_Destroy()	終了します
ACat_Gene_Read_Input()	入力 PDO を読み込みます
ACat_Gene_Read_Output()	出力 PDO を読み込みます
ACat_Gene_Write_Output()	出力 PDO を書き込みます
ACat_Gene_Sdo_Download()	SDO ダウンロードを行います
ACat_Gene_Sdo_Upload()	SDO アップロードを行います

4) CiA402 ドライブプロファイル制御関数

関 数	機 能
ACat_Motion_Create()	CiA402 準拠スレーブを初期化します
ACat_Motion_Destroy()	CiA402 準拠スレーブを終了します
ACat_Motion_ServoOn()	CiA402 準拠スレーブをサーボ ON します
ACat_Motion_ServoOff()	CiA402 準拠スレーブをサーボ OFF します
ACat_Motion_MoveAbsolute()	CiA402 準拠スレーブを絶対位置制御します
ACat_Motion_MoveRelative()	CiA402 準拠スレーブを相対位置制御します
ACat_Motion_MoveVelocity()	CiA402 準拠スレーブを速度制御します
ACat_Motion_MoveTorque()	CiA402 準拠スレーブをトルク制御します
ACat_Motion_MoveParameter()	CiA402 準拠スレーブの動作パラメータを設定します
ACat_Motion_Homing()	CiA402 準拠スレーブを原点復帰動作します
ACat_Motion_HomingParameter()	CiA402 準拠スレーブの原点復帰パラメータを設定します
ACat_Motion_Stop()	CiA402 準拠スレーブを停止します
ACat_Motion_Reset()	CiA402 準拠スレーブをエラーリセットします
ACat_Motion_ReadTargetReached()	CiA402 準拠スレーブの目標到達を取得します
ACat_Motion_ReadHomingEnd()	CiA402 準拠スレーブの原点復帰完了を取得します
ACat_Motion_ReadAxisError()	CiA402 準拠スレーブのエラーコードを取得します
ACat_Motion_ReadActualPosition()	CiA402 準拠スレーブの現在位置を取得します
ACat_Motion_ReadActualVelocity()	CiA402 準拠スレーブの現在速度を取得します
ACat_Motion_ReadActualTorque()	CiA402 準拠スレーブの現在トルクを取得します
ACat_Motion_ReadTargetPosition()	CiA402 準拠スレーブの目標位置を取得します。
ACat_Motion_ReadTargetVelocity()	CiA402 準拠スレーブの目標速度を取得します
ACat_Motion_ReadTargetTorque()	CiA402 準拠スレーブの目標トルクを取得します
ACat_Motion_ReadInputData()	CiA402 準拠スレーブのデジタル入力を取得します
ACat_Motion_WriteOutputData()	CiA402 準拠スレーブのデジタル出力を書込みます
ACat_Motion_SetParameter()	CiA402 準拠スレーブのパラメータを書込みます
ACat_Motion_GetParameter()	CiA402 準拠スレーブのパラメータを取得します
ACat_Motion_WriteControlword()	CiA402 準拠スレーブのコントロールワードを書込みます
ACat_Motion_ReadStatusword()	CiA402 準拠スレーブのステータスワードを取得します
ACat_Motion_WriteShutdown()	CiA402 準拠スレーブに「Shutdown」を書込みます
ACat_Motion_WriteSwitchOn()	CiA402 準拠スレーブに「SwitchOn」を書込みます
ACat_Motion_WriteEnableOperation()	CiA402 準拠スレーブに「Enable Operation」を書込みます
ACat_Motion_WriteDisableOperation()	CiA402 準拠スレーブに「Disable Operation」を書込みます
ACat_Motion_WriteQuickStop()	CiA402 準拠スレーブに「QuickStop」を書込みます
ACat_Motion_WriteDisableVoltage()	CiA402 準拠スレーブに「DisableVoltage」を書込みます
ACat_Motion_WriteFaultReset()	CiA402 準拠スレーブに「FaultReset」を書込みます
ACat_Motion_ReadFSASState()	CiA402 準拠スレーブの FSA ステータスを取得します

5) デジタル入出力ユニット制御関数

ユニットタイプ別記号	関 数	機 能
DIO	ACat_Dio_Create()	ユニットを初期化します
	ACat_Dio_Destroy()	ユニットを終了します
	ACat_Dio_Open()	ユニットをオープンします
	ACat_Dio_Close()	ユニットをクローズします
	ACat_Dio_Read()	ユニットの Input データを取得します
	ACat_Dio_Write()	ユニットに Output データを出力します
	ACat_Dio_GetParam()	ユニットの設定パラメータを取得します
	ACat_Dio_SetParam()	ユニットに設定パラメータを出力します
	ACat_Dio_SaveParam()	ユニットに設定パラメータを保存します
	ACat_Dio_LoadParam()	ユニットの設定パラメータを初期化します
ACat_Dio_SoftReset()	ユニットをソフトウェアリセットします	

6) アナログ入出力ユニット制御関数

ユニットタイプ別記号	関 数	機 能
AIO	ACat_Aio_Create()	ユニットを初期化します
	ACat_Aio_Destroy()	ユニットを終了します
	ACat_Aio_Open()	ユニットをオープンします
	ACat_Aio_Close()	ユニットをクローズします
	ACat_Aio_Read()	ユニットの Input データを取得します
	ACat_Aio_Write()	ユニットに Output データを出力します
	ACat_Aio_AD_GetParam()	ユニットの設定パラメータを取得します
	ACat_Aio_AD_SetMode()	ユニットのモードを設定します
	ACat_Aio_AD_SetParam()	ユニットに設定パラメータを設定します
	ACat_Aio_AD_SetCalib()	ユニットのキャリブレーションを行います
	ACat_Aio_DA_GetParam()	ユニットの設定パラメータを取得します
	ACat_Aio_DA_SetMode()	ユニットのモードを設定します
	ACat_Aio_DA_SetParam()	ユニットに設定パラメータを出力します
	ACat_Aio_DA_SetCalib()	ユニットのキャリブレーションを行います
ACat_Aio_SoftReset()	ユニットの通信状態を取得します	

7) モーションコントローラユニット制御関数

ユニットタイプ別記号	関数	機能
AXIS	ACat_Axis_Create()	ユニットを初期化します
	ACat_Axis_Destroy()	ユニットを終了します
	ACat_Axis_Open()	ユニットをオープンします
	ACat_Axis_Close()	ユニットをクローズします
	ACat_Axis_Read()	ユニットの Input データを取得します
	ACat_Axis_Write()	ユニットに Output データを出力します
	ACat_Axis_GetParam()	ユニットの設定パラメータを取得します
	ACat_Axis_SetParam()	ユニットに設定パラメータを設定します
	ACat_Axis_SoftReset()	ユニットをソフトウェアリセットします

8) エンコーダユニット制御関数

ユニットタイプ別記号	関数	機能
ENC	ACat_Enc_Create()	ユニットを初期化します
	ACat_Enc_Destroy()	ユニットを終了します
	ACat_Enc_Open()	ユニットをオープンします
	ACat_Enc_Close()	ユニットをクローズします
	ACat_Enc_Read()	ユニットの Input データを取得します
	ACat_Enc_Write()	ユニットに Output データを出力します
	ACat_Enc_GetParam()	ユニットの設定パラメータを取得します
	ACat_Enc_SetDirection()	ユニットに回転方向を設定します
	ACat_Enc_SetEdgeEvaluation()	ユニットに入力方式を設定します
	ACat_Enc_SetMaxRingCount()	ユニットにリングカウンタ最大値を設定します
	ACat_Enc_SaveParam()	ユニットに設定パラメータを保存します
	ACat_Enc_LoadParam()	ユニットの設定パラメータを初期化します
	ACat_Enc_SoftReset()	ユニットをソフトウェアリセットします

9) SIO ゲートウェイユニット制御関数

ユニットタイプ別記号	関数	機能
SIO	ACat_Sio_Create()	ユニットを初期化します
	ACat_Sio_Destroy()	ユニットを終了します
	ACat_Sio_Open()	ユニットをオープンします
	ACat_Sio_Close()	ユニットをクローズします
	ACat_Sio_Read()	ユニットの Input データを取得します
	ACat_Sio_Write()	ユニットに Output データを出力します
	ACat_Sio_GetCommInfo()	ユニットの COMMINFO パラメータを取得します
	ACat_Sio_SetCommInfo()	ユニットに COMMINFO パラメータを設定します
	ACat_Sio_GetCommState()	ユニットの通信状態を取得します
	ACat_Sio_ClearError()	ユニットの通信エラーをクリアします
	ACat_Sio_PurgeComm()	ユニットの入出力バッファを破棄します
	ACat_Sio_SaveParam()	ユニットに設定パラメータを保存します
	ACat_Sio_LoadParam()	ユニットの設定パラメータを初期化します
	ACat_Sio_SoftReset()	ユニットをソフトウェアリセットします

10) A-Link ゲートウェイユニット制御関数

ユニットタイプ別記号	関数	機能
ALINK	ACat_Alink_Create()	ユニットを初期化します
	ACat_Alink_Destroy()	ユニットを終了します
	ACat_Alink_Open()	ユニットをオープンします
	ACat_Alink_Close()	ユニットをクローズします
	ACat_Alink_Start()	ユニットを通信開始します
	ACat_Alink_Stop()	ユニットを通信停止します
	ACat_Alink_Read()	ユニットの Input データを取得します
	ACat_Alink_Write()	ユニットに Output データを出力します
	ACat_Alink_ReadChk2Occur()	ユニットの CHK2 発生状況を取得します
	ACat_Alink_GetStatus()	ユニットのステータスを取得します
	ACat_Alink_GetSysSts()	ユニットのシステムステータスを取得します
	ACat_Alink_GetChk1Count()	ユニットの CHK1 発生回数を取得します
	ACat_Alink_GetChk2Count()	ユニットの CHK2 発生回数を取得します
	ACat_Alink_ClrChk1Count()	ユニットの CHK1 発生回数をクリアします
	ACat_Alink_ClrChk2Count()	ユニットの CHK2 発生回数をクリアします
	ACat_Alink_GetCommParam()	ユニットの通信設定を取得します
	ACat_Alink_SetCommParam()	ユニットの通信設定を行います
	ACat_Alink_SoftReset()	ユニットをソフトウェアリセットします

1.1) Modbus ゲートウェイユニット制御関数

ユニットタイプ別記号	関 数	機 能
MBG	ACat_Mbg_Create()	ユニットを初期化します
	ACat_Mbg_Destroy()	ユニットを終了します
	ACat_Mbg_Open()	ユニットをオープンします
	ACat_Mbg_Close()	ユニットをクローズします
	ACat_Mbg_ReadByte()	ユニットの入力データを Byte 単位で読み込みます
	ACat_Mbg_WriteByte()	ユニットの出力データを Byte 単位で書き込みます
	ACat_Mbg_ReadWord()	ユニットの入力データを Word 単位で読み込みます
	ACat_Mbg_WriteWord()	ユニットに出力データを Word 単位で書き込みます
	ACat_Mbg_Start()	ターゲット機器との通信を開始します
	ACat_Mbg_Stop()	ターゲット機器との通信を停止します
	ACat_Mbg_GetTargetInfo()	ユニットのターゲット機器設定を取得します
	ACat_Mbg_SetTargetInfo()	ユニットにターゲット機器設定を設定します
	ACat_Mbg_GetCommInfo()	ユニットのシリアル通信設定を取得します
	ACat_Mbg_SetCommInfo()	ユニットにシリアル通信設定を設定します
	ACat_Mbg_GetMonCmd()	ユニットのモニタデータコマンドを取得します
	ACat_Mbg_SetMonCmd()	ユニットにモニタデータコマンドを設定します
	ACat_Mbg_GetSpotReqCmd()	ユニットの即時要求データコマンドを取得します
	ACat_Mbg_SetSpotReqCmd()	ユニットに即時要求データコマンドを設定します
	ACat_Mbg_GetManReqCmd()	ユニットの手動要求データコマンドを取得します
	ACat_Mbg_SetManReqCmd()	ユニットに手動要求データコマンドを設定します
	ACat_Mbg_GetMonCmdErr()	ユニットのモニタエラーを取得します
	ACat_Mbg_GetSpotReqCmdErr()	ユニットの即時要求エラーを取得します
	ACat_Mbg_GetManReqCmdErr()	ユニットの手動要求エラーを取得します
	ACat_Mbg_GetSpotReqCmdRes()	ユニットの即時要求レスポンスを取得します
	ACat_Mbg_GetManReqCmdRes()	ユニットの手動要求レスポンスを取得します
	ACat_Mbg_ClrMonCmdErr()	ユニットのモニタエラーをクリアします
	ACat_Mbg_ClrSpotReqCmdErr()	ユニットの即時要求エラーをクリアします
	ACat_Mbg_ClrManReqCmdErr()	ユニットの手動要求エラーをクリアします
	ACat_Mbg_ClrSpotReqCmdRes()	ユニットの即時要求レスポンスをクリアします
	ACat_Mbg_ClrManReqCmdRes()	ユニットの手動要求レスポンスをクリアします
	ACat_Mbg_GetErrorStatus()	ユニットのエラー状態を取得します
	ACat_Mbg_GetResponseStatus()	ユニットのレスポンス状態を取得します
	ACat_Mbg_SaveParam()	ユニットにパラメータを保存します
ACat_Mbg_LoadParam()	ユニットのパラメータを初期化します	
ACat_Mbg_SoftReset()	ユニットをソフトウェアリセットします	

1 2) CUnet ゲートウェイユニット制御関数

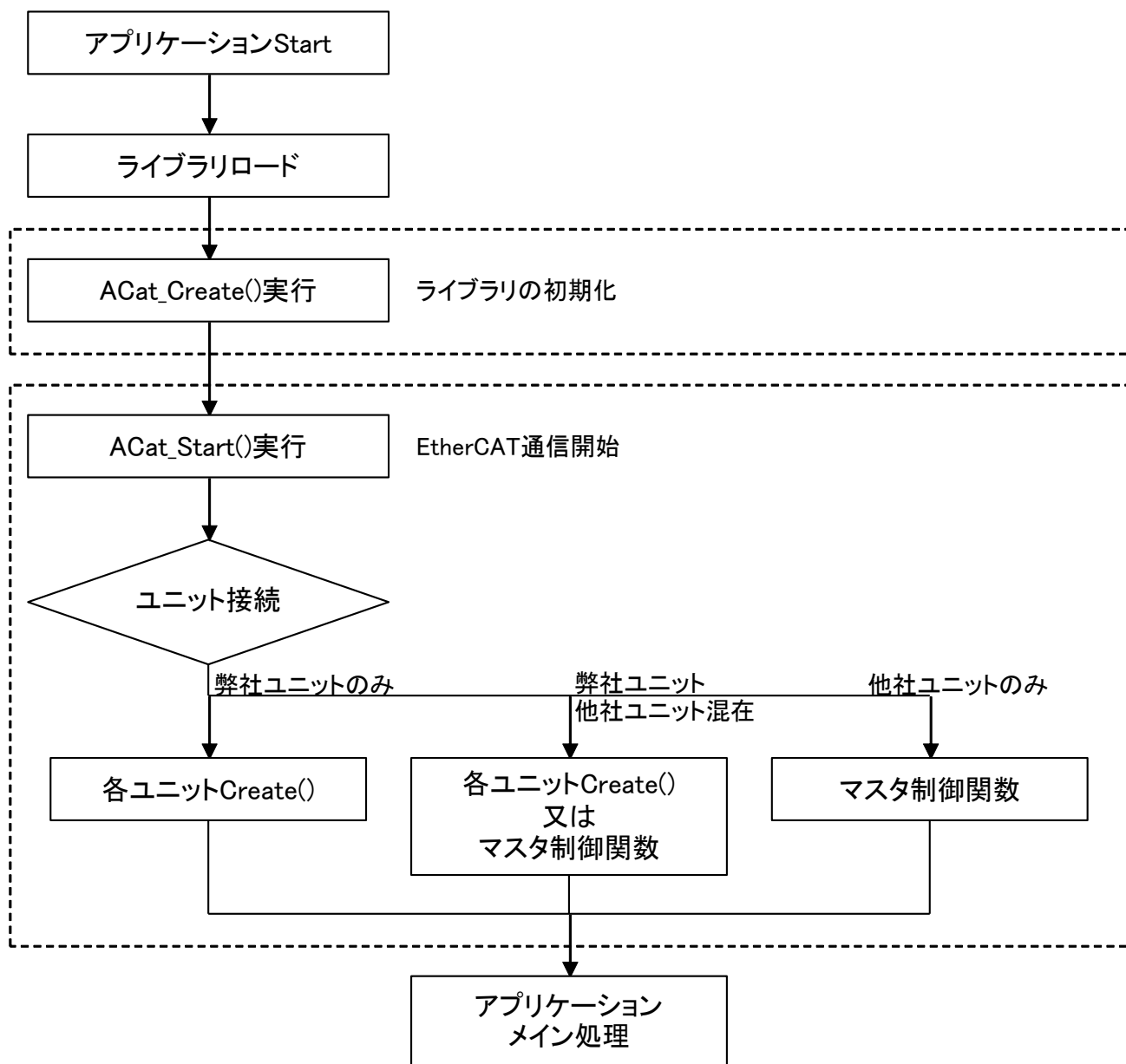
ユニットタイプ別記号	関 数	機 能
CN	ACat_CN_Create()	ユニットを初期化します
	ACat_CN_Destroy()	ユニットを終了します
	ACat_CN_open()	ユニットをオープンします
	ACat_CN_close()	ユニットをクローズします
	ACat_CN_start()	ユニットを通信開始します
	ACat_CN_start_lf()	ユニットを LF モードで通信開始します
	ACat_CN_start_gmm()	ユニットを GMM モードで通信開始します
	ACat_CN_stop()	ユニットを通信停止します
	ACat_CN_reset()	ユニットをリセットします
	ACat_CN_GetCnectStat()	ユニットの接続状態を取得します
	ACat_CN_config()	ユニットの通信設定を行います
	ACat_CN_Status()	ユニットのステータスを取得します
	ACat_CN_Put()	ユニットのグローバルメモリを読み込みます
	ACat_CN_Get()	ユニットのグローバルメモリに書き込みます
	ACat_CN_MailOpen()	ユニットのメールを通信開始します
	ACat_CN_MailClose()	ユニットのメールを通信停止します
	ACat_CN_MailSend()	ユニットからメールを送信します。
	ACat_CN_MailReceive()	ユニットのメールを受信します
	ACat_CN_MailStatus()	ユニットのメール状況を取得します
	ACat_CN_GetReg64()	ユニットのレジスタを 64 ビットサイズで読み込みます
	ACat_CN_GetReg32()	ユニットのレジスタを 32 ビットサイズで読み込みます
	ACat_CN_GetReg16()	ユニットのレジスタを 16 ビットサイズで読み込みます
	ACat_CN_SetReg64()	ユニットのレジスタを 64 ビットサイズで書き込みます
	ACat_CN_SetReg32()	ユニットのレジスタを 32 ビットサイズで書き込みます
	ACat_CN_SetReg16()	ユニットのレジスタを 16 ビットサイズで書き込みます
	ACat_CN_GetMemByte()	ユニットのグローバルメモリを Byte 単位で読み込みます
	ACat_CN_GetMemShort()	ユニットのグローバルメモリを Word 単位で読み込みます
	ACat_CN_GetMemLong()	ユニットのグローバルメモリを Long 単位で読み込みます
	ACat_CN_GetMemDLong()	ユニットのグローバルメモリを DLong 単位で読み込みます
	ACat_CN_SetMemByte()	ユニットのグローバルメモリを Byte 単位で書き込みます
	ACat_CN_SetMemShort()	ユニットのグローバルメモリを Word 単位で書き込みます
	ACat_CN_SetMemLong()	ユニットのグローバルメモリを Long 単位で書き込みます
	ACat_CN_SetMemDLong()	ユニットのグローバルメモリを DLong 単位で書き込みます
ACat_CN_HardReset()	ユニットをハードリセットします	

第2章 関数仕様

2-1 ライブラリ使用方法

2-1-1 アプリケーション開始

ライブラリを使用したアプリケーション開始のフローチャートを以下に示します。

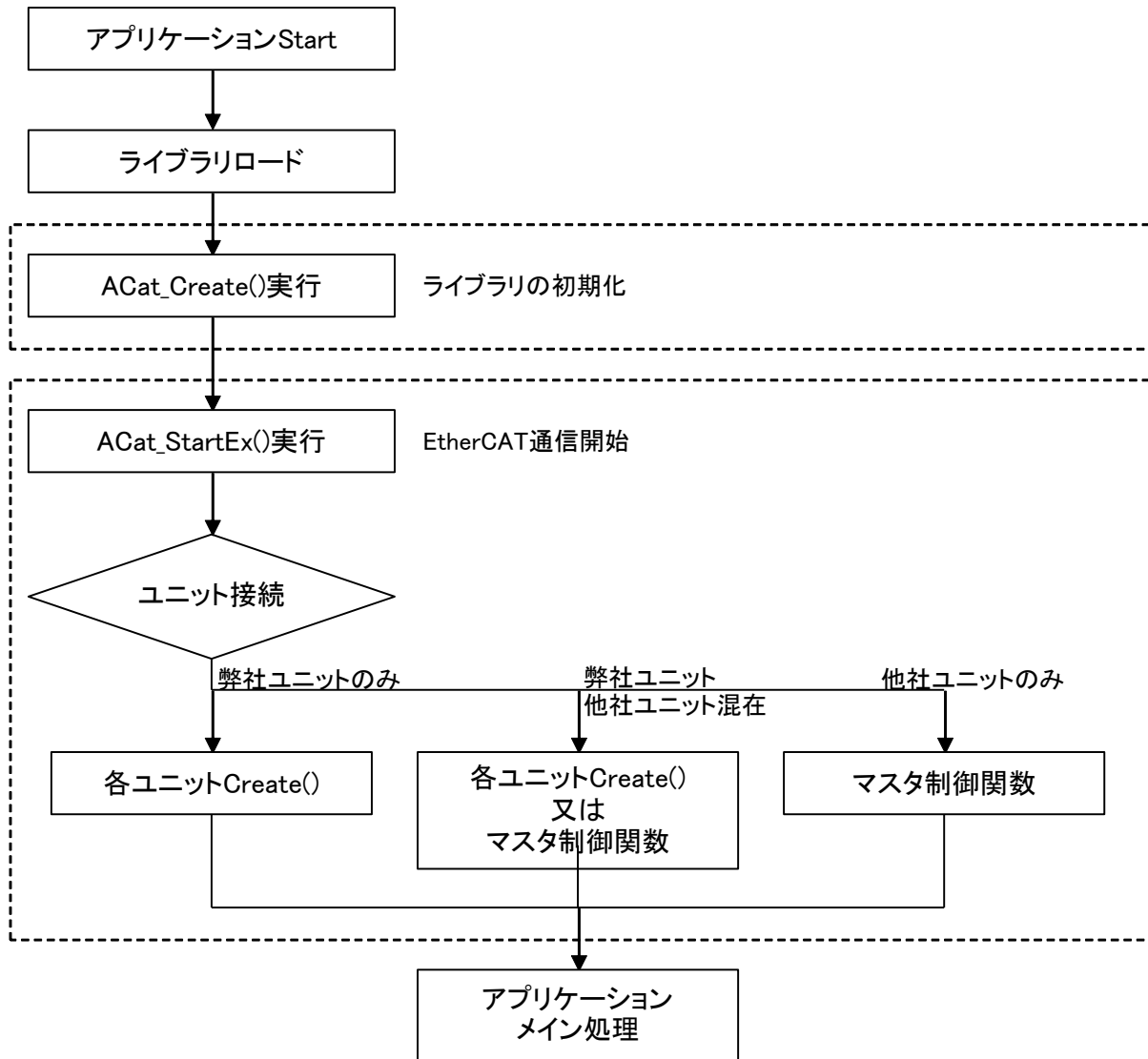


アルゴシステム製品のスレーブユニットを使用する場合は、スレーブユニットの Create() 関数を実行後、スレーブユニットのライブラリ関数を実行する事が出来ます。

他社製品へのアクセスは、マスタ制御関数や汎用 PDO/SDO 関数を使用します。

2-1-2 アプリケーション開始 (DCモード有効のスレーブを使用する場合)

DCモード有効のスレーブが含まれる EtherCAT システム構成でライブラリを使用したアプリケーション開始のフローチャートを以下に示します。

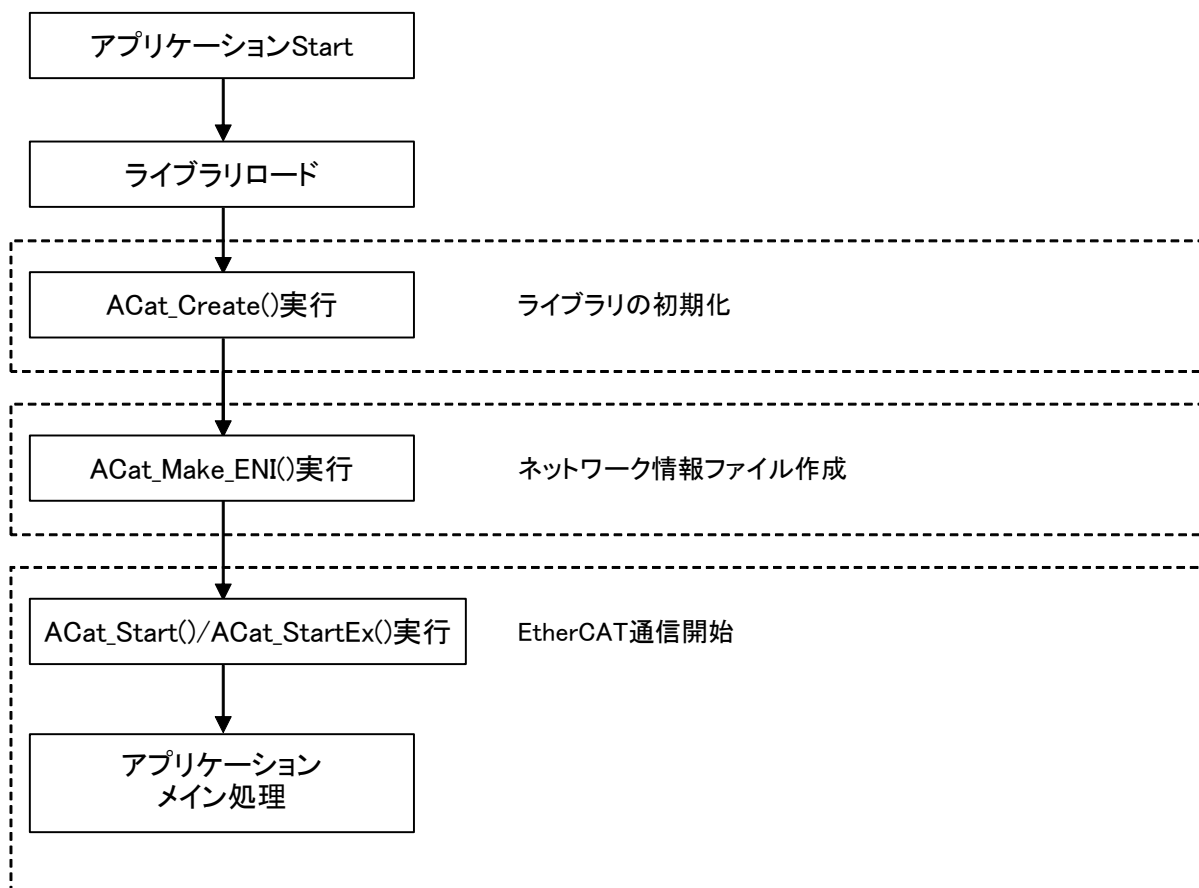


アルゴシステム製品のスレーブユニットを使用する場合は、スレーブユニットの Create() 関数を実行後、スレーブユニットのライブラリ関数を実行する事が出来ます。

他社製品へのアクセスは、マスタ制御関数や汎用 PDO/SDO 関数、CiA402 ドライブプロファイル制御関数を使用します。

2-1-3 ネットワーク情報ファイル(config.xml)生成

ライブラリを使用したネットワーク情報ファイル(config.xml)作成を含むアプリケーション開始のフローチャートを以下に示します。

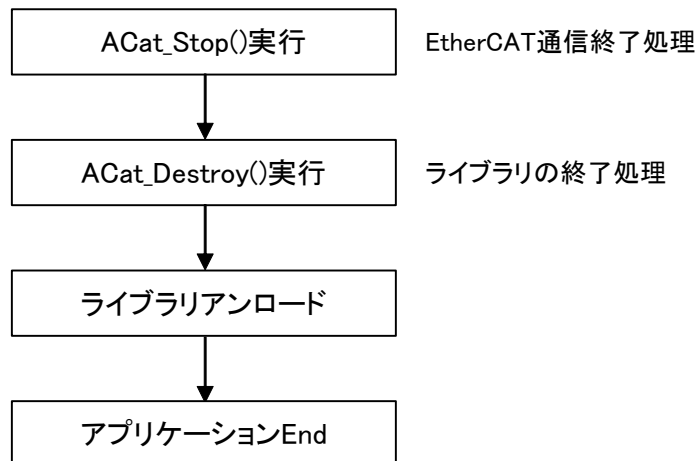


ネットワーク情報ファイル(config.xml)を作成する場合は、スレーブ設定ファイル、システム構成ファイルが別途必要になります。

スレーブ設定ファイル、システム構成ファイルについては、別紙「EtherCAT ネットワーク情報ファイル config.xml 設定マニュアル」を参照してください。

2-1-4 アプリケーション終了

ライブラリを使用したアプリケーション終了のフローチャートを以下に示します。



2-2 ライブラリ関数

ACat_Create 関数

機能	EtherCAT マスタ、ライブラリを初期化します
書式	<code>int ACat_Create(void)</code>
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	EtherCAT マスタ、ライブラリ本体を初期化します。 この関数コール後、ライブラリ内の関数群にアクセス可能になります。 アプリケーションの起動時に必ずコールする必要があります。

ACat_Destroy 関数

機能	EtherCAT マスタ、ライブラリを終了します
書式	<code>int ACat_Destroy(void)</code>
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	EtherCAT マスタ、ライブラリ本体を終了します。 この関数コール後は、ライブラリ内の関数群にアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_Make_ENI 関数

機能	ネットワーク情報ファイル(config.xml)を生成します
書式	int ACat_Make_ENI(int Timeout)
引数	Timeout : タイムアウト値(ミリ秒)
戻り値	ACAT_ER_OK : 正常 ACAT_ER_CREATEENI : ENI ファイル作成失敗(ソケットの生成に失敗) ACAT_ER_CREATEENI1 : ENI ファイル作成失敗(サーバの接続に失敗) ACAT_ER_CREATEENI2 : ENI ファイル作成失敗 (サーバとのデータの送受信に失敗) 上記以外 : エラーコード(エラーコード一覧を参照)
説明	ネットワーク情報ファイル(config.xml)を生成します。 この関数コールは EtherCAT マスタ初期化後、且つ EtherCAT 通信開始前に使用することができます。 EtherCAT 通信中は、使用できませんので注意してください。 ネットワーク情報ファイルは、「C:\Program Files\ALGOSYSTEM\AlgoEtherCAT」内に生成されます。 ACat_Make_ENI の機能の詳細に関しては、別紙「EtherCAT ネットワーク情報ファイル config.xml 設定マニュアル」を参照ください。

ACat_Start 関数

機能	EtherCAT 通信を開始します
書式	int ACat_Start(int Timeout)
引数	Timeout : タイムアウト値(ミリ秒)
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	EtherCAT 通信を開始し、スレーブの構成情報を取得します。 マスタ制御関数以外のライブラリを使用する場合は、本関数で EtherCAT 通信を開始後に使用可能となります。 EtherCAT 通信開始には、ネットワーク情報ファイル(config.xml)の用意が必要です。

ACat_StartEx 関数

機能 ディストリビュートクロックのコンフィグ設定後(※1)、EtherCAT 通信を開始します

書式 int ACat_StartEx(int Timeout, BOOL BusShift)

引数 Timeout : タイムアウト値(ミリ秒)
BusShift : バスシフト設定 [FALSE:無効, TRUE:有効]

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 ディストリビュートクロック(以下、DC)のコンフィグ設定を実行した後、EtherCAT 通信を開始し、スレーブの構成情報を取得します。
接続するスレーブの中に DC モードをサポートしているスレーブが含まれる場合は DC モードを有効にする必要があります。本設定の他にネットワーク情報ファイル(config.xml)にも同様に DC モード有効の設定が必要です。
ネットワーク情報ファイル(config.xml)の DC モード設定方法は、別紙「EtherCAT ネットワーク情報ファイル config.xml 設定マニュアル」を参照ください。
DC のコンフィグ設定はバスシフトの使用の有無を設定できます。バスシフトを有効にすると DC 基準クロックに関するバス時間を調整することができます(※2)。

マスタ制御関数以外のライブラリを使用する場合は、本関数で EtherCAT 通信を開始後に使用可能となります。
EtherCAT 通信開始には、ネットワーク情報ファイル(config.xml)の用意が必要です。

(※1) DC モード搭載ユニットが含まれる場合のみサポートします。
DC モード搭載ユニットが無い場合は、影響ありません。

(※2) バスシフト設定の詳細は、EtherCAT マスタ設定ファイル(ACMst.ini)にて設定できます。
EtherCAT マスタ設定ファイルの設定方法は、別紙「ACMst.ini 設定マニュアル」を参照ください。
通信周期が早すぎますとバスシフトが十分に機能しない場合があります。

ACat_Stop 関数

機能

EtherCAT 通信を停止します

書式

int ACat_Stop(int Timeout)

引数

Timeout : タイムアウト値(ミリ秒)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

EtherCAT 通信を停止します。
この関数コール後は、EtherCAT 通信ができなくなりますので、再度、通信を開始する場合は、ACat_Start 関数、ACat_StartEx 関数をコールしてください。
アプリケーションの終了時に必ずコールする必要があります。

ACat_DIIname 関数

機能	ライブラリ名称を取得します
書式	<code>void ACat_DIIname(char *DIIname)</code>
引数	DIIname : ライブラリ名称を格納するポインタ
戻り値	なし
説明	ライブラリ名称を取得します。 ライブラリ名称を格納するメモリは関数呼び出し側で確保する必要があります (10 バイト以上)

ACat_Version 関数

機能	ライブラリのバージョンを取得します
書式	void ACat_Version(char *Version)
引数	Version : ライブラリバージョンを格納するポインタ
戻り値	なし
説明	ライブラリのバージョン番号を取得します。 ライブラリバージョンを格納するメモリは関数呼び出し側で確保する必要があります (8 バイト以上)

2-3 マスタアクセス関数

ACat_Mst_Start 関数

機能 EtherCAT マスタの通信を開始します

書式 int ACat_Mst_Start(char *ConfigFile, int Timeout)

引数 ConfigFile : ネットワーク情報ファイル名
Timeout : タイムアウト値 (msec)

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 EtherCAT 通信を開始します。
マスタアクセス関数のみを使用して、EtherCAT を制御する場合は、本関数のコールが必要となります。
マスタアクセス関数以外の制御関数を使用する場合は、ACat_Start() / ACat_StartEx() を使用して通信開始してください。
EtherCAT 通信開始には、ネットワーク情報ファイル(config.xml)の用意が必要です。
タイムアウトに0[msec]を指定して通信開始しますと、ESM は[INIT]の状態では通信を行います。

ACat_Mst_Stop 関数

機能 EtherCAT 通信を停止します

書式 int ACat_Mst_Stop(int Timeout)

引数 Timeout : タイムアウト値 (msec)

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 EtherCAT 通信を停止します。
この関数コール後は、EtherCAT 通信ができなくなりますので、再度、通信を開始する場合は、ACat_Mst_Start 関数をコールしてください。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Mst_Get_State 関数

機能 EtherCAT マスタの ESM を取得します

書式 int ACat_Mst_Get_State(int *pState)

引数 pState : ESM を格納するポインタ

EtherCAT State Machine (ESM)	内容
ACAT_STATE_UNKNOWN	0: State [UNKNOWN]
ACAT_STATE_INIT	1: State [INIT]
ACAT_STATE_BOOTSTRAP	3: State [BOOTSTRAP]
ACAT_STATE_PREOP	2: State [PREOP]
ACAT_STATE_SAFEOP	4: State [SAFEOP]
ACAT_STATE_OP	8: State [OP]

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 EtherCAT マスタの ESM を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。
ESM は [UNKNOWN→INIT→PRE-OP→SAFE-OP→OP→Bootstrap] の 6 種類の状態があります。
ESM の詳細情報 EtherCAT Technology Group (ETG) より入手できます「EtherCAT Specification」を参照ください。

ACat_Mst_Set_State 関数

機能 EtherCAT マスタの ESM を設定します

書式 int ACat_Mst_Set_State(int State, int Timeout)

引数
 State : 変更する ESM
 Timeout : タイムアウト時間

EtherCAT State Machine (ESM)	内容
ACAT_STATE_UNKNOWN	0: State [UNKNOWN]
ACAT_STATE_INIT	1: State [INIT]
ACAT_STATE_BOOTSTRAP	3: State [BOOTSTRAP]
ACAT_STATE_PREOP	2: State [PREOP]
ACAT_STATE_SAFEOP	4: State [SAFEOP]
ACAT_STATE_OP	8: State [OP]

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明
 EtherCAT マスタの ESM を設定します。
 この関数コールは EtherCAT 通信開始後しか使用できません。
 ESM は [UNKNOWN→INIT→PRE-OP→SAFE-OP→OP→Bootstrap] の 6 種類の状態があります。
 ESM の詳細情報は EtherCAT Technology Group (ETG) より入手できます「EtherCAT Specification」を参照ください。

ACat_Mst_Get_Slave_Count 関数

- 機能** スレーブ数を取得します
- 書式** int ACat_Mst_Get_Slave_Count(int *SlaveCount)
- 引数** SlaveCount : スレーブ数を格納するポインタ
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** 現在、EtherCAT 通信を行っているスレーブ数を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。
EtherCAT マスタの ESM はどの状態であっても構いません。

ACat_Mst_Get_Slave_Info 関数

機能 スレーブ情報を取得します

書式 int ACat_Mst_Get_Slave_Info(int SlaveInstance, ACAT_SLAVE_INFO *pSlaveInfo)

引数
 SlaveInstance : スレーブ番号[0~]
 pSlaveInfo : スレーブ情報を格納する構造体ポインタ

スレーブ情報

```
typedef struct {
    BYTE   DeviceName[DEVICE_NAMESIZE];
    DWORD  VendorID;
    DWORD  ProductCode;
    DWORD  RevisionNo;
    DWORD  SerialNo;
    WORD   PortState;
    BOOL   DcSupport;
    WORD   AliasAddress;
    WORD   PhysAddress;
    DWORD  PdInputOffset;
    DWORD  PdInputSize;
    DWORD  PdOutputOffset;
    DWORD  PdOutputSize;
} ACAT_SLAVE_INFO;
```

DeviceName : スレーブデバイス名称
 VendorID : ベンダーID
 ProductCode : プロダクトコード
 RevisionNo : レビジョン番号
 SerialNo : シリアル番号
 PortState : ステータス
 DcSupport : DC サポート [FALSE:無効, TRUE:有効]
 AliasAddress : スレーブアドレス
 PhysAddress : フィジカルアドレス
 PdInputOffset : PDO 入力オフセット
 PdInputSize : PDO 入力サイズ
 PdOutputOffset : PDO 出力オフセット
 PdOutputSize : PDO 出力サイズ

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

ネットワーク情報ファイル(config.xml)にて設定しているスレーブ情報を取得します。
スレーブ番号は0からの通し番号です。
ACat_Mst_Get_Slave_Count 関数にて取得したスレーブ数までスレーブ情報が取得できます。
この関数コールはEtherCAT 通信開始後しか使用できません。

ACat_Mst_Dc_Start 関数

機能 ディストリビュートクロックを有効にします。(*1)

書式 int ACat_Mst_Dc_Start(void)

引数 なし

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 ディストリビュートクロック(以下、DC)を有効に設定します。
接続するスレーブにDCモードをサポートしているスレーブが含まれる場合はDCモードを有効にする必要があります。
本設定の他にネットワーク情報ファイル(config.xml)にも同様にDCモード有効の設定が必要です。
ネットワーク情報ファイル(config.xml)のDCモード設定方法は、別紙「EtherCAT ネットワーク情報ファイル config.xml 設定マニュアル」を参照ください。

(*1) DCモード搭載ユニットが含まれる時のみサポート

(*1) ネットワーク情報ファイル(config.xml)にもDCモード有効の設定が必要

ACat_Mst_Dc_Stop 関数

機能	ディストリビュートクロックを無効にします。(*1)
書式	int ACat_Mst_Dc_Stop(void)
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	<p>ディストリビュートクロック(以下、DC)を無効に設定します。 接続するスレーブに DC モードをサポートしているスレーブが含まれる場合は DC モードを有効にする必要があります。 本設定の他にネットワーク情報ファイル(config.xml)にも同様に DC モード有効の設定が必要です。 ネットワーク情報ファイル(config.xml)の DC モード設定方法は、別紙「EtherCAT ネットワーク情報ファイル config.xml 設定マニュアル」を参照ください。</p>

(*1) DC モード搭載ユニットが含まれる時のみサポート

(*1) ネットワーク情報ファイル(config.xml)にも DC モード有効の設定が必要

ACat_Mst_Dc_Config 関数

機能	ディストリビュートクロックのコンフィグ設定をします。(※1)
書式	int ACat_Mst_Dc_Config(BOOL BusShift)
引数	BusShift : バスシフト設定 [FALSE:無効, TRUE:有効]
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

ディストリビュートクロック(以下、DC)のコンフィグ設定をします。
接続するスレーブの中に DC モードをサポートしているスレーブが含まれる場合は DC モードを有効にする必要があります。本設定の他にネットワーク情報ファイル(config.xml)にも同様に DC モード有効の設定が必要です。
ネットワーク情報ファイル(config.xml)の DC モード設定方法は、別紙「EtherCAT ネットワーク情報ファイル config.xml 設定マニュアル」を参照ください。
DC のコンフィグ設定はバスシフトの使用の有無を設定できます。バスシフトを有効にすると DC 基準クロックに関するバス時間を調整することができます(※2)。
この関数コールは EtherCAT 通信の ESM が [INIT] の時のみしか使用できません。

(※1) DC モード搭載ユニットが含まれる場合のみサポートします。

DC モード搭載ユニットが無い場合は、影響ありません。

(※2) バスシフト設定の詳細は、EtherCAT マスタ設定ファイル(ACMst.ini)にて設定できます。

EtherCAT マスタ設定ファイルの設定方法は、別紙「ACMst.ini 設定マニュアル」を参照ください。

通信周期が早すぎますとバスシフトが十分に機能しない場合があります。

ACat_Mst_RxPdo_Read 関数

機能

入力 PDO を読み込みます

書式

```
int ACat_Mst_RxPdo_Read(int Offset, int Length, char *pBuffer)
```

引数

Offset : PDO オフセット
Length : PDO データサイズ
pBuffer : PDO データを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

入力 PDO (EtherCAT マスタからスレーブへのデータ転送用 PDO) から PDO オフセット、PDO データサイズで指定したデータを読み込みます。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Mst_TxPdo_Read 関数

機能

出力 PDO を読み込みます

書式

```
int ACat_Mst_TxPdo_Read(int Offset, int Length, char *pBuffer)
```

引数

Offset : PDO オフセット
Length : PDO データサイズ
pBuffer : PDO データを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

出力 PDO(スレーブから EtherCAT マスタへのデータ転送用 PDO)から PDO オフセット、PDO データサイズで指定したデータを読み込みます。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Mst_TxPdo_Write 関数

機能 出力 PDO に書込みます

書式 int ACat_Mst_TxPdo_Write(int Offset, int Length, char *pBuffer)

引数

Offset	:	PDO オフセット
Length	:	PDO データサイズ
pBuffer	:	PDO データを格納するポインタ

戻り値

ACAT_ER_OK	:	正常
ACAT_ER_OK 以外	:	エラーコード(エラーコード一覧を参照)

説明 出力 PDO(スレーブから EtherCAT マスタへのデータ転送用 PDO)に PDO オフセット、PDO データサイズで指定したデータを書込みます。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Mst_Set_Notification_Hook 関数

機能 EtherCAT マスタからの通知割込みのイベントハンドラを設定します

書式 `int ACat_Mst_Set_Notification_Hook (ACAT_NOTIFICATION_HANDLER NotHandler,
void *pArg)`

引数 NotHandler : 通知割込み処理ユーザー関数
pArg : ユーザーデータのポインタ

通知割込み処理ユーザー関数

```
typedef BOOL (*ACAT_NOTIFICATION_HANDLER) (  
    void *pInst,  
    DWORD Code,  
    void *pJobData  
);
```

* 詳細は、ACAT_NOTIFICATION_HANDLER 関数を参照してください。

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 EtherCAT マスタから通知割込みが発生した時にコールされるイベントハンドラの設定を行います。
イベントハンドラには引数に通知コード(Code)と通知パラメータ(pJobData)が渡されます。
割込み設定は EtherCAT 通信開始とともに有効となります。通信通信中は設定を行うことが出来ません。通信を終了させてから設定を行う必要があります。
割込み設定が正常に行われると、割込み発生時に割込み処理ユーザー関数が呼び出される様になります。

ACAT_NOTIFICATION_HANDLER 関数

機能

通知割込み処理ユーザー関数

書式

```
BOOL ACAT_NOTIFICATION_HANDLER (  
    void *pInst,  
    DWORD Code,  
    void *pJobData  
);
```

引数

pInst : ユーザーデータを格納したポインタ
Code : 通知コード、通知エラーコード
pJobData : 通知パラメータを格納したポインタ

※ 通知コード、通知エラーコード、通知パラメータについては、補足を参照

戻り値

TRUE: 正常終了, FALSE: 異常終了

説明

通知割込み発生時にコールバックされる関数です。

ユーザーはこの書式に従って、割込み処理ユーザー関数を作成します。

マスタは、通知する必要があるイベントが発生するたびにこの関数を呼び出します。

このコールバック関数内で、EtherCAT コマンドの送信につながる EtherCAT 関数を呼び出してはいけません。

デッドロック状態を回避するために、コールバック関数はミューテックスセマフォを使用してはいけません。

この関数の呼び出し中に EtherCAT 操作全体がブロックされるため、エラー処理は CPU 時間をあまり使用せず、ブロックする可能性のあるオペレーティングシステム関数を呼び出してはいけません。

補足 通知割込み発生時にコールバックされる通知コード、及び通知エラーコードです。

通知コード(応答)

Notification Code	内容
ACAT_NOTIFY_STATECHANGED	EtherCAT の動作状態 (ESM) が変更された
ACAT_NOTIFY_ETH_LINK_CONNECTED	Ethernet ケーブルが接続された
ACAT_NOTIFY_DC_STATUS	Distributed clocks (DC) が初期化された
ACAT_NOTIFY_DC_SLV_SYNC	DC の同期通信が開始/逸脱した
ACAT_NOTIFY_DCM_SYNC	DCM 同期が発生した
ACAT_NOTIFY_SLAVE_PRESENCE	非表示のスレーブが表示された
ACAT_NOTIFY_SB_STATUS	バススキャンが正常に終了した
ACAT_NOTIFY_SB_MISMATCH	バススキャンの結果が不一致
ACAT_NOTIFY_SB_DUPLICATE_HC_NODE	全く同じユニットが検出された
ACAT_NOTIFY_MBOXRCV	メールボックスを受信した
ACAT_NOTIFY_AUTO_RETURN	自動復帰が発生した
ACAT_NOTIFY_ILLEGAL_CODE	例外コードが検出された。

通知コード(エラー)

Notification Code	内容
ACAT_NOTIFY_ER_CYCCMD_WKC	サイクリック通信で WKC エラーが発生
ACAT_NOTIFY_ER_MASTER_INITCMD_WKC	マスタ初期化で WKC エラーが発生
ACAT_NOTIFY_ER_SLAVE_INITCMD_WKC	スレーブ初期化で WKC エラーが発生
ACAT_NOTIFY_ER_COE_MBXSEND_WKC	CoE の MailBox 送信で WKC エラーが発生
ACAT_NOTIFY_ER_FOE_MBXSEND_WKC	FoE の MailBox 送信で WKC エラーが発生
ACAT_NOTIFY_ER_FRAME_RESPONSE	受信フレームが異常
ACAT_NOTIFY_ER_SLAVE_INITCMD_RESPONSE	スレーブ初期化コマンドで無応答
ACAT_NOTIFY_ER_MASTER_INITCMD_RESPONSE	マスタ初期化コマンドで無応答
ACAT_NOTIFY_ER_MBSLAVE_INITCMD_TIMEOUT	メールボックス初期化コマンドでタイムアウト
ACAT_NOTIFY_ER_NOT_ALL_DEVICES_OPERATIONAL	全デバイスが OP にならない
ACAT_NOTIFY_ER_ETH_LINK_NOT_CONNECTED	Ethernet ケーブルが切断された
ACAT_NOTIFY_ER_STATUS_SLAVE	1 つ以上のスレーブにエラーが発生
ACAT_NOTIFY_ER_SLAVE_STATUS_INFO	スレーブステータスエラー
ACAT_NOTIFY_ER_SLAVE_NOT_ADDRESSABLE	スレーブアドレスエラー
ACAT_NOTIFY_ER_MBSLAVE_COE_SDO_ABORT	メールボックスの SDO アポートが発生
ACAT_NOTIFY_ER_CLIENTREGISTRATION_DROPPED	すでにコンフィグ設定されている
ACAT_NOTIFY_ER_FOE_MBSLAVE	FoE メールボックスのアポートが発生
ACAT_NOTIFY_ER_PDIWATCHDOG	PDI Watchdog がスレーブで発生
ACAT_NOTIFY_ER_SLAVE_NOTSUPPORTED	スレーブはサポートされていない
ACAT_NOTIFY_ER_SLAVE_UNEXPECTED_STATE	スレーブが予期しない状態 (ESM) です
ACAT_NOTIFY_ER_ALL_DEVICES_OPERATIONAL	全デバイスが OP になった
ACAT_NOTIFY_ER_EEPROM_CHECKSUM	EEPROM のチェックサム異常が発生
ACAT_NOTIFY_ER_LINE_CROSSED	ライン交差を検出
ACAT_NOTIFY_ER_FRAMELOSS_AFTER_SLAVE	スレーブ後がフレームロス

ACat_Mst_Set_CycleEvent_Hook 関数

機能 EtherCAT マスタからの通信周期割込みを設定します

書式 `int ACat_Mst_Set_CycleEvent_Hook (ACAT_EVENT_HANDLER EvtHandler,
void *pArg)`

引数 EvtHandler : 周期割込み処理ユーザー関数
pArg : ユーザーデータのポインタ

周期割込み処理ユーザー関数

```
typedef void (*ACAT_EVENT_HANDLER) (  
    void *pInst,  
    QWORD qwStartCount  
);
```

* 詳細は、ACAT_EVENT_HANDLER 関数を参照してください。

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 EtherCAT マスタの通信周期処理から呼び出されるプロセスデータの通信周期割込みの設定を行います。
割込み設定は EtherCAT 通信開始とともに有効となります。
通信通信中は設定を行うことが出来ません。
通信を終了させてから設定を行う必要があります。
割込み設定が正常に行われると、EtherCAT マスタの通信周期毎に割込み処理ユーザー関数が呼び出されるようになります。

ACAT_EVENT_HANDLER 関数

機能

周期割込み処理ユーザー関数

書式

```
void ACAT_EVENT_HANDLER (  
    void *pInst,  
    QWORD qwStartCount  
);
```

引数

pInst : ユーザーデータを格納したポインタ
qwStartCount : 割込み時タイムスタンプカウンタ

戻り値

なし

説明

EtherCAT マスタからプロセスデータ通信周期割込み発生時にコールされる関数です。ユーザーはこの書式に従って、割込み処理ユーザー関数を作成します。受信したすべてのフレームを処理後、割込みが発生し、割込み処理終了後にフレームを送信して、プロセス出力データを更新します。受信処理と送信処理の間に割込み処理が実行されるため、ユーザーは全ての処理が正常に処理できるように通信周期を決定する必要があります。

割込み時タイムスタンプカウンタは、プロセスデータ通信周期毎の Pentium CPU が内部に持つ 64bit のパフォーマンスカウンタレジスタ(タイムスタンプカウンタ)のカウンタ値が格納されます。CPU 速度から処理時間を測定することができます。

ACat_Mst_Get_Link_Connected 関数

- 機能** EtherCAT ポートのリンク状態を取得します
- 書式** `int ACat_Mst_Get_Link_Connected(int *pLinkConnected);`
- 引数** `pLinkConnected` : リンク状態を格納するポインタ (0: 非リンク状態、1: リンク状態)
- 戻り値** `ACAT_ER_OK` : 正常
`ACAT_ER_OK` 以外 : エラーコード(エラーコード一覧を参照)
- 説明** EtherCAT ポートのリンク状態を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Mst_Get_Link_Connected_Main 関数

- 機能** EtherCAT ポート(メイン)のリンク状態を取得します
- 書式** `int ACat_Mst_Get_Link_Connected_Main(int *pLinkConnected);`
- 引数** pLinkConnected : リンク状態を格納するポインタ (0: 非リンク状態、1: リンク状態)
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** EtherCAT メインポートのリンク状態を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。
本関数は Cable Redundancy の機能の為、未サポートとなります。

ACat_Mst_Get_Link_Connected_Sub 関数

- 機能** EtherCAT ポート (サブ) のリンク状態を取得します
- 書式** `int ACat_Mst_Get_Link_Connected_Sub(int *pLinkConnected);`
- 引数** `pLinkConnected` : リンク状態を格納するポインタ (0: 非リンク状態、1: リンク状態)
- 戻り値** `ACAT_ER_OK` : 正常
`ACAT_ER_OK` 以外 : エラーコード (エラーコード一覧を参照)
- 説明** EtherCAT サブポートのリンク状態を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。
本関数は Cable Redundancy の機能の為、未サポートとなります。

ACat_Mst_Get_Connected_Slave_Count 関数

- 機能** 通信しているスレーブ数を取得します
- 書式** int ACat_Mst_Get_Connected_Slave_Count (DWORD* pSlaveCount)
- 引数** pSlaveCount : 取得するスレーブ数を格納するポインタ
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** 通信しているスレーブ数を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Mst_Get_Connected_Slave_Count_Main 関数

- 機能** 通信しているスレーブ数を取得します
- 書式** int ACat_Mst_Get_Connected_Slave_Count_Main(DWORD* pSlaveCount)
- 引数** pSlaveCount : 取得するスレーブ数を格納するポインタ
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** メインポートで通信しているスレーブ数を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。
本関数は Cable Redundancy の機能の為、未サポートとなります。

ACat_Mst_Get_Connected_Slave_Count_Sub 関数

- 機能** 通信しているスレーブ数を取得します
- 書式** int ACat_Mst_Get_Connected_Slave_Count_Sub (DWORD* pSlaveCount)
- 引数** pSlaveCount : 取得するスレーブ数を格納するポインタ
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** サポートで通信しているスレーブ数を取得します。
この関数コールは EtherCAT 通信開始後しか使用できません。
本関数は Cable Redundancy の機能の為、未サポートとなります。

ACat_Mst_Get_Configured_Slave_Count 関数

機能	コンフィギュレーションしているスレーブ数を取得します
書式	int ACat_Mst_Get_Configured_Slave_Count (DWORD* pSlaveCount)
引数	pSlaveCount : 取得するスレーブ数を格納するポインタ
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	コンフィギュレーションしているスレーブ数を取得します。 この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Read_Input 関数

機能

スレーブユニットの入力からデータを取得します

書式

```
int ACat_Slv_Read_Input(int Address, int Offset, int Length, char *pBuffer)
```

引数

Address : フィジカルアドレス
Offset : オフセット
Length : 取得するデータサイズ
pBuffer : 取得するデータを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットの入力からデータを取得します。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
スレーブユニットによって PDO マッピングが異なりますので、詳細は各スレーブユニットの
マニュアルを参照してください(*1)。
この関数コールは EtherCAT 通信開始後しか使用できません。

(*1)オフセット、データサイズを間違えますと異なるスレーブの入力データを取得すること
になりますので、PDO サイズは十分に確認してください。

ACat_Slv_Read_Output 関数

機能

スレーブユニットの出力からデータを取得します

書式

```
int ACat_Slv_Read_Output(int Address, int Offset, int Length, char *pBuffer)
```

引数

Address : フィジカルアドレス
Offset : オフセット
Length : 取得するデータサイズ
pBuffer : 取得するデータを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットの出力からデータを取得します。フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。スレーブユニットによって PDO マッピングが異なりますので、詳細は各スレーブユニットのマニュアルを参照してください(*1)。
この関数コールは EtherCAT 通信開始後しか使用できません。

(*1) オフセット、データサイズを間違えますと異なるスレーブの入力データを取得することになりますので、PDO サイズは十分に確認してください。

ACat_Slv_Write_Output 関数

機能 スレーブユニットの出力にデータを出力します

書式 int ACat_Slv_Write_Output(int Address, int Offset, int Length, char *pBuffer)

引数

Address	: フィジカルアドレス
Offset	: オフセット
Length	: 出力するデータサイズ
pBuffer	: 出力するデータを格納するポインタ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットの指定したエリアに指定したデータを出力します。

フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。スレーブユニットによって出力できるデータサイズ(出力 PDO)が異なりますので、詳細は各スレーブユニットのマニュアルを参照してください。

この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Get_State 関数

機能 スレーブユニットの ESM を取得します

書式 int ACat_Slv_Get_State(int Address, int *pState, int Timeout)

引数

Address	:	フィジカルアドレス
pState	:	ESM を格納するポインタ
Timeout	:	タイムアウト

戻り値

ACAT_ER_OK	:	正常
ACAT_ER_OK 以外	:	エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットの ESM を取得します。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Set_State 関数

機能 スレーブユニットの ESM を設定します

書式 int ACat_Slv_Set_State(int Address, int State, int Timeout)

引数

Address	: フィジカルアドレス
State	: ESM を格納するポインタ
Timeout	: タイムアウト

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットの ESM を設定します。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Get_Info 関数

機能 スレーブユニットのスレーブ情報を取得します

書式 int ACat_Slv_Get_Info(int Address, ACAT_SLAVE_INFO *pSlaveInfo)

引数
 Address : フィジカルアドレス
 pSlaveInfo : スレーブ情報を格納するポインタ

スレーブ情報

```
typedef struct {
    BYTE   DeviceName[DEVICE_NAMESIZE];
    DWORD  VendorID;
    DWORD  ProductCode;
    DWORD  RevisionNo;
    DWORD  SerialNo;
    WORD   PortState;
    BOOL   DcSupport;
    WORD   AliasAddress;
    WORD   PhysAddress;
    DWORD  PdInputOffset;
    DWORD  PdInputSize;
    DWORD  PdOutputOffset;
    DWORD  PdOutputSize;
} ACAT_SLAVE_INFO;
```

DeviceName : スレーブデバイス名称
 VendorID : ベンダーID
 ProductCode : プロダクトコード
 RevisionNo : レビジョン番号
 SerialNo : シリアル番号
 PortState : ステータス
 DcSupport : DC サポート [FALSE:無効, TRUE:有効]
 AliasAddress : スレーブアドレス
 PhysAddress : フィジカルアドレス
 PdInputOffset : PDO 入力オフセット
 PdInputSize : PDO 入力サイズ
 PdOutputOffset : PDO 出力オフセット
 PdOutputSize : PDO 出力サイズ

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットのスレーブ情報を取得します。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
スレーブ情報は ACat_Mst_Get_Slave_Info() で取得できるスレーブ情報と同じです。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Get_Address 関数

機能

フィジカルアドレスを取得します

書式

```
int ACat_Slv_Get_Address(DWORD SlaveID, int *pAddress)
```

引数

SlaveID : スレーブ ID
pAddress : フィジカルアドレスを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したスレーブユニットのフィジカルアドレスを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Get_SlaveID 関数

機能 スレーブ ID を取得します

書式 int ACat_Slv_Get_SlaveID(int Address, DWORD *pSlaveID)

引数 Address : フィジカルアドレス
 pSlaveID : スレーブ ID を格納するポインタ

戻り値 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 フィジカルアドレスで指定したスレーブユニットのスレーブ ID を取得します。
 この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Reset 関数

機能

スレーブユニットをソフトウェアリセットします

書式

```
int ACat_Slv_Reset(int Address, int Timeout)
```

引数

Address : フィジカルアドレス
Timeout : タイムアウト時間

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットをソフトウェアリセットします。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
この関数コールは EtherCAT 通信開始後しか使用できません。

ACat_Slv_Sdo_Download 関数

機能

スレーブユニットの SDO メッセージをダウンロードします

書式

```
int ACat_Slv_Sdo_Download(int Address, int Index, int SubIndex, char *pBuffer,  
                           int Length)
```

引数

Address : フィジカルアドレス
Index : インデックス
SubIndex : サブインデックス
pBuffer : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットからインデックス、サブインデックス(*1)で指定したオブジェクトディクショナリのパラメータをダウンロードします。フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。この関数コールは EtherCAT 通信開始後しか使用できません。

(*1) インデックス、サブインデックス、データサイズについては各スレーブユニットのマニュアルを参照ください。

ACat_Slv_Sdo_DownloadReq 関数

機能

スレーブユニットのSDOメッセージのダウンロードをリクエストします

書式

```
int ACat_Slv_Sdo_DownloadReq(int Address, int Index, int SubIndex, char *pBuffer,
                              int Length)
```

引数

Address : フィジカルアドレス
Index : インデックス
SubIndex : サブインデックス
pBuffer : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットにインデックス、サブインデックス(*1)で指定したオブジェクトディクショナリのパラメータのダウンロードをリクエストし、関数は直ぐに戻ります。

リクエスト結果は、通知機能を使用して受け取ることができます。

通知機能については、ACat_Mst_Set_Notification_Hook 関数を参照ください。

フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。この関数コールはEtherCAT通信開始後しか使用できません。

(*1)インデックス、サブインデックス、データサイズについては各スレーブユニットのマニュアルを参照ください。

ACat_Slv_Sdo_Upload 関数

機能

スレーブユニットの SDO メッセージをアップロードします

書式

```
int ACat_Slv_Sdo_Upload(int Address, int Index, int SubIndex, char *pBuffer,  
                        int Length)
```

引数

Address : フィジカルアドレス
Index : インデックス
SubIndex : サブインデックス
pBuffer : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットからインデックス、サブインデックス(*1)で指定したオブジェクトディクショナリのパラメータをアップロードします。フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。この関数コールは EtherCAT 通信開始後しか使用できません。

(*1) インデックス、サブインデックス、データサイズについては各スレーブユニットのマニュアルを参照ください。

ACat_Slv_Sdo_UploadReq 関数

機能

スレーブユニットのSDOメッセージのアップロードをリクエストします

書式

```
int ACat_Slv_Sdo_UploadReq(int Address, int Index, int SubIndex, char *pBuffer,  
int Length)
```

引数

Address : フィジカルアドレス
Index : インデックス
SubIndex : サブインデックス
pBuffer : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットにインデックス、サブインデックス(*1)で指定したオブジェクトディクショナリのパラメータのアップロードをリクエストし、関数は直ぐに戻ります。

リクエスト結果は、通知機能を使用して受け取ることができます。

通知機能については、ACat_Mst_Set_Notification_Hook 関数を参照ください。

フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。この関数コールはEtherCAT通信開始後しか使用できません。

(*1)インデックス、サブインデックス、データサイズについては各スレーブユニットのマニュアルを参照ください。

ACat_Slv_Foe_Download 関数

機能

スレーブユニットの FoE メッセージをダウンロードします

書式

```
int ACat_Slv_Foe_Download(int Address, char *pFilename, int DataLen, DWORD Password,
                          int Timeout)
```

引数

Address : フィジカルアドレス
pFilename : ダウンロードするファイル名を格納するポインタ
DataLen : ダウンロードするファイルのデータサイズ
Password : ダウンロードパスワード
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットへファイル名で指定したファイルをダウンロードします。

フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。この関数コールは EtherCAT 初期化の後、EtherCAT ステートマシーンを Bootstrap(スレーブのファームウェアを書き換えるための特殊な状態)に変更後に使用できます。

(※1) ファームウェアの書き換えはスレーブが対応している必要があり、更新するファームウェアファイルとパスワードが必要です。

更新するファームウェアファイルとパスワードはスレーブメーカーから入手してください。

(※2) もし、ファームウェアの更新に失敗した場合は、スレーブメーカーに復旧方法を確認ください。

ACat_Slv_Foe_DownloadReq 関数

機能

スレーブユニットのFoEメッセージのダウンロードをリクエストします

書式

```
int ACat_Slv_foe_DownloadReq(int Address, char *pFilename, int DataLen,  
                             DWORD Password, int Timeout)
```

引数

Address : フィジカルアドレス
pFilename : ダウンロードするファイル名を格納するポインタ
DataLen : ダウンロードするファイルのデータサイズ
Password : ダウンロードパスワード
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットへファイル名で指定したファイルのダウンロードをリクエストし、関数は直ぐに戻ります。

リクエスト結果は、通知機能を使用して受け取ることができます。

通知機能については、ACat_Mst_Set_Notification_Hook 関数を参照ください。

フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。この関数コールはEtherCAT初期化の後、EtherCATステートマシーンをBootstrap(スレーブのファームウェアを書き換えるための特殊な状態)に変更後に使用できます。

(※1) ファームウェアの書き換えはスレーブが対応している必要があり、更新するファームウェアファイルとパスワードが必要です。

更新するファームウェアファイルとパスワードはスレーブメーカーから入手してください。

(※2) もし、ファームウェアの更新に失敗した場合は、スレーブメーカーに復旧方法を確認ください。

ACat_Slv_Foe_Upload 関数

機能

スレーブユニットの FoE メッセージをアップロードします

書式

```
int ACat_Slv_Foe_Upload(int Address, char *pFilename, int DataLen, DWORD Password,
                        int Timeout)
```

引数

Address : フィジカルアドレス
pFilename : アップロードするファイル名を格納するポインタ
DataLen : アップロードするファイルのデータサイズ
Password : アップロードパスワード
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットからファームウェアをアップロードし、ファイル名で指定したファイルパスに保存します。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
この関数コールは EtherCAT 初期化の後、EtherCAT ステートマシーンを Bootstrap(スレーブのファームウェアを書き換えるための特殊な状態)に変更後に使用できます。

(※1) ファームウェアのアップロードはスレーブが対応している必要があり、アップロードにはパスワードが必要です。
パスワードはスレーブメーカーから入手してください。

ACat_Slv_Foe_UploadReq 関数

機能

スレーブユニットのFoEメッセージのアップロードをリクエストします

書式

```
int ACat_Slv_Foe_UploadReq(int Address, char *pFilename, int DataLen,  
                           DWORD Password, int Timeout)
```

引数

Address : フィジカルアドレス
pFilename : アップロードするファイル名を格納するポインタ
DataLen : アップロードするファイルのデータサイズ
Password : アップロードパスワード
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

フィジカルアドレスで指定したスレーブユニットからファームウェアのアップロードをリクエストし、関数は直ぐに戻ります。
リクエスト結果は、通知機能を使用して受け取ることができます。
通知機能については、ACat_Mst_Set_Notification_Hook 関数を参照ください。
リクエスト結果が成功の場合、ファイル名で指定したファイルパスに保存します。
フィジカルアドレスはスレーブ情報の読み込みで取得したフィジカルアドレスを使用します。
この関数コールは EtherCAT 初期化の後、EtherCAT ステートマシンの Bootstrap(スレーブのファームウェアを書き換えるための特殊な状態)に変更後に使用できます。

(※1) ファームウェアのアップロードはスレーブが対応している必要があり、アップロードにはパスワードが必要です。
パスワードはスレーブメーカーから入手してください。

ACat_Slv_Read_Register 関数

- 機能** スレーブユニットの ASIC のレジスタを読みみます
- 書式** `int ACat_Slv_Read_Register (int Address, int Offset, int Length, void *pBuffer, int Timeout)`
- 引数**
- | | |
|---------|-----------------------------------|
| Address | : フィジカルアドレス |
| Offset | : レジスタのオフセット値 |
| Length | : 読むレジスタ値のサイズ (1 バイト、2 バイト、4 バイト) |
| pBuffer | : 読んだレジスタ値を格納するバッファのポインタ |
| Timeout | : タイムアウトまでの時間 [ms] |
- 戻り値**
- | | |
|---------------|------------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード (エラーコード一覧を参照) |
- 説明** スレーブユニットの ASIC のレジスタを読みみます。
ASIC レジスタの詳細は、ASIC のマニュアルを参照してください。

ACat_Slv_Write_Register 関数

- 機能** スレーブユニットの ASIC のレジスタを書込みます
- 書式** `int ACat_Slv_Write_Register (int Address, int Offset, int Length, void *pBuffer, int Timeout)`
- 引数**
- | | |
|---------|------------------------------------|
| Address | : フィジカルアドレス |
| Offset | : レジスタのオフセット値 |
| Length | : 書込むレジスタ値のサイズ (1 バイト、2 バイト、4 バイト) |
| pBuffer | : 書込むレジスタ値を格納するバッファのポインタ |
| Timeout | : タイムアウトまでの時間 [ms] |
- 戻り値**
- | | |
|---------------|------------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード (エラーコード一覧を参照) |
- 説明** スレーブユニットの ASIC のレジスタを書込みます。
ASIC レジスタの詳細は、ASIC のマニュアルを参照してください。

ACat_Slv_Read_Eemrom 関数

機能 スレーブユニットの EEPROM を読みます

書式 `int ACat_Slv_Read_Eeprom (int Address, int Offset, WORD *pBuffer,
DWORD *pNumOutData, int Timeout)`

引数

Address	: フィジカルアドレス
Offset	: EEPROM オフセット値
pBuffer	: 読み込んだ値を格納するバッファのポインタ
pNumOutData	: 読み込まれたサイズを格納するバッファのポインタ
Timeout	: タイムアウトまでの時間 [ms]

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブユニットの EEPROM のレジスタを読みます。
EEPROM にアクセスするには、ACat_Slv_Assign_Eeprom 関数の SlavePDIAccessEnable を false に設定する必要があります。

ACat_Slv_Write_Eeprom 関数

機能

スレーブユニットの EEPROM を書込みます

書式

```
int ACat_Slv_Write_Eeprom (int Address, int Offset, WORD *pBuffer,  
                           int Timeout)
```

引数

Address : フィジカルアドレス
Offset : EEPROM オフセット値
pBuffer : 書込む値を格納するバッファのポインタ
Timeout : タイムアウトまでの時間 [ms]

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブユニットの EEPROM を書込みます。
EEPROM にアクセスするには、ACat_Slv_Assign_Eeprom 関数の SlavePDIAccessEnable を false に設定する必要があります。

ACat_Slv_Reload_Eeprom 関数

機能

スレーブユニットのEEPROMをリロードします

書式

```
int ACat_Slv_Reload_Eeprom (int Address, int Timeout)
```

引数

Address : フィジカルアドレス
Timeout : タイムアウトまでの時間 [ms]

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブユニットのEEPROMをリロードします。
EEPROMにアクセスするには、ACat_Slv_Assign_Eeprom関数のSlavePDIAccessEnableをfalseに設定する必要があります。

ACat_Slv_Assign_Eeprom 関数

機能

スレーブユニットのEEPROMアクセスを設定します

書式

```
int ACat_Slv_Assign_Eeprom (int Address, BOOL SlavePDIAccessEnable,  
                             BOOL ForceAssign, int Timeout)
```

引数

Address	: フィジカルアドレス
SlavePDIAccessEnable	: スレーブ PDI アクセス有効/無効
ForceAssign	: フォース割当て
Timeout	: タイムアウトまでの時間 [ms]

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明

スレーブユニットのEEPROMアクセスを設定します。
EEPROMにアクセスするには、SlavePDIAccessEnableをfalseに設定する必要があります。

ACat_Slv_Active_Eeprom 関数

- 機能** スレーブユニットの EEPROM アクセス設定を読み出します
- 書式** `int ACat_Slv_Active_Eeprom (int Address, BOOL *pSlavePDIAccessActive, int Timeout)`
- 引数**
- | | |
|-----------------------|-----------------------------|
| Address | : フィジカルアドレス |
| pSlavePDIAccessActive | : スレーブ PDI アクセス設定値を格納するポインタ |
| Timeout | : タイムアウトまでの時間 [ms] |
- 戻り値**
- | | |
|---------------|-----------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード(エラーコード一覧を参照) |
- 説明** スレーブユニットの EEPROM アクセス設定を読み出します。

2-4 マスタ通知

ACAT_NOTIFY_STATECHANGED 通知

機能 動作状態の変更を通知

コード ACAT_NOTIFY_STATECHANGED

パラメータ ACAT_STATECHANGE 動作状態 データ構造体

通知パラメータ

```
typedef struct {
    int oldState;
    int newState;
} ACAT_STATECHANGE;
```

oldState : 古い動作ステータス
newState : 新しい動作ステータス

説明 マスタの動作状態の変更が発生した時に通知します。

ACAT_NOTIFY_ETH_LINK_CONNECTED 通知

機能	Ethernet ケーブルが接続時に通知
コード	ACAT_NOTIFY_ETH_LINK_CONNECTED
パラメータ	なし
説明	Ethernet ケーブルの接続が発生した時に通知します。

ACAT_NOTIFY_DC_STATUS 通知

機能 Distributed clocks (DC) の初期化を通知

コード ACAT_NOTIFY_DC_STATUS

パラメータ DWORD State : エラーコード(エラーコード一覧を参照)

説明 DC ステータス通知。DC が初期化されるか、トポロジーの変更が行われるとすぐに、ACAT_NOTIFY_DC_SLV_SYNC の後に通知します。
トポロジーが変更された後、スレーブが同期しなかった場合、ACAT_NOTIFY_DC_SLV_SYNC なしで受信される可能性があります。

ACAT_NOTIFY_DC_SLV_SYNC 通知

- 機能** DCの同期通信が開始、または逸脱を通知
- コード** ACAT_NOTIFY_DC_SLV_SYNC
- パラメータ** ACAT_DC_SYNC_NTIFY : DC同期が発生時のデータ構造体

通知パラメータ

```
typedef struct {
    WORD StationAddress;
    WORD AutoIncAddr;
    char Name[80];
} SLAVE_PROPERTIES;
```

StationAddress : フィジカルアドレス
AutoIncAddr : インクリメントアドレス
Name : スレーブ名

```
typedef struct {
    DWORD IsInSync;
    DWORD IsNegative;
    DWORD Deviation;
    SLAVE_PROPERTIES SlaveProp;
} ACAT_DC_SYNC_NTIFY;
```

IsInSync : 同期が偏差値の範囲内にあるかどうか [TRUE, FALSE]
IsNegative : 偏差が正か負かどちらか [TRUE, FALSE]
Deviation : 同期偏差値 [nsec]
SlaveProp : スレーブ情報

- 説明** DCスレーブ同期通知。
スレーブが同期するか、同期が外れると通知します。

ACAT_NOTIFY_DCM_SYNC 通知

機能 DCM 同期が発生時に通知

コード ACAT_NOTIFY_DCM_SYNC

パラメータ ACAT_DCM_SYNC_NTIFY : DCM 同期が発生時のデータ構造体

通知パラメータ

```
typedef struct {  
    DWORD IsInSync;  
    int CtlErrorNsecCur;  
    int CtlErrorNsecAvg;  
    int CtlErrorNsecMax;  
} ACAT_DCM_SYNC_NTIFY;
```

IsInSync : マスタの時間と基準クロックが同期しているかどうか [TRUE, FALSE]
CtlErrorNsecCur : マスタの設定値と実際の値の現在の差 [nsec]
CtlErrorNsecAvg : マスタの設定値と実際の値の平均差 [nsec]
CtlErrorNsecMax : マスタの設定値と実際の値の最大差 [nsec]

説明 マスタの動作状態の変更が発生した時に通知します。

ACAT_NOTIFY_SLAVE_PRESENCE 通知

機能 非表示のスレーブが表示された時に通知

コード ACAT_NOTIFY_SLAVE_PRESENCE

パラメータ ACAT_SLAVE_PRESENCE_NTFY : スレーブ表示のデータ構造体

通知パラメータ

```
typedef struct {  
    WORD StationAddress;  
    BYTE Present;  
} ACAT_SLAVE_PRESENCE_NTFY;
```

StationAddress : フィジカルアドレス
Present : 0: absent 1: present

説明 この通知はスレーブがネットワークに現れたり消えたりした場合に通知します。

ACAT_NOTIFY_SB_STATUS 通知

- 機能** バススキャン終了時の通知
- コード** ACAT_NOTIFY_SB_STATUS
- パラメータ** ACAT_SB_STATUS_NOTIFY : バススキャン結果データ構造体

通知パラメータ

```
typedef struct {  
    DWORD ResultCode;  
    DWORD SlaveCount;  
} ACAT_SB_STATUS_NOTIFY;
```

ResultCode : エラーコード(エラーコード一覧を参照)
SlaveCount : バスに接続されたスレーブの数

- 説明** バススキャンが終了した時に通知されます。

ACAT_NOTIFY_SB_MISMATCH 通知

機能 バススキャンが不一致を通知

コード ACAT_NOTIFY_SB_MISMATCH

パラメータ ACAT_SB_MISMATCH : バススキャン不一致が発生時のデータ構造体

通知パラメータ

```
typedef struct {
    WORD PrevFixedAddress;
    WORD PrevPort;
    WORD PrevAIncAddress;
    WORD BusAIncAddress;
    DWORD BusVendorId;
    DWORD BusProdCode;
    DWORD BusRevisionNo;
    DWORD BusSerialNo;
    WORD BusFixedAddress;
    WORD IdentificationVal;
    WORD CfgFixedAddress;
    WORD CfgAIncAddress;
    DWORD CfgVendorId;
    DWORD CfgProdCode;
    DWORD CfgRevisionNo;
    DWORD CfgSerialNo;
    DWORD IdentValidationError;
    WORD IdentCmdHdr[5];
    DWORD CmdData;
    DWORD CmdVMask;
    DWORD CmdVData;
} ACAT_SB_MISMATCH;
```

PrevFixedAddress : 前スレーブのフィジカルアドレス
 PrevPort : 前スレーブのポート
 PrevAIncAddress : 前スレーブインクリメントアドレス
 BusAIncAddress : 予期しないスレーブ(バス)のインクリメントアドレス
 BusVendorId : 予期しないスレーブ(バス)のベンダーID
 BusProdCode : 予期しないスレーブ(バス)のプロダクトコード
 BusRevisionNo : 予期しないスレーブ(バス)のリビジョン番号
 BusSerialNo : 予期しないスレーブ(バス)のシリアル番号
 BusFixedAddress : 予期しないスレーブ(バス)のフィジカルアドレス
 IdentificationVal : 最後に使用時にスレーブから読み取られた最後の識別値

CfgFixedAddress	: 不足スレーブ(コンフィグ)のフィジカルアドレス
CfgAIncAddress	: 不足スレーブ(コンフィグ)のインクリメントアドレス
CfgVendorId	: 不足スレーブ(コンフィグ)のベンダーID
CfgProdCode	: 不足スレーブ(コンフィグ)のプロダクトコード
CfgRevisionNo	: 不足スレーブ(コンフィグ)のリビジョン番号
CfgSerialNo	: 不足スレーブ(コンフィグ)のシリアル番号
IdentValidationError	: ホットコネクト識別コマンドのスレーブへの送信結果
IdentCmdHdr	: 最後の HotConnect 識別コマンドヘッダー
CmdData	: 最後の識別コマンドのデータ部分の最初の DWORD
CmdVMask	: 最後の識別コマンドの検証マスクの最初の DWORD
CmdVData	: 最後の識別コマンドの検証データの最初の DWORD

説明

予期しないスレーブまたは必須のスレーブがないために、スキャンバスが接続されたスレーブと構成の不一致を検出した場合に通知されます。

ACAT_NOTIFY_SB_DUPLICATE_HC_NODE 通知

機能 全く同じユニット検出を通知

コード ACAT_NOTIFY_SB_DUPLICATE_HC_NODE

パラメータ ACAT_SB_MISMATCH : バススキャン不一致が発生時のデータ構造体
(※「ACAT_NOTIFY_SB_MISMATCH 通知」を参照)

説明 重複したスレーブが原因で、スキャン中にスキャンバスの不一致が検出されました。ネットワーク上に同じ製品コード、ベンダーID、および識別値（エイリアスアドレスまたはスイッチ ID）を持つ2つのスレーブがある場合、アプリケーションはこの通知を受け取りません。

ACAT_NOTIFY_MBOXRCV 通知

- 機能** メールボックス転送の完了通知
- コード** ACAT_NOTIFY_MBOXRCV
- パラメータ** ACAT_MBXTFER : メールボックス転送データ構造体

通知パラメータ

```
typedef struct {  
    DWORD          ClientId;  
    ACAT_MBXTFER_DESC MbxTferDesc;  
    int            MbxTferType;  
    DWORD          DataLen;  
    BYTE*          MbxTferData;  
    int            TferStatus;  
    DWORD          ErrorCode;  
    DWORD          TferId;  
    ACAT_MBX_DATA  MbxData;  
} ACAT_MBXTFER;
```

ClientId : クライアント ID
MbxTferDesc : メールボックス転送記述子
MbxTferType : メールボックス転送オブジェクトに書き込まれたタイプ情報
DataLen : メールボックス転送のデータバイト量
MbxTferData : メールボックス転送のデータを格納するポインタ
TferStatus : メールボックス転送状態
ErrorCode : エラーコード
TferId : 新しいメールボックスを転送する毎に割り当てる転送 ID
MbxData : メールボックスデータ (使用しません)

- 説明** メールボックス転送の完了した時に通知されます。

ACAT_NOTIFY_AUTO_RETURN 通知

機能	自動復帰の通知
コード	ACAT_NOTIFY_AUTO_RETURN
パラメータ	なし
説明	自動復帰が終了した時に通知されます。

ACAT_NOTIFY_ER_CYCCMD_WKC 通知

- 機能** サイクリック通信で WKC エラーが発生時に通知
- コード** ACAT_NOTIFY_ER_CYCCMD_WKC
- パラメータ** ACAT_WKCERROR : WKC エラー結果データ構造体

通知パラメータ

```
typedef struct {
    SLAVE_PROPERTIES SlaveProp;
    BYTE  Cmd;
    BYTE  Rsvd[3];
    DWORD Addr;
    WORD  WkcSet;
    WORD  WkcAct;
} ACAT_WKCERROR;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
Cmd : このエラーが発生した EtherCAT コマンド
Addr : 論理アドレスまたは物理アドレス (ADP / ADO)
WkcSet : ワーキングカウンターの設定値
WkcAct : ワーキングカウンターの実際の値

- 説明** プロセスデータを更新するために、いくつかの EtherCAT コマンドがマスタによって周期的に送信されます。
これらのコマンドは、1 つまたは複数のスレーブに対応します。
これらの EtherCAT コマンドには、アドレス指定されたスレーブごとにインクリメントする必要のあるワーキングカウンターが含まれています。
EtherCAT コマンドがマスタによって受信された後、ワーキングカウンターがチェックされません。
予想されるワーキングカウンターが受信したコマンドのワーキングカウンターと一致しない場合、エラー ACAT_NOTIFY_ER_CYCCMD_WKC が示されます。
詳細なエラー情報は、データ構造体 ACAT_WKCERROR に保存されます。

ACAT_NOTIFY_ER_MASTER_INITCMD_WKC 通知

機能 マスタ init コマンドで WKC エラーが発生時の通知

コード ACAT_NOTIFY_ER_MASTER_INITCMD_WKC

パラメータ ACAT_WKCERROR : WKC エラー結果データ構造体

説明 このエラーは、マスタ init コマンドを送信するときに、カウンターが機能しない場合に表示されます。
マスタが期待するワーキングカウンター値は、各マスタ init コマンドのコンフィグファイルによって決定されます。
コンフィグファイルに init コマンドの「Cnt」エントリがない場合、カウンターの検証は機能しません。
ワーキングカウンターは、この init コマンドを処理する必要があるすべてのスレーブによってインクリメントされる必要があります。
詳細なエラー情報は、データ構造体 ACAT_WKCERROR に保存されます。

ACAT_NOTIFY_ER_SLAVE_INITCMD_WKC 通知

機能 スレーブ init コマンドで WKC エラーが発生時の通知

コード ACAT_NOTIFY_ER_SLAVE_INITCMD_WKC

パラメータ ACAT_WKCERROR : WKC エラー結果データ構造体

説明 このエラーは、スレーブの初期化コマンドを送信するときに、カウンターの不一致が機能している場合に表示されます。
マスタが期待するワーキングカウンタ値は、各スレーブ init コマンドのコンフィグファイルによって決定されます。
コンフィグファイルに init コマンドの「Cnt」エントリがない場合、カウンターの検証は機能しません。
詳細なエラー情報は、データ構造体 ACAT_WKCERROR に保存されます。
構造体メンバーSlaveProp には、対応するスレーブデバイスに関する情報が含まれています。

ACAT_NOTIFY_ER_COE_MBXSNW_WKC 通知

機能 CoE の MailBox 送信で WKC エラーが発生時の通知

コード ACAT_NOTIFY_ER_COE_MBXSNW_WKC

パラメータ ACAT_WKCERROR : WKC エラー結果データ構造体

説明 このエラーは、CoE メールボックス書き込みコマンドのワーキングカウンターが期待値 1 に設定されていない場合に表示されます。
詳細なエラー情報は、データ構造体 ACAT_WKCERROR に保存されます。
構造体メンバー SlaveProp には、対応するスレーブデバイスに関する情報が含まれています。

ACAT_NOTIFY_ER_FOE_MBXSEND_WKC 通知

機能 FoE の MailBox 送信で WKC エラーが発生時の通知

コード ACAT_NOTIFY_ER_FOE_MBXSEND_WKC

パラメータ ACAT_WKCERROR : WKC エラー結果データ構造体

説明 このエラーは、FoE メールボックス書き込みコマンドのワーキングカウンターが期待値 1 に設定されていない場合に表示されます。
詳細なエラー情報は、データ構造体 ACAT_WKCERROR に保存されます。
構造体メンバー SlaveProp には、対応するスレーブデバイスに関する情報が含まれています。

ACAT_NOTIFY_ER_FRAME_RESPONSE 通知

- 機能** 受信されたフレームが異常時の通知
- コード** ACAT_NOTIFY_ER_FRAME_RESPONSE
- パラメータ** ACAT_FRAME_RSPERROR : フレーム受信結果データ構造体

通知パラメータ

```
typedef struct {  
    BOOL IsCyclicFrame;  
    int  ErrorType;  
    BYTE EcCmdHeaderIdxSet;  
    BYTE EcCmdHeaderIdxAct;  
} ACAT_FRAME_RSPERROR;
```

IsCyclicFrame : フレームに周期的なコマンドが含まれるかどうか [TRUE, FALSE]
ErrorType : エラータイプ [0~6]
0 : イーサネットフレームを未受信 (タイムアウト、フレーム損失)
1 : 非循環フレームでの誤った IDX 値
2 : 予期しないフレームが受信
3 : イーサネットフレームが再送信 (タイムアウト、フレーム損失)
4 : すべての再試行メカニズムが非循環フレームの再送信に失敗
5 : 他のマスタの MAC を持つフレームを受信
6 : 非 EtherCAT フレームを受信

EcCmdHeaderIdxSet : 期待される IDX 値
EcCmdHeaderIdxAct : 実際の IDX 値

- 説明** このエラーは、実際に受信したイーサネットフレームが予想されるフレームと一致しない場合、または予想されるフレームが受信されなかった場合に通知されます。

ACAT_NOTIFY_ER_SLAVE_INITCMD_RESPONSE 通知

- 機能** スレーブ初期化コマンドで無応答時の通知
- コード** ACAT_NOTIFY_ER_SLAVE_INITCMD_RESPONSE
- パラメータ** ACAT_INITCMD_ERROR : 初期化コマンド結果データ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
    char StateChangeName[20];  
    int EErrorType;  
} ACAT_INITCMD_ERROR;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)

StateChangeName : エラー発生時の状態変更の説明

EErrorType : エラータイプ

- 0 : イーサネットフレームを未受信 (タイムアウト)
- 1 : 検証エラー (無効なスレーブコマンド応答)
- 2 : 初期化コマンドが失敗
- 3 : バスに存在しないスレーブ
- 4 : エラー (AL ステータスレジスタ)
- 5 : メールボックスの初期化コマンドでのエラー
- 6 : PDI ウォッチドッグが検出

- 説明** このエラーコードは、スレーブがスレーブ初期化コマンドの送信中に適切に応答しない場合に
 表示されます。
 スレーブの初期化コマンドは、コンフィグファイルで定義されています。
 ここでタイムアウト値が定義されていない場合、フレーム応答は1つの単一サイクル内で
 予期されます。

ACAT_NOTIFY_ER_MASTER_INITCMD_RESPONSE 通知

機能 マスタ初期化コマンドで無応答時の通知

コード ACAT_NOTIFY_ER_MASTER_INITCMD_RESPONSE

パラメータ ACAT_INITCMD_ERROR : 初期化コマンド結果データ構造体

説明 マスタ初期化コマンドの送信中に欠落または誤ったコマンド応答が検出された場合、このエラーコードが表示されます。
マスタ init コマンドは、コンフィグファイルで定義されています。
タイムアウト値が定義されていない場合、フレーム応答は1つの単一サイクル内で予期されます。
詳細なエラー情報は、ACAT_INITCMD_ERROR に保存されます。

ACAT_NOTIFY_ER_MBSLAVE_INITCMD_TIMEOUT 通知

機能 メールボックス初期化コマンドでタイムアウト時の通知

コード ACAT_NOTIFY_ER_MBSLAVE_INITCMD_TIMEOUT

パラメータ ACAT_INITCMD_ERROR : 初期化コマンド結果データ構造体

説明 本通知は ACAT_NOTIFY_ER_SLAVE_INITCMD_RESPONSE と同じですが、メールボックスの初期化コマンドを処理するときにタイムアウトが発生した場合に通知されます。
CoE メールボックススレーブに使用されるタイムアウト値は、コンフィグファイルで定義されています。
この値が0に設定されている場合、EtherCAT マスタは 500 ミリ秒の固定タイムアウト値を使用します。

ACAT_NOTIFY_ER_NOT_ALL_DEVICES_OPERATIONAL 通知

機能 全デバイスがOPでなくなった時の通知

コード ACAT_NOTIFY_ER_NOT_ALL_DEVICES_OPERATIONAL

パラメータ なし

説明 周期的なフレームを処理するとき、EtherCAT マスタはすべてのスレーブがまだ動作状態にあるかどうかをチェックします。
少なくとも1つのスレーブデバイスが動作可能でない場合、このエラーが表示されます。

ACAT_NOTIFY_ER_ETH_LINK_NOT_CONNECTED 通知

機能 Ethernet ケーブルが切断された時の通知

コード ACAT_NOTIFY_ER_ETH_LINK_NOT_CONNECTED

パラメータ なし

説明 この通知はイーサネットリンクが切断された場合に表示されます。
永続的なフレーム損失の場合、スレーブは接続されていますが、スレーブは見つかりません。
これはリンク接続の検出に影響しないため、この通知は永続的なフレーム損失では示されません。

ACAT_NOTIFY_ER_STATUS_SLAVE 通知

機能 1つ以上のスレーブにエラーが発生時の通知

コード ACAT_NOTIFY_ER_STATUS_SLAVE

パラメータ なし

説明 周期的なフレームを処理するとき、EtherCAT マスタは、少なくとも1つのスレーブが AL-STATUS レジスタセットに ERROR ビットを持っているかどうかをチェックします。マスタは、エラーを示すスレーブの詳細なエラー情報を自動的に判断し、エラーステータスを確認し、エラーが発生している場合に通知します。

ACAT_NOTIFY_ER_SLAVE_STATUS_INFO 通知

機能 スレーブステータスエラー時の通知

コード ACAT_NOTIFY_ER_SLAVE_STATUS_INFO

パラメータ ACAT_SLAVE_ERROR_INFO : スレーブステータスエラー結果データ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
    WORD Status;  
    WORD StatusCode;  
} ACAT_SLAVE_ERROR_INFO;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
Status : スレーブステータス (AL STATUS)
StatusCode : スレーブステータスコード (AL STATUS CODE)

説明 マスタがスレーブエラーを検出するたびに、特定のスレーブのエラービットがクリアされ、このエラーコードが通知されます。
詳細なエラー情報は、ACAT_SLAVE_ERROR_INFO に保存されます。

ACAT_NOTIFY_ER_SLAVE_NOT_ADDRESSABLE 通知

機能

スレーブアドレスエラー時の通知

コード

ACAT_NOTIFY_ER_SLAVE_NOT_ADDRESSABLE

パラメータ

ACAT_WKCERROR : WKC エラー結果データ構造体

説明

マスタがそのステーションアドレスを使用してスレーブデバイスの状態を取得できない(ワーキングカウンタがインクリメントされない)場合、このエラーが通知されます。
詳細なエラー情報は、ACAT_WKCERROR に保存されます。
構造体メンバーSlaveProp には、対応するスレーブデバイスに関する情報が含まれています。

ACAT_NOTIFY_ER_MBSLAVE_COE_SDO_ABORT 通知

- 機能** メールボックスの SDO アボートが発生時の通知
- コード** ACAT_NOTIFY_ER_MBSLAVE_COE_SDO_ABORT
- パラメータ** ACAT_MAILBOX_SDO_ABORT : SDO アボートデータ構造体

通知パラメータ

```
typedef struct {
    SLAVE_PROPERTIES SlaveProp;
    DWORD ErrorCode;
    WORD ObjIndex;
    BYTE SubIndex;
} ACAT_MAILBOX_SDO_ABORT;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
ErrorCode : エラーコード (エラーコード一覧を参照)
ObjIndex : SDO オブジェクトインデックス
SubIndex : SDO オブジェクトサブインデックス

- 説明** スレーブは CoE アップロードおよびダウンロードを中止する場合があります。
init コマンドの送信中に SDO 転送が中止されると、本通知が発生します。
詳細なエラー情報は、ACAT_MAILBOX_SDO_ABORT に保存されます。

ACAT_NOTIFY_ER_CLIENTREGISTRATION_DROPPED 通知

機能 クライアント登録が削除された時の通知

コード ACAT_NOTIFY_ER_CLIENTREGISTRATION_DROPPED

パラメータ なし

説明 この通知は別のスレッドによって通信開始が呼び出されたためにクライアント登録が削除された場合に示されます。

ACAT_NOTIFY_ER_FOE_MBSLAVE 通知

- 機能** FoE メールボックスのアボートが発生時の通知
- コード** ACAT_NOTIFY_ER_FOE_MBSLAVE
- パラメータ** ACAT_MAILBOX_FOE_ABORT : FoE エラーデータ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
    DWORD dwErrorCode;  
    char achErrorString[80];  
} ACAT_MAILBOX_FOE_ABORT;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
dwErrorCode : エラーコード (エラーコード一覧を参照)
achErrorString : エラーメッセージ

- 説明** このエラーは、FoE メールボックスでアボートが発生するたびに通知されます。
詳細なエラー情報は、ACAT_MAILBOX_FOE_ABORT に保存されます。

ACAT_NOTIFY_ER_PDIWATCHDOG 通知

機能 PDI Watchdog がスレーブで発生時の通知

コード ACAT_NOTIFY_ER_PDIWATCHDOG

パラメータ ACAT_PDI_WATCHDOG : PDI ウォッチドッグエラーデータ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
} ACAT_PDI_WATCHDOG;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)

説明 このエラーは、PDI ウォッチドッグエラーが検出されるたびに通知されます。
詳細なエラー情報は、ACAT_PDI_WATCHDOG に保存されます。

ACAT_NOTIFY_ER_SLAVE_NOTSUPPORTED 通知

機能 スレーブ未サポート時の通知

コード ACAT_NOTIFY_ER_SLAVE_NOTSUPPORTED

パラメータ ACAT_SLAVE_NOT_SUPPORTED : スレーブ未サポートデータ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
} ACAT_SLAVE_NOT_SUPPORTED;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)

説明 通信開始時に間違ったカテゴリタイプが EEPROM で検出された場合、バススキャン中に通知されます。

ACAT_NOTIFY_ER_SLAVE_UNEXPECTED_STATE 通知

- 機能** スレーブが予期しない状態 (ESM) 時の通知
- コード** ACAT_NOTIFY_ER_SLAVE_UNEXPECTED_STATE
- パラメータ** ACAT_SLAVE_UNEXPECTED_STATE : 不測状態データ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
    int curState;  
    int expState;  
} ACAT_SLAVE_UNEXPECTED_STATE;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
curState : 現在の状態
expState : 予定の状態

- 説明** このエラーは、スレーブが予期しない状態に変わるたびに通知されます。
詳細なエラー情報は、ACAT_SLAVE_UNEXPECTED_STATE に保存されます。

ACAT_NOTIFY_ER_ALL_DEVICES_OPERATIONAL 通知

機能 全デバイスが OPERATIONAL 状態に戻った時の通知

コード ACAT_NOTIFY_ER_ALL_DEVICES_OPERATIONAL

パラメータ なし

説明 周期的なフレームを処理するとき、マスタはすべてのスレーブがまだ OPERATIONAL 状態にあるかどうかをチェックします。
これは ACAT_NOTIFY_ER_NOT_ALL_DEVICES_OPERATIONAL の通知発生後、すべてのスレーブが OPERATIONAL 状態に戻った時に通知されます。

ACAT_NOTIFY_ER_EEPROM_CHECKSUM 通知

- 機能** EEPROM のチェックサム異常時の通知
- コード** ACAT_NOTIFY_ER_EEPROM_CHECKSUM
- パラメータ** ACAT_SB_STATUS_NTIFY : バススキャン結果データ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
} ACAT_EEPROM_CHECKSUM_ERROR;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)

- 説明** このエラーは、EEPROM チェックサムエラーが検出されるたびに通知されます。詳細なエラー情報は、ACAT_EEPROM_CHECKSUM_ERROR 構造体に保存されます。

ACAT_NOTIFY_ER_LINE_CROSSED 通知

- 機能** ライン交差を検出時の通知
- コード** ACAT_NOTIFY_ER_LINE_CROSSED
- パラメータ** ACAT_SB_STATUS_NTIFY : ライン交差データ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
    WORD              InputPort;  
} ACAT_LINE_CROSSED;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
InputPort : フレームが受信されたポート

- 説明** ケーブル交換が検出されました。
すべてのスレーブのポート0はマスタに接続する必要があります。

ACAT_NOTIFY_ER_FRAMELOSS_AFTER_SLAVE 通知

機能 スレーブ後がフレームロス時の通知

コード ACAT_NOTIFY_ER_FRAMELOSS_AFTER_SLAVE

パラメータ ACAT_FRAMELOSS_AFTER_SLAVE_NTFY : スレーブ後のフレームロスデータ構造体

通知パラメータ

```
typedef struct {  
    SLAVE_PROPERTIES SlaveProp;  
    WORD Port;  
} ACAT_FRAMELOSS_AFTER_SLAVE_NTFY;
```

SlaveProp : スレーブ情報 (ACAT_NOTIFY_DC_SLV_SYNC 通知参照)
Port : ポート

説明 ポートを開くと通信が失われる場合、自動的に処理されます。
この通知は、ログメッセージのみが出力されるか、無視されます。

2-5 汎用 PDO/SDO アクセス関数

ACat_Gene_Create 関数

機能	汎用 PDO/SDO アクセス関数を初期化します
書式	<code>int ACat_Gene_Create(void)</code>
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある汎用 PDO/SDO アクセス関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 本関数コール後、スレーブユニットの PDO 及び SDO にアクセス可能となります。 汎用 PDO/SDO アクセス関数を使用する際には、必ずコールする必要があります。

ACat_Gene_Destroy 関数

機能 汎用 PDO/SDO アクセス関数を終了します

書式 int ACat_Gene_Destroy(void)

引数 なし

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 ライブラリ内にある汎用 PDO/SDO アクセス関数の終了処理を行います。
この関数コール後は、汎用 PDO/SDO アクセス関数からスレーブユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Gene_Read_Input 関数

機能

スレーブユニットから入力 PDO データを取得します

書式

```
int ACat_Gene_Read_Input(DWORD SlaveID, int Offset, int Length, char *pBuffer)
```

引数

SlaveID : スレーブ ID
Offset : PDO オフセット
Length : PDO データサイズ
pBuffer : 読込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したスレーブユニットの入力 PDO から PDO オフセット、PDO データサイズで指定したデータを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
読込データサイズはバイト単位で指定します。

ACat_Gene_Read_Output 関数

機能

スレーブユニットから出力 PDO データを取得します

書式

```
int ACat_Gene_Read_Output(DWORD SlaveID, int Offset, int Length, char *pBuffer)
```

引数

SlaveID : スレーブ ID
Offset : PDO オフセット
Length : PDO データサイズ
pBuffer : 読込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したスレーブユニットの出力 PDO から PDO オフセット、PDO データサイズで指定したデータを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
読込データサイズはバイト単位で指定します。

ACat_Gene_Write_Output 関数

- 機能** スレーブユニットの出力 PDO にデータを出力します
- 書式** `int ACat_Gene_Write_Output(DWORD SlaveID, int Offset, int Length, char *pBuffer)`
- 引数**
- | | |
|---------|--------------|
| SlaveID | : スレーブ ID |
| Offset | : PDO オフセット |
| Length | : PDO データサイズ |
| pBuffer | : 書込データ |
- 戻り値**
- | | |
|---------------|-----------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード(エラーコード一覧を参照) |
- 説明**
- スレーブ ID で指定したスレーブユニットの出力 PDO の PDO オフセット、PDO データサイズで指定したアドレスへデータを出力します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
書込データサイズはバイト単位で指定します。

ACat_Gene_Sdo_Download 関数

機能

スレーブユニットの SDO メッセージをダウンロードします

書式

```
int ACat_Gene_Sdo_Download(DWORD SlaveID, int Index, int SubIndex, char *pBuffer,
                           int Length)
```

引数

SlaveID : スレーブ ID
Index : インデックス
SubIndex : サブインデックス
pBuffer : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したスレーブユニットからインデックス、サブインデックス(*1)で指定したオブジェクトディクショナリのパラメータをダウンロードします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
データサイズはバイト単位で指定します。

(*1) インデックス、サブインデックス、データサイズについては各スレーブユニットのマニュアルを参照ください。

ACat_Gene_Sdo_Upload 関数

機能

スレーブユニットの SDO メッセージをアップロードします

書式

```
int ACat_Gene_Sdo_Upload(DWORD SlaveID, int Index, int SubIndex, char *pBuffer,
                          int Length)
```

引数

SlaveID : スレーブ ID
Index : インデックス
SubIndex : サブインデックス
pBuffer : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したスレーブユニットからインデックス、サブインデックス(*1)で指定したオブジェクトディクショナリのパラメータをアップロードします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
データサイズはバイト単位で指定します。

(*1) インデックス、サブインデックス、データサイズについては各スレーブユニットのマニュアルを参照ください。

2-6 CiA402 ドライブプロファイル制御関数

ACat_Motion_Create 関数

機能 CiA402 ドライブプロファイル制御関数を初期化します

書式 int ACat_Motion_Create(void)

引数 なし

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 ライブラリ内にある CiA402 ドライブプロファイル制御関数の初期化を行います。
この関数コールは EtherCAT 通信開始後しか使用できません。
本関数コール後、その他の CiA402 ドライブプロファイル制御関数にアクセス可能となります。
CiA402 ドライブプロファイル制御関数を使用する際には、必ずコールする必要があります。

ACat_Motion_Destroy 関数

機能	CiA402 ドライブプロファイル制御関数を終了します
書式	int ACat_Motion_Destroy(void)
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある CiA402 ドライブプロファイル制御関数の終了処理を行います。 この関数コール後は、CiA402 ドライブプロファイル制御関数にアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_Motion_ServoOn 関数

機能 CiA402 準拠スレーブをサーボ ON します

書式 int ACat_Motion_ServoOn(DWORD SlaveID, int Ch, int Timeout)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
Timeout	: タイムアウト

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットをサーボ ON します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後、モータ動作が可能になります。
接続されるスレーブユニットによっては、正常にサーボ ON しない場合があります。その場合は、各スレーブユニットのマニュアルに従ってサーボ ON してください。

ACat_Motion_ServoOff 関数

機能 CiA402 準拠スレーブをサーボ OFF します

書式 int ACat_Motion_ServoOff (DWORD SlaveID, int Ch, int Timeout)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
Timeout	: タイムアウト

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットをサーボ OFF します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常にサーボ OFF しない場合があります。その場合は、各スレーブユニットのマニュアルに従ってサーボ OFF してください。

ACat_Motion_MoveAbsolute 関数

機能

CiA402 準拠スレーブを絶対位置制御します

書式

```
int ACat_Motion_MoveAbsolute(DWORD SlaveID, int Ch, int Position, int Immediately,
                              int Timeout)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
Position : 目標位置
Immediately : 即時実行フラグ
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットにプロファイル位置制御モード (pp mode) で絶対位置制御を実行します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
目標位置 [607Ah] 以外のパラメータは、本関数を実行する前に、事前に設定しておく必要があります。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_MoveRelative 関数

機能

CiA402 準拠スレーブを相対位置制御します

書式

```
int ACat_Motion_MoveRelative(DWORD SlaveID, int Ch, int Position, int Immediately,
                             int Timeout)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
Position : 目標位置
Immediately : 即時実行フラグ
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットにプロファイル位置制御モード (pp mode) で相対位置制御を実行します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
目標位置 [607Ah] 以外のパラメータは、本関数を実行する前に、事前に設定しておく必要があります。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_MoveVelocity 関数

機能 CiA402 準拠スレーブを速度制御します

書式 int ACat_Motion_MoveVelocity (DWORD SlaveID, int Ch, int Velocity, int Timeout)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
Velocity	: 目標速度
Timeout	: タイムアウト

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットにプロファイル速度制御モード (pv mode) で速度制御を実行します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。目標速度 [60FFh] 以外のパラメータは、本関数を実行する前に、事前に設定しておく必要があります。

接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_MoveTorque 関数

機能 CiA402 準拠スレーブをトルク制御します

書式 int ACat_Motion_MoveTorque(DWORD SlaveID, int Ch, int Torque, int Timeout)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
Torque	: 指令トルク
Timeout	: タイムアウト

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットにプロファイルトルク制御モード (tq mode) で速度制御を実行します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。指令トルク [6071h] 以外のパラメータは、本関数を実行する前に、事前に設定しておく必要があります。

接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_MoveParameter 関数

機能

CiA402 準拠スレーブの動作パラメータを設定します

書式

int ACat_Motion_MoveParameter (DWORD SlaveID, int Ch, int Velocity, int Acceleration, int Deceleration, int Timeout)

引数

SlaveID : スレーブ ID
Ch : チャンネル
Velocity : 指令速度
Acceleration : 指令加速度
Deceleration : 指令減速度
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットに指令速度 [6081h]、指令加速度 [6083h] 指令減速度 [6084h] の設定を行います。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_Homing 関数

機能 CiA402 準拠スレーブを原点復帰動作します

書式 int ACat_Motion_Homing(DWORD SlaveID, int Ch, int HomeMethod, int Timeout)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
HomeMethod	: 原点復帰モード
Timeout	: タイムアウト

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットにホーミングモード (hm mode) で原点復帰を実行します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。原点復帰方式 [6098h] 以外のパラメータは、本関数を実行する前に、事前に設定しておく必要があります。

接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_HomingParameter 関数

機能

CiA402 準拠スレーブの原点復帰パラメータを設定します

書式

int ACat_Motion_HomingParameter (DWORD SlaveID, int Ch, int SwitchSearchSpeed,
int ZeroSearchSpeed, int Acceleration, int Timeout)

引数

SlaveID : スレーブ ID
Ch : チャンネル
SwitchSearchSpeed : スイッチサーチ速度
ZeroSearchSpeed : ゼロサーチ速度
Acceleration : 指令加速度
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットにスイッチサーチ速度 [6099h]、ゼロサーチ速度 [6099h]、指令加速度 [609Ah] の設定を行います。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_Stop 関数

機能	CiA402 準拠スレーブを停止します	
書式	int ACat_Motion_Stop(DWORD SlaveID, int Ch, int Timeout)	
引数	SlaveID	: スレーブ ID
	Ch	: チャンネル
	Timeout	: タイムアウト
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
	上記以外	: エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CiA402 準拠スレーブユニットの動作の停止を行います。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、 各スレーブユニットのマニュアルに従って設定してください。	

ACat_Motion_Reset 関数

機能

CiA402 準拠スレーブをエラーリセットします

書式

int ACat_Motion_Reset (DWORD SlaveID, int Ch, int Timeout)

引数

SlaveID : スレーブ ID
Ch : チャンネル
Timeout : タイムアウト

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのリセットを行います。
リセット後は、サーボ ON 状態 (Operation Enabled) に遷移します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、
各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadTargetReached 関数

機能

CiA402 準拠スレーブの目標到達を取得します

書式

```
int ACat_Motion_ReadTargetReached(DWORD SlaveID, int Ch, int *pTargetReached)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
TargetReached : 目標到達を格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットの目標到達 [6041h(Statusword) の bit10] を取得します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadHomingEnd 関数

機能 CiA402 準拠スレーブの原点復帰完了を取得します

書式 int ACat_Motion_ReadHomingEnd(DWORD SlaveID, int Ch, int *pHomingEnd)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
HomingEnd	: 原点復帰完了を格納するポインタ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットの原点復帰完了を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadAxisError 関数

機能

CiA402 準拠スレーブのエラーコードを取得します

書式

```
int ACat_Motion_ReadAxisError (DWORD SlaveID, int Ch, WORD *ErrorCode)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
ErrorCode : エラーコードを格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのエラーコード[603Fh]を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadActualPosition 関数

機能

CiA402 準拠スレーブの現在位置を取得します

書式

```
int ACat_Motion_ReadActualPosition(DWORD SlaveID, int Ch, int *pPosition)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
pPosition : 現在位置を格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットの現在位置 [6064h] を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadActualVelocity 関数

機能

CiA402 準拠スレーブの現在速度を取得します

書式

```
int ACat_Motion_ReadActualVelocity(DWORD SlaveID, int Ch, int *pVelocity)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
pVelocity : 現在速度を格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットの現在速度 [606Ch] を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadActualTorque 関数

機能	CiA402 準拠スレーブの現在トルクを取得します	
書式	int ACat_Motion_ReadActualTorque(DWORD SlaveID, int Ch, int *pTorque)	
引数	SlaveID	: スレーブ ID
	Ch	: チャンネル
	pTorque	: 現在トルクを格納するためのポインタ
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
	上記以外	: エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CiA402 準拠スレーブユニットの現在トルク [6077h] を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。	

ACat_Motion_ReadTargetPosition 関数

機能

CiA402 準拠スレーブの目標位置を取得します

書式

```
int ACat_Motion_ReadTargetPosition(DWORD SlaveID, int Ch, int *pPosition)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
pPosition : 目標位置を格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットの目標位置 [607Ah] を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadTargetVelocity 関数

機能

CiA402 準拠スレーブの目標速度を取得します

書式

```
int ACat_Motion_ReadTargetVelocity(DWORD SlaveID, int Ch, int *pVelocity)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
pVelocity : 目標速度を格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットの目標速度 [60FFh] を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadTargetTorque 関数

機能

CiA402 準拠スレーブの目標トルクを取得します

書式

```
int ACat_Motion_ReadTargetTorque(DWORD SlaveID, int Ch, int *pTorque)
```

引数

SlaveID : スレーブ ID

Ch : チャンネル

pTorque : 目標トルクを格納するためのポインタ

戻り値

ACAT_ER_OK : 正常

ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません

上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットの目標トルク [6071h] を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadInputData 関数

機能

CiA402 準拠スレーブのデジタル入力を取得します

書式

```
int ACat_Motion_ReadInputData(DWORD SlaveID, int Ch, DWORD *pIndata)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
pIndata : デジタル入力を格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのデジタル入力[60FDh]を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteOutputData 関数

機能

CiA402 準拠スレーブのデジタル出力を書込みます

書式

```
int ACat_Motion_WriteOutputData(DWORD SlaveID, int Ch, DWORD Outdata)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
Outdata : デジタル出力データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのデジタル出力[60FEh]を設定します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_SetParameter 関数

機能

CiA402 準拠スレーブのパラメータを書込みます

書式

```
int ACat_Motion_SetParameter (DWORD SlaveID, int Ch, WORD Index, BYTE SubIndex,  
                               BYTE *pData, WORD Length)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
Index : インデックス
SubIndex : サブインデックス
pData : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのインデックス、サブインデックスで指定したパラメータを設定します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_GetParameter 関数

機能

CiA402 準拠スレーブのパラメータを取得します

書式

```
int ACat_Motion_GetParameter (DWORD SlaveID, int Ch, WORD Index, BYTE SubIndex,  
                               BYTE *pData, WORD Length)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
Index : インデックス
SubIndex : サブインデックス
pData : データを格納するポインタ
Length : データサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのインデックス、サブインデックスで指定したパラメータを取得します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteControlword 関数

機能 CiA402 準拠スレーブのコントロールワードを書込みます

書式 int ACat_Motion_WriteControlword(DWORD SlaveID, int Ch, WORD Controlword)

引数

SlaveID	: スレーブ ID
Ch	: チャンネル
Controlword	: コントロールワード

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_NOTCIA402	: CiA402 仕様のデバイスではありません
上記以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード [6040h] を設定します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadStatusword 関数

機能

CiA402 準拠スレーブのステータスワードを取得します

書式

```
int ACat_Motion_ReadStatusword(DWORD SlaveID, int Ch, WORD *pStatusword)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル
pStatusword : ステータスワードを格納するためのポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのステータスワード [6041h] を取得します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteShutdown 関数

機能 CiA402 準拠スレーブに「Shutdown」を書込みます

書式 int ACat_Motion_WriteShutdown(DWORD SlaveID, int Ch)

引数 SlaveID : スレーブ ID
Ch : チャンネル

戻り値 ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Shutdown」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteSwitchOn 関数

機能 CiA402 準拠スレーブに「SwitchOn」を書込みます

書式 int ACat_Motion_WriteSwitchOn(DWORD SlaveID, int Ch)

引数 SlaveID : スレーブ ID
Ch : チャンネル

戻り値 ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Switch On」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteEnableOperation 関数

機能

CiA402 準拠スレーブに「Enable Operation」を書込みます

書式

```
int ACat_Motion_WriteEnableOperation(DWORD SlaveID, int Ch)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Enable Operation」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteDisableOperation 関数

機能

CiA402 準拠スレーブに「Disable Operation」を書込みます

書式

int ACat_Motion_WriteDisableOperation(DWORD SlaveID, int Ch)

引数

SlaveID : スレーブ ID
Ch : チャンネル

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Disable Operation」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteQuickStop 関数

機能 CiA402 準拠スレーブに「QuickStop」を書込みます

書式 int ACat_Motion_WriteQuickStop(DWORD SlaveID, int Ch)

引数 SlaveID : スレーブ ID
Ch : チャンネル

戻り値 ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Quick Stop」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteDisableVoltage 関数

機能 CiA402 準拠スレーブに「DisableVoltage」を書込みます

書式 int ACat_Motion_WriteDisableVoltage(DWORD SlaveID, int Ch)

引数 SlaveID : スレーブ ID
Ch : チャンネル

戻り値 ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Disable Voltage」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_WriteFaultReset 関数

機能

CiA402 準拠スレーブに「FaultReset」を書込みます

書式

```
int ACat_Motion_WriteFaultReset (DWORD SlaveID, int Ch)
```

引数

SlaveID : スレーブ ID
Ch : チャンネル

戻り値

ACAT_ER_OK : 正常
ACAT_ER_NOTCIA402 : CiA402 仕様のデバイスではありません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CiA402 準拠スレーブユニットのコントロールワード(6040h)に「Fault Reset」コマンドを書き込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

ACat_Motion_ReadFSASState 関数

- 機能** CiA402 準拠スレーブの FSA のステータスを取得します
- 書式** `int ACat_Motion_ReadFSASState(DWORD SlaveID, int Ch, WORD *pFSASState)`
- 引数**
- | | |
|------------|----------------------|
| SlaveID | : スレーブ ID |
| Ch | : チャンネル |
| pFSASState | : FSA ステータスを格納するポインタ |
- 戻り値**
- | | |
|-------------------|-------------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_NOTCIA402 | : CiA402 仕様のデバイスではありません |
| 上記以外 | : エラーコード(エラーコード一覧を参照) |
- 説明** スレーブ ID で指定した CiA402 準拠スレーブユニットの FSA のステータスを取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。接続されるスレーブユニットによっては、正常に動作しない場合があります。その場合は、各スレーブユニットのマニュアルに従って設定してください。

2-7 デジタル入出力ユニット関数

ACat_Dio_Create 関数

機能	デジタル入出力ユニット関数群を初期化します
書式	<code>int ACat_Dio_Create(void)</code>
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にあるデジタル入出力ユニット関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 本関数コール後、デジタル入出力ユニットにアクセス可能となります。 デジタル入出力ユニットを使用する際には、必ずコールする必要があります。

ACat_Dio_Destroy 関数

機能	デジタル入出力ユニット関数群を終了します
書式	int ACat_Dio_Destroy(void)
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にあるデジタル入出力ユニット関数の終了処理を行います。 この関数コール後は、デジタル入出力ユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_Dio_Open 関数

機能 デジタル入出力ユニットをオープンします

書式 int ACat_Dio_Open(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したデジタル入出力ユニットをオープンします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後、指定したデジタル入出力ユニットにアクセス可能となります。
デジタル入出力ユニットを使用する際には、必ずコールする必要があります。

ACat_Dio_Close 関数

機能 デジタル入出力ユニットをクローズします

書式 int ACat_Dio_Close(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したデジタル入出力ユニットをクローズします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後は、指定したデジタル入出力ユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Dio_Read 関数

機能

デジタル入出力ユニットから入力データを取得します

書式

```
int ACat_Dio_Read(DWORD SlaveID, DWORD len, char* pRead)
```

引数

SlaveID : スレーブ ID
len : 読込データサイズ
pRead : 読込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したデジタル入出力ユニットから入力データを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
読込データサイズはバイト単位で指定します。
例) DI32 ユニットの場、len=4 と指定します。

ACat_Dio_Write 関数

機能

デジタル入出力ユニットの出力にデータを出力します

書式

```
int ACat_Dio_Write(DWORD SlaveID, DWORD len, char* pWrite)
```

引数

SlaveID : スレーブ ID
len : 書込データサイズ
pWrite : 書込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したデジタル入出力ユニットへデータを出力します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
書込データサイズはバイト単位で指定します。
例) D032 ユニットの場、len=4 と指定します。

ACat_Dio_GetParam 関数

機能 デジタル入出力ユニットの設定パラメータを取得します

書式 int ACat_Dio_GetParam(DWORD SlaveID, WORD* pInFilter, WORD* pErrorOutput)

引数
 SlaveID : スレーブ ID
 pInFilter : 入力フィルタ
 pErrorOutput : 異常時出力

入力フィルタ	内容
DIO_DI_FILTER_TYPE_0MS	0: フィルタなし
DIO_DI_FILTER_TYPE_05MS	1: 0.5ms
DIO_DI_FILTER_TYPE_1MS	2: 1ms
DIO_DI_FILTER_TYPE_2MS	3: 2ms
DIO_DI_FILTER_TYPE_4MS	4: 4ms
DIO_DI_FILTER_TYPE_8MS	5: 8ms
DIO_DI_FILTER_TYPE_16MS	6: 16ms
DIO_DI_FILTER_TYPE_32MS	7: 32ms

異常時出力	内容
DIO_DO_ERROUTPUT_HOLD	0: 出力保持
DIO_DO_ERROUTPUT_CLEAR	1: 出力クリア

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明
 スレーブ ID で指定したデジタル入出力ユニットの設定パラメータを取得します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
 入力フィルタは DI, DIO ユニット、異常時出力は DO, DIO ユニットの時のみ、それぞれ有効になります。

ACat_Dio_SetParam 関数

機能 デジタル入出力ユニットの設定パラメータを設定します

書式 int ACat_Dio_SetParam(DWORD SlaveID, WORD InFilter, WORD ErrorOutput)

引数
 SlaveID : スレーブ ID
 InFilter : 入力フィルタ
 ErrorOutput : 異常時出力

入力フィルタ	内容
DIO_DI_FILTER_TYPE_0MS	0: フィルタなし
DIO_DI_FILTER_TYPE_05MS	1: 0.5ms
DIO_DI_FILTER_TYPE_1MS	2: 1ms
DIO_DI_FILTER_TYPE_2MS	3: 2ms
DIO_DI_FILTER_TYPE_4MS	4: 4ms
DIO_DI_FILTER_TYPE_8MS	5: 8ms
DIO_DI_FILTER_TYPE_16MS	6: 16ms
DIO_DI_FILTER_TYPE_32MS	7: 32ms

異常時出力	内容
DIO_DO_ERROUTPUT_HOLD	0: 出力保持
DIO_DO_ERROUTPUT_CLEAR	1: 出力クリア

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明
 スレーブ ID で指定したデジタル入出力ユニットの設定パラメータを設定します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
 入力フィルタは DI, DIO ユニット、異常時出力は DO, DIO ユニットの時のみ、それぞれ有効になります。

ACat_Dio_SaveParam 関数

- 機能** デジタル入出力ユニットの設定パラメータを保存します
- 書式** int ACat_Dio_SaveParam(DWORD SlaveID)
- 引数** SlaveID : スレーブ ID
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** デジタル入出力ユニットの設定パラメータを保存します。
ACat_Dio_SetParam 関数でパラメータを設定した後に実行してください。

ACat_Dio_LoadParam 関数

機能	デジタル入出力ユニットの設定パラメータを初期化します
書式	int ACat_Dio_LoadParam(DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	デジタル入出力ユニットの設定パラメータを初期化します。

ACat_Dio_SoftReset 関数

機能	デジタル入出力ユニットのソフトウェアリセットを行います
書式	int ACat_Dio_SoftReset (DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定したデジタル入出力ユニットのソフトウェアリセットを行います。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

2-8 アナログ入出力ユニット関数

ACat_Aio_Create 関数

機能	アナログ入出力ユニット関数群を初期化します
書式	<code>int ACat_Aio_Create(void)</code>
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にあるアナログ入出力ユニット関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 本関数コール後、アナログ入出力ユニットにアクセス可能となります。 アナログ入出力ユニットを使用する際には、必ずコールする必要があります。

ACat_Aio_Destroy 関数

- 機能** アナログ入出力ユニット関数群を終了します
- 書式** int ACat_Aio_Destroy(void)
- 引数** なし
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** ライブラリ内にあるアナログ入出力ユニット関数の終了処理を行います。
この関数コール後は、アナログ入出力ユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Aio_Open 関数

機能 アナログ入出力ユニットをオープンします

書式 int ACat_Aio_Open(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したアナログ入出力ユニットをオープンします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後、指定したアナログ入出力ユニットにアクセス可能となります。
アナログ入出力ユニットを使用する際には、必ずコールする必要があります。

ACat_Aio_Close 関数

機能 アナログ入出力ユニットをクローズします

書式 int ACat_Aio_Close(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したアナログ入出力ユニットをクローズします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後は、指定したアナログ入出力ユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Aio_Read 関数

機能

アナログ入出力ユニットから 16 ビットの AD データを取得します

書式

```
int ACat_Aio_Read(DWORD SlaveID, DWORD ch, DWORD len, char* pRead)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-4)
len : 読込データサイズ
pRead : 読込データ

AD データ

```
typedef struct {  
    unsigned short data;  
    unsigned short conn;  
} TADDATA, *PTADDATA;
```

data : アナログ入力データ
conn : 断線検出 [0: 断線なし、1: 断線あり]

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットの指定した AD チャンネルから 16 ビットデータを取得します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

本関数はモードが「サンプリングモード」時のみ有効です。

モードに関しては ACat_Aio_AD_SetMode() を参照してください。

取得した AD データは、TADDATA の形で格納されます。断線検出は、電流モード、4~20mA の設定時のみ有効です。

引数 len には、TADDATA 型のサイズを指定してください。

ACat_Aio_Write 関数

機能

アナログ入出力ユニットへ 16 ビットの DA データを出力します

書式

```
int ACat_Aio_Write(DWORD SlaveID, DWORD ch, DWORD len, char* pWrite)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-4)
len : 書込データサイズ
pWrite : 書込データ

DA データ

```
typedef struct {  
    unsigned short data;  
} TDADATA, *PTDADATA;
```

`data` : アナログ出力データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットの指定した DA チャンネルに 16 ビットデータを出力します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

本関数はモードが「サンプリングモード」時のみ有効です。

モードに関しては ACat_Aio_AD_SetMode() を参照してください。

出力する DA データは、TDADATA の形で格納してください。

引数 len には、TDADATA 型のサイズを指定してください。

ACat_Aio_AD_GetParam 関数

機能 アナログ入出力ユニットの AD チャンネルの設定値を取得します

書式 int ACat_Aio_AD_GetParam(DWORD SlaveID, DWORD ch, WORD* pMode, WORD* pFilter, WORD* pMinMaxSw, WORD* pSamplingNum, WORD* pCalibState)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 pMode : モード
 pFilter : フィルタ
 pMinMaxSw : 最大/最小除外
 pSamplingNum : サンプルング回数
 pCalibState : キャリブレーション

モード

pMode	内容
AIO_AD_MODE_SAMPLING	サンプルングモード (初期値)
AIO_AD_MODE_CONFIG	設定モード
AIO_AD_MODE_CALIBRATION	キャリブレーションモード
AIO_AD_MODE_DISABLE_CH	チャンネル無効

フィルタタイプ

pFilter	内容
AIO_AD_FILTYPE_DISABLE	フィルタ なし (初期値)
AIO_AD_FILTYPE_SIMPLE	フィルタ 単純平均
AIO_AD_FILTYPE_MOVEAVG	フィルタ 移動平均

最大/最小除外

pMinMaxSw	内容
AIO_AD_ENABLE_MINMAX	最大/最小値除外しない (初期値)
AIO_AD_DISABLE_MINMAX	最大/最小値除外する

サンプル回数 (0-6)

2 の (pSamplingNum + 1) 乗で設定される (初期値=0)

キャリブレーション

pCalibState	内容
AIO_AD_END_CALIBRATION	キャリブレーション終了 (初期値)
AIO_AD_MIN_CALIBRATION	最小値登録
AIO_AD_MAX_CALIBRATION	最大値登録

戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したアナログ入出力ユニットの指定した AD チャンネルから現在のユニットの設定値を取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Aio_AD_SetMode 関数

機能 アナログ入出力ユニットの AD チャンネルのモードを変更します

書式 int ACat_Aio_AD_SetMode(DWORD SlaveID, DWORD ch, WORD mode)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 mode : モード

モード

pMode	内容
AIO_AD_MODE_SAMPLING	サンプリングモード (初期値)
AIO_AD_MODE_CONFIG	設定モード
AIO_AD_MODE_CALIBRATION	キャリブレーションモード
AIO_AD_MODE_DISABLE_CH	チャンネル無効

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットの指定した AD チャンネルの現在のモードを変更します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Aio_AD_SetParam 関数

機能 アナログ入出力ユニットの AD に関するパラメータ設定を行います

書式 int ACat_Aio_AD_SetParam(DWORD SlaveID, DWORD ch, WORD Filter, WORD MinMaxSw, WORD SamplingNum)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 Filter : フィルタタイプ
 MinMaxSw : 最大/最小値除外
 SamplingNum : サンプル回数 (0-6)

フィルタタイプ

Filter	内容
AIO_AD_FILTYPE_DISABLE	フィルタ なし (初期値)
AIO_AD_FILTYPE_SIMPLE	フィルタ 単純平均
AIO_AD_FILTYPE_MOVEAVG	フィルタ 移動平均

最大/最小除外

MinMaxSw	内容
AIO_AD_ENABLE_MINMAX	最大/最小値除外しない (初期値)
AIO_AD_DISABLE_MINMAX	最大/最小値除外する

サンプル回数 (0-6)

2 の (SamplingNum + 1) 乗で設定される (初期値=0)

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットの AD に関するパラメータ設定を行います。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。本関数はモードが「設定モード」時のみ有効です。モードに関しては ACat_Aio_AD_SetMode() を参照してください。

ACat_Aio_AD_SetCalib 関数

機能 アナログ入出力ユニットの AD のキャリブレーションを設定します

書式 int ACat_Aio_AD_SetCalib(DWORD SlaveID, DWORD ch, DWORD CalibState)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 CalibState : キャリブレーション

キャリブレーション

CalibState	内容
AIO_AD_END_CALIBRATION	キャリブレーション終了 (初期値)
AIO_AD_MIN_CALIBRATION	最小値登録
AIO_AD_MAX_CALIBRATION	最大値登録

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットのキャリブレーション設定を行います。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。本関数はモードが「キャリブレーションモード」時のみ有効です。モードに関しては ACat_Aio_AD_SetMode() を参照してください。

本関数を呼び出した際、その時に入力されている AD 値を引数で指定された最小値/最大値の値として登録します。キャリブレーション終了後は引数を「キャリブレーション終了」として本関数を実行してください。

ACat_Aio_DA_GetParam 関数

機能 アナログ入出力ユニットの DA チャンネルの設定値を取得します

書式 int ACat_Aio_DA_GetParam(DWORD SlaveID, DWORD ch, WORD* pMode, WORD* pErrorOutput, WORD* pErrorOutputData, WORD* pCalibState)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 pMode : モード
 pErrorOutput : 異常時出力
 pErrorOutputData : 異常時出力データ
 pCalibState : キャリブレーション

モード

pMode	内容
AIO_DA_MODE_OUTPUT	D/A 出力モード (初期値)
AIO_DA_MODE_CONFIG	設定モード
AIO_DA_MODE_CALIBRATION	キャリブレーションモード
AIO_DA_MODE_DISABLE_CH	チャンネル無効

異常時出力

pErrorOutput	内容
AIO_DA_ERROUTPUT_0	0V 出力 (初期値)
AIO_DA_ERROUTPUT_KEEP	出力データ保持
AIO_DA_ERROUTPUT_LOWLIMIT	LowLimit データ出力
AIO_DA_ERROUTPUT_HIGHLIMIT	HighLimit データ出力
AIO_DA_ERROUTPUT_USERNUM	ユーザー設定データ出力

異常時出力データ

pErrorOutputData	内容
AIO_DA_ERROUTPUT_RANGE_MIN	最小設定 (初期値)
AIO_DA_ERROUTPUT_RANGE_MAX	最大設定

キャリブレーション

pCalibState	内容
AIO_DA_END_CALIBRATION	キャリブレーション終了 (初期値)
AIO_DA_MIN_CALIBRATION	最小値登録
AIO_DA_MAX_CALIBRATION	最大値登録

戻り値	ERROR_OK	: 正常
	ERROR_NOTOPEN	: オープンされていません
	ERROR_INVALIDPARAM	: 無効な引数です

説明 スレーブ ID で指定したアナログ入出力ユニットの指定した DA チャンネルから現在のユニットの設定値を取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Aio_DA_SetMode 関数

機能 アナログ入出力ユニットの DA チャンネルのモードを変更します

書式 int ACat_Aio_DA_SetMode(DWORD SlaveID, DWORD ch, WORD mode)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 mode : モード

モード

mode	内容
AIO_DA_MODE_OUTPUT	D/A 出力モード (初期値)
AIO_DA_MODE_CONFIG	設定モード
AIO_DA_MODE_CALIBRATION	キャリブレーションモード
AIO_DA_MODE_DISABLE_CH	チャンネル無効

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットの指定した DA チャンネルの現在のモードを変更します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Aio_DA_SetParam 関数

機能 アナログ入出力ユニットの DA に関するパラメータ設定を行います

書式 int ACat_Aio_DA_SetParam(DWORD SlaveID, DWORD ch, WORD ErrorOutput, WORD ErrorOutputData)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 ErrorOutput : 異常時出力
 ErrorOutputData : 異常時出力データ

異常時出力

ErrorOutput	内容
AIO_DA_ERROUTPUT_0	0V 出力 (初期値)
AIO_DA_ERROUTPUT_KEEP	出力データ保持
AIO_DA_ERROUTPUT_LOWLIMIT	LowLimit データ出力
AIO_DA_ERROUTPUT_HIGHLIMIT	HighLimit データ出力
AIO_DA_ERROUTPUT_USERNUM	ユーザー設定データ出力

異常時出力データ

ErrorOutputData	内容
AIO_DA_ERROUTPUT_RANGE_MIN	最小設定 (初期値)
AIO_DA_ERROUTPUT_RANGE_MAX	最大設定

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットの DA に関するパラメータ設定を行います。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。本関数はモードが「設定モード」時のみ有効です。モードに関しては ACat_Aio_DA_SetMode() を参照してください。異常時出力データは異常時出力の設定がユーザー設定データ出力の時のみ有効です。

ACat_Aio_DA_SetCalib 関数

機能 アナログ入出力ユニットの DA のキャリブレーションを設定します

書式 int ACat_Aio_DA_SetCalib(DWORD SlaveID, DWORD ch, WORD CalibState)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 CalibState : キャリブレーション

キャリブレーション

pCalibState	内容
AIO_DA_END_CALIBRATION	キャリブレーション終了 (初期値)
AIO_DA_MIN_CALIBRATION	最小値登録
AIO_DA_MAX_CALIBRATION	最大値登録

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したアナログ入出力ユニットのキャリブレーション設定を行います。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。本関数はモードが「キャリブレーションモード」時のみ有効です。モードに関しては ACat_Aio_DA_SetMode() を参照してください。

本関数を呼び出した際、その時に出力している DA 値を引数で指定された最小値/最大値の値として登録します。キャリブレーション終了後は引数を「キャリブレーション終了」として本関数を実行してください。

ACat_Aio_SoftReset 関数

機能	アナログ入出力ユニットのソフトウェアリセットを行います
書式	int ACat_Aio_SoftReset (DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定したアナログ入出力ユニットのソフトウェアリセットを行います。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

2-9 モーションコントローラユニット関数

ACat_Axis_Create 関数

機能	モーションコントローラユニット関数群を初期化します
書式	int ACat_Axis_Create(void)
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にあるモーションコントローラユニット関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 本関数コール後、モーションコントローラユニットにアクセス可能となります。 モーションコントローラユニットを使用する際には、必ずコールする必要があります。

ACat_Axis_Destroy 関数

- 機能** モーションコントローラユニット関数群を終了します
- 書式** `int ACat_Axis_Destroy(void)`
- 引数** なし
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** ライブラリ内にあるモーションコントローラユニット関数の終了処理を行います。この関数コール後は、モーションコントローラユニットにアクセス不可になります。アプリケーションの終了時に必ずコールする必要があります。

ACat_Axis_Open 関数

機能

モーションコントローラユニットをオープンします

書式

```
int ACat_Axis_Open(DWORD SlaveID)
```

引数

SlaveID : スレーブ ID

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したモーションコントローラユニットをオープンします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後、指定したモーションコントローラユニットにアクセス可能となります。
モーションコントローラユニットを使用する際には、必ずコールする必要があります。

ACat_Axis_Close 関数

機能

モーションコントローラユニットをクローズします

書式

int ACat_Axis_Close(DWORD SlaveID)

引数

SlaveID : スレーブ ID

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したモーションコントローラユニットをクローズします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後は、指定したモーションコントローラユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Axis_Read 関数

機能

モーションコントローラユニットから入力データを取得します

書式

```
int ACat_Axis_Read(DWORD SlaveID, DWORD offset, DWORD len, char* pRead)
```

引数

SlaveID : スレーブ ID
offset : PDO オフセット
len : 読込データサイズ
pRead : 読込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したモーションコントローラユニットから入力データを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Axis_Write 関数

- 機能** モーションコントローラユニットヘデータを出力します
- 書式** `int ACat_Axis_Write(DWORD SlaveID, DWORD offset, DWORD len, char* pWrite)`
- 引数**
- | | |
|---------|-------------|
| SlaveID | : スレーブ ID |
| offset | : PDO オフセット |
| len | : 書込データサイズ |
| pWrite | : 書込データ |
- 戻り値**
- | | |
|---------------|-----------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード(エラーコード一覧を参照) |
- 説明** スレーブ ID で指定したモーションコントローラユニットヘデータを出力します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Axis_GetParam 関数

機能

モーションコントローラユニットから設定パラメータを取得します

書式

```
int ACat_Axis_GetParam(DWORD SlaveID, WORD ParamIdx, WORD ParamSubIdx, DWORD len,
char* pRead)
```

引数

SlaveID : スレーブ ID
ParamIdx : COE インデックス
ParamSubIdx : COE サブインデックス
len : 読み込みデータサイズ
pRead : 読み込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したモーションコントローラユニットから設定パラメータを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Axis_SetParam 関数

機能

モーションコントローラユニットから設定パラメータを設定します

書式

```
int ACat_Axis_SetParam(DWORD SlaveID, WORD ParamIdx, WORD ParamSubIdx, DWORD len,
char* pWrite)
```

引数

SlaveID : スレーブ ID
ParamIdx : COE インデックス
ParamSubIdx : COE サブインデックス
len : 書込みデータサイズ
pWrite : 書込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したモーションコントローラユニットへ設定パラメータを設定します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Axis_SoftReset 関数

機能

モーションコントローラユニットのソフトウェアリセットを行います

書式

int ACat_Axis_SoftReset (DWORD SlaveID)

引数

SlaveID : スレーブ ID

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したモーションコントローラユニットのソフトウェアリセットを行います。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

2-10 エンコーダユニット関数

ACat_Enc_Create 関数

機能 エンコーダユニット関数群を初期化します

書式 int ACat_Enc_Create(void)

引数 なし

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 ライブラリ内にあるエンコーダユニット関数の初期化を行います。
この関数コールは EtherCAT 通信開始後しか使用できません。
この関数コール後、エンコーダユニットにアクセス可能となります。
エンコーダユニットを使用する際には、必ずコールする必要があります。

ACat_Enc_Destroy 関数

機能 エンコーダユニット関数群を終了します

書式 int ACat_Enc_Destroy(void)

引数 なし

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 ライブラリ内にあるエンコーダユニット関数の終了処理を行います。
この関数コール後は、エンコーダユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Enc_Open 関数

機能 エンコーダユニットをオープンします

書式 int ACat_Enc_Open(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したエンコーダユニットをオープンします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後、指定したエンコーダユニットにアクセス可能となります。
エンコーダユニットを使用するには、必ずコールする必要があります。

ACat_Enc_Close 関数

機能 エンコーダユニットをクローズします

書式 int ACat_Enc_Close(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したエンコーダユニットをクローズします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後は、指定したエンコーダユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Enc_Read 関数

機能 エンコーダユニットから入力データを取得します

書式 int ACat_Enc_Read(DWORD SlaveID, DWORD ch, TENCSTATE* pEncState)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-2)
pEncState	: エンコーダステータス

エンコーダステータス

```
typedef struct {  
    WORD    Status;  
    DWORD   CurCount;  
    DWORD   Latch_A;  
    DWORD   Latch_B;  
} TENCSTATE, *PENCSTATE;
```

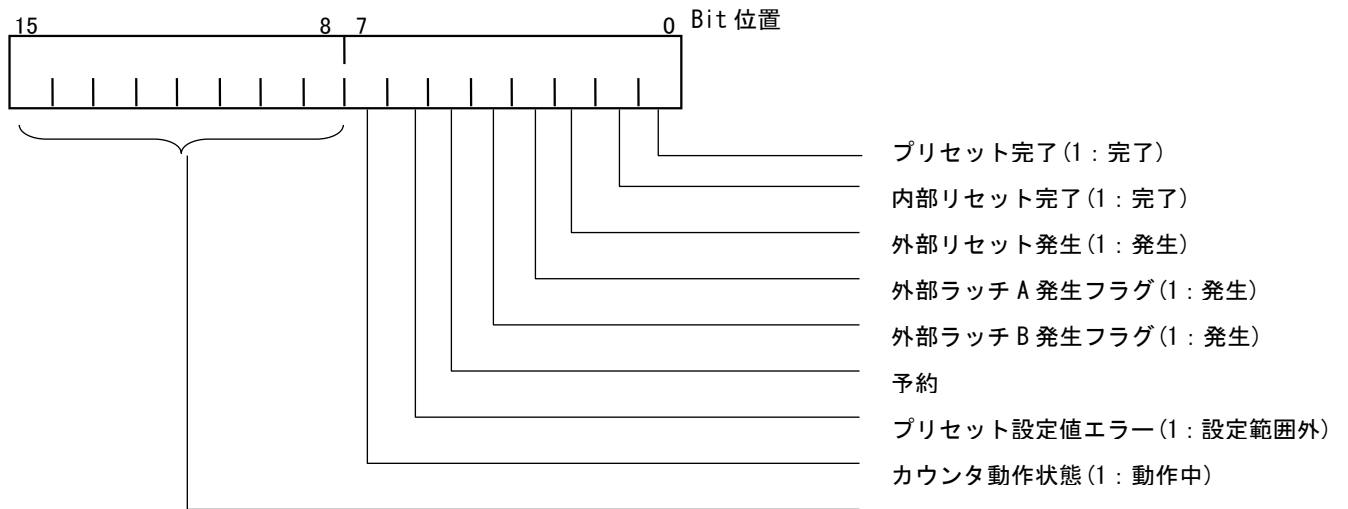
CurCount : カウンタ現在値
Latch_A : ラッチ A 値
Latch_B : ラッチ B 値
Status : ステータス (詳細は次ページ参照)

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定したエンコーダユニットから入力データを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ステータスの意味(下図参照)



ACat_Enc_Write 関数

機能 エンコーダユニットヘデータを出力します

書式 int ACat_Enc_Write(DWORD SlaveID, DWORD ch, TENCCommand* pEncCmd)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-2)
pEncCmd	: コマンド

エンコーダコマンド

```
typedef struct {  
    WORD    SoftSw;  
    DWORD   Preset;  
} TENCCommand, *PENCCommand;
```

SoftSw : ソフト Sw (詳細は次ページ参照)

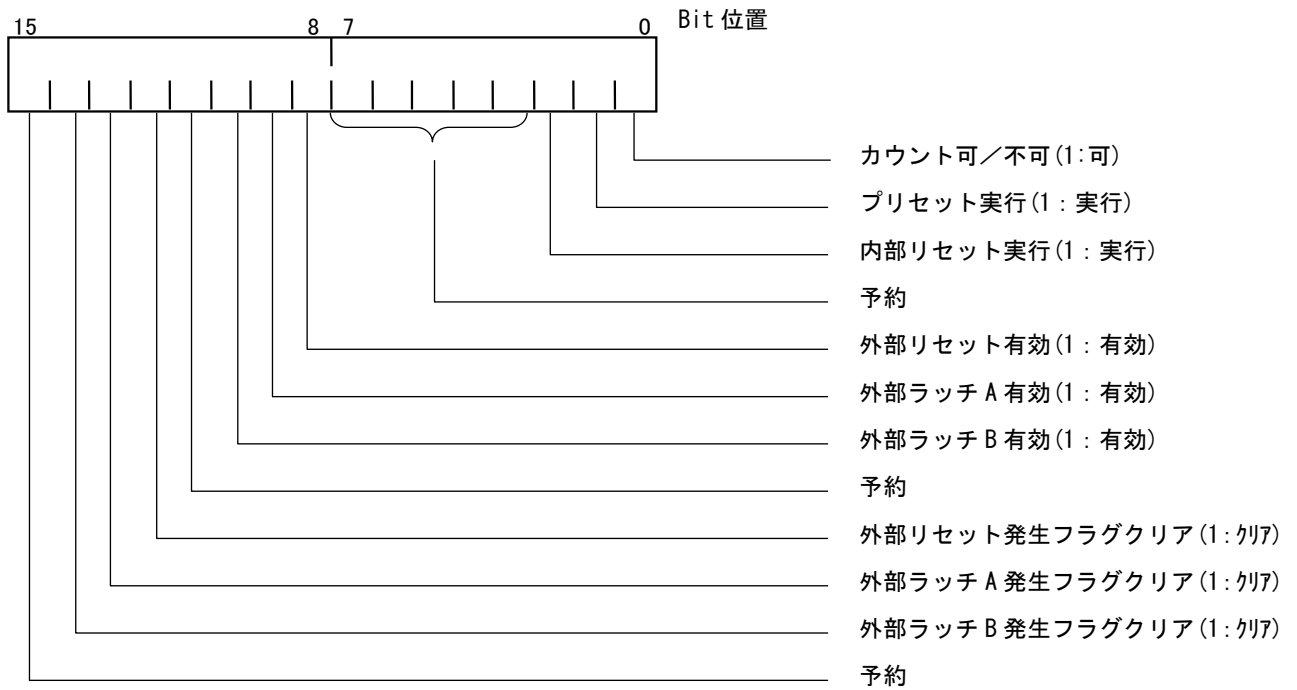
Preset : プリセット値

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定したエンコーダユニットヘデータを出力します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ソフト Sw の意味 (下図参照)



ACat_Enc_GetParam 関数

機能 エンコーダユニットの設定パラメータを取得します

書式 int ACat_Enc_GetParam(DWORD SlaveID, DWORD ch, TENC PARA* pEncPara)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-2)
pEncPara	: パラメータ

エンコーダパラメータ

```
typedef struct {  
    WORD    SoftSw;  
    DWORD   Preset;  
    BYTE    Direction;  
    BYTE    Evaluation;  
    DWORD   MaxCount;  
} TENC PARA, *PENC PARA;
```

SoftSw : ソフト Sw

Preset : プリセット値

Direction : 回転方向

Evaluation : 入力方式 (通倍)

MaxCount : リングカウンタ最大値

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定したエンコーダユニットから設定パラメータを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Enc_SetDirection 関数

機能 エンコーダユニットの回転方向を設定します

書式 int ACat_Enc_SetDirection(DWORD SlaveID, DWORD ch, BYTE Direction)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 Direction : 回転方向

回転方向

Direction	内容
ENC_RND_CW	CW (初期値)
ENC_RND_CCW	CCW

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したエンコーダユニットの回転方向を設定します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Enc_SetEdgeEvaluation 関数

機能 エンコーダユニットの入力方式（通倍）を設定します

書式 int ACat_Enc_SetEdgeEvaluation(DWORD SlaveID, DWORD ch, BYTE Evaluation)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 Evaluation : 入力方式 (通倍)

入力方式 (通倍)

Evaluation	内容
IN_TYPE_1TMS	A-B 相位相パルス 1 通倍
IN_TYPE_2TMS	A-B 相位相パルス 2 通倍
IN_TYPE_4TMS	A-B 相位相パルス 4 通倍
IN_TYPE_PLS	符号+パルス
IN_TYPE_PLMN	加算・減算

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定したエンコーダユニットの入力方式（通倍）を設定します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Enc_SetMaxRingCount 関数

- 機能** エンコーダユニットのリングカウンタ最大値を設定します
- 書式** int ACat_Enc_SetMaxRingCount (DWORD SlaveID, DWORD ch, DWORD MaxRingCount)
- 引数** SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
MaxRingCount : リングカウンタ最大値 (0-4, 294, 967, 295)
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** スレーブ ID で指定したエンコーダユニットのリングカウンタ最大値を設定します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Enc_SaveParam 関数

機能 エンコーダユニットのパラメータを保存します

書式 int ACat_Enc_SaveParam(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 エンコーダユニットの回転方向、入力方式、リングカウンタ最大値を保存します。
ACat_Enc_SetDirection、ACat_Enc_SetEdgeEvaluation、ACat_Enc_SetMaxRingCount 関数で
パラメータを設定した後に実行してください。

ACat_Enc_LoadParam 関数

機能	エンコーダユニットのパラメータを初期化します
書式	int ACat_Enc_SaveParam(DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	エンコーダユニットのパラメータを初期化します。

ACat_Enc_SoftReset 関数

機能 エンコーダユニットのソフトウェアリセットを行います

書式 int ACat_Enc_SoftReset (DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定したエンコーダユニットのソフトウェアリセットを行います。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

2-1-1 SIO ゲートウェイユニット関数

ACat_Sio_Create 関数

機能

SIO ゲートウェイユニット関数群を初期化します

書式

```
int ACat_Sio_Create(void)
```

引数

なし

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

ライブラリ内にある SIO ゲートウェイユニット関数の初期化を行います。
この関数コールは EtherCAT 通信開始後しか使用できません。
この関数コール後、Open 関数をコールすることで SIO ゲートウェイユニットにアクセス可能となります。
SIO ゲートウェイユニットを使用する際には、必ずコールする必要があります。

ACat_Sio_Destroy 関数

- 機能** SIO ゲートウェイユニット関数群を終了します
- 書式** int ACat_Sio_Destroy(void)
- 引数** なし
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** ライブラリ内にある SIO ゲートウェイユニット関数の終了処理を行います。
この関数コール後は、SIO ゲートウェイユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Sio_Open 関数

機能	SIO ゲートウェイユニットをオープンします	
書式	int ACat_Sio_Open(DWORD SlaveID, DWORD ch, DWORD sw)	
引数	SlaveID	: スレーブ ID
	ch	: チャンネル番号 (1-4)
	sw	: RS422/485 選択 sw
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した SIO ゲートウェイユニットをオープンします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後、指定した SIO ゲートウェイユニットにアクセス可能となります。 SIO ゲートウェイユニットを使用する際には、必ずコールする必要があります。	

ACat_Sio_Close 関数

機能	SIO ゲートウェイユニットをクローズします
書式	int ACat_Sio_Close(DWORD SlaveID, DWORD ch)
引数	SlaveID : スレーブ ID ch : チャンネル番号 (1-4)
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した SIO ゲートウェイユニットをクローズします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後は、指定した SIO ゲートウェイユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_Sio_Read 関数

機能 シリアル通信でバイト列を受信します

書式 int ACat_Sio_Read(DWORD SlaveID, DWORD ch, DWORD len, char* pRead, DWORD* plen, DWORD timeout)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-4)
len	: 受信するデータのサイズ
pRead	: 受信データを格納するバッファの先頭ポインタ
plen	: 実際に受信したデータのサイズを格納するポインタ
timeout	: タイムアウトまでの時間 [ms] (0 の時無限待ち)

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した SIO ゲートウェイユニットのチャンネルの受信バッファからデータを取得します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

データ受信が成功した場合は、受信したデータのサイズを plen に格納し [ACAT_ER_OK] を返します。

各種エラーが生じた場合は、エラー発生までに受信したデータサイズを plen に格納します。

[AL_ER_COMMAND] で通信エラー状態となった場合は、ClearError 関数でエラーを解除するようにして下さい。

ACat_Sio_Write 関数

機能 シリアル通信でバイト列を送信します

書式 int ACat_Sio_Write(DWORD SlaveID, DWORD ch, DWORD len, char* pWrite, DWORD* plen, DWORD timeout)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-4)
len	: 送信するデータのサイズ
pWrite	: 送信するデータバッファの先頭ポインタ
plen	: 実際に送信したデータのサイズを格納するポインタ
timeout	: タイムアウトまでの時間 [ms] (0 の時無限待ち)

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した SIO ゲートウェイユニットを介したシリアル通信を用いてデータの送信を行います。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

データ送信が成功した場合は、送信したデータのサイズを plen に格納し [ACAT_ER_OK] を返します。

各種エラーが生じた場合は、エラー発生までに送信したデータサイズを plen に格納します。

[AL_ER_COMMAND] で通信エラー状態となった場合は、ClearError 関数でエラーを解除するようにして下さい。

ACat_Sio_GetCommInfo 関数

機能 シリアル通信の通信設定を行います

書式 int ACat_Sio_GetCommInfo(DWORD SlaveID, DWORD ch, TPORTINFO* pPortInfo)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 pPortInfo : 通信パラメータ

通信パラメータ

```
typedef struct {
    BYTE  RsType;
    BYTE  Open;
    BYTE  Baudrate;
    BYTE  Length;
    BYTE  Stopbit;
    BYTE  Parity;
    BYTE  CtsFlow;
    BYTE  RtsFlow;
    BYTE  OutXFlag;
    BYTE  InXFlag;
    BYTE  XonChar;
    BYTE  XoffChar;
} TPORTINFO, *PPORTINFO;
```

RsType : 通信タイプ
 Open : オープンフラグ
 Baudrate : 通信ボーレート
 Length : データ長
 Stopbit : ストップビット
 Parity : パリティ
 CtsFlow : CTS フロー制御
 RtsFlow : RTS フロー制御
 OutXFlag : Xon/Xoff 出力フロー制御
 InXFlag : Xon/Xoff 入力フロー制御
 XonChar : Xon 文字
 XoffChar : Xoff 文字

通信タイプ

RsType	内容
SIO_CONF_COMMTYPE_RS232	通信タイプ RS232 (初期値)
SIO_CONF_COMMTYPE_RS422	通信タイプ RS422 (初期値)
SIO_CONF_COMMTYPE_RS485	通信タイプ RS485

オープンフラグ

Open	内容
SIO_CONF_OPENFLAG_ON	オープン
SIO_CONF_OPENFLAG_OFF	クローズ (初期値)

通信ボーレート

Baudrate	内容
SIO_CONF_BAUDRATE_1200	1200 bps
SIO_CONF_BAUDRATE_2400	2400 bps
SIO_CONF_BAUDRATE_4800	4800 bps
SIO_CONF_BAUDRATE_9600	9600 bps (初期値)
SIO_CONF_BAUDRATE_19200	19200 bps
SIO_CONF_BAUDRATE_38400	38400 bps
SIO_CONF_BAUDRATE_57600	57600 bps
SIO_CONF_BAUDRATE_115200	115200 bps

データ長

Length	内容
SIO_CONF_DATALEN_7BIT	7bit
SIO_CONF_DATALEN_8BIT	8bit (初期値)

ストップビット

Stopbit	内容
SIO_CONF_STOPBIT_1BIT	1bit (初期値)
SIO_CONF_STOPBIT_2BIT	2bit

パリティ

Parity	内容
SIO_CONF_PARITY_NON	なし (初期値)
SIO_CONF_PARITY_EVEN	偶数
SIO_CONF_PARITY_ODD	奇数

CTS フロー制御

CtsFlow	内容
SIO_CONF_CTSFLOW_ON	ON
SIO_CONF_CTSFLOW_OFF	OFF (初期値)

RTS フロー制御

RtsFlow	内容
SIO_CONF_RTSFLOW_ON	ON
SIO_CONF_RTSFLOW_OFF	OFF (初期値)

Xon/Xoff 出力フロー制御

OutXFlag	内容
SIO_CONF_XOUTFLOW_ON	ON
SIO_CONF_XOUTFLOW_OFF	OFF (初期値)

Xon/Xoff 入力フロー制御

InXFlag	内容
SIO_CONF_XINFLOW_ON	ON
SIO_CONF_XINFLOW_OFF	OFF (初期値)

戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明	スレーブ ID で指定した SIO ゲートウェイユニットの通信設定を取得します。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
-----------	--

ACat_Sio_SetCommInfo 関数

機能 シリアル通信の通信設定を行います

書式 int ACat_Sio_SetCommInfo(DWORD SlaveID, DWORD ch, TPORTINFO* pPortInfo)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-4)
pPortInfo	: 通信パラメータ

通信パラメータ

```
typedef struct {  
    BYTE    RsType;  
    BYTE    Open;  
    BYTE    Baudrate;  
    BYTE    Length;  
    BYTE    Stopbit;  
    BYTE    Parity;  
    BYTE    CtsFlow;  
    BYTE    RtsFlow;  
    BYTE    OutXFlag;  
    BYTE    InXFlag;  
    BYTE    XonChar;  
    BYTE    XoffChar;  
} TPORTINFO, *PPORTINFO;
```

RsType	: 通信タイプ
Open	: オープンフラグ
Baudrate	: 通信ボーレート
Length	: データ長
Stopbit	: ストップビット
Parity	: パリティ
CtsFlow	: CTS フロー制御
RtsFlow	: RTS フロー制御
OutXFlag	: Xon/Xoff 出力フロー制御
InXFlag	: Xon/Xoff 入力フロー制御
XonChar	: Xon 文字
XoffChar	: Xoff 文字

通信タイプ

RsType	内容
SIO_CONF_COMMTYPE_RS232	通信タイプ RS232 (初期値)
SIO_CONF_COMMTYPE_RS422	通信タイプ RS422 (初期値)
SIO_CONF_COMMTYPE_RS485	通信タイプ RS485

オープンフラグ

Open	内容
SIO_CONF_OPENFLAG_ON	オープン
SIO_CONF_OPENFLAG_OFF	クローズ (初期値)

通信ボーレート

Baudrate	内容
SIO_CONF_BAUDRATE_1200	1200 bps
SIO_CONF_BAUDRATE_2400	2400 bps
SIO_CONF_BAUDRATE_4800	4800 bps
SIO_CONF_BAUDRATE_9600	9600 bps (初期値)
SIO_CONF_BAUDRATE_19200	19200 bps
SIO_CONF_BAUDRATE_38400	38400 bps
SIO_CONF_BAUDRATE_57600	57600 bps
SIO_CONF_BAUDRATE_115200	115200 bps

データ長

Length	内容
SIO_CONF_DATALEN_7BIT	7bit
SIO_CONF_DATALEN_8BIT	8bit (初期値)

ストップビット

Stopbit	内容
SIO_CONF_STOPBIT_1BIT	1bit (初期値)
SIO_CONF_STOPBIT_2BIT	2bit

パリティ

Parity	内容
SIO_CONF_PARITY_NON	なし (初期値)
SIO_CONF_PARITY_EVEN	偶数
SIO_CONF_PARITY_ODD	奇数

CTS フロー制御

CtsFlow	内容
SIO_CONF_CTSFLOW_ON	ON
SIO_CONF_CTSFLOW_OFF	OFF (初期値)

RTS フロー制御

RtsFlow	内容
SIO_CONF_RTSFLOW_ON	ON
SIO_CONF_RTSFLOW_OFF	OFF (初期値)

Xon/Xoff 出力フロー制御

OutXFlag	内容
SIO_CONF_XOUTFLOW_ON	ON
SIO_CONF_XOUTFLOW_OFF	OFF (初期値)

Xon/Xoff 入力フロー制御

InXFlag	内容
SIO_CONF_XINFLOW_ON	ON
SIO_CONF_XINFLOW_OFF	OFF (初期値)

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した SIO ゲートウェイユニットの通信設定を行います。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

SIO ゲートウェイユニットを使用する場合は必ず呼ぶようにします。
 受信、送信のバッファはデフォルトで 4096 となっています。これ以下を指定した場合、バッファサイズは 4096 に設定されます。また、4096 を超えた値を指定した場合は 40960 となります。受信バッファはシステムで唯一のバッファとして作成されるため、複数プロセスでの使用の場合は、必ず同じサイズを指定するようにして下さい。

ACat_Sio_ClearError 関数

機能 SIO ゲートウェイユニットの通信エラー状態を解除します

書式 int ACat_Sio_ClearError (DWORD SlaveID, DWORD ch)

引数 SlaveID : スレーブ ID
ch : チャンネル番号 (1-4)

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した SIO ゲートウェイユニットの通信エラーを解除します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
エラー解除を実行する際には、データ送受信を終了してから行って下さい。

ACat_Sio_GetCommState 関数

機能 SIO ゲートウェイユニットの通信ステータスを取得します

書式 int ACat_Sio_GetCommState(DWORD SlaveID, DWORD ch, TRSSTATUS* pRsState)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 pRsState : 通信ステータス

通信ステータス

```
typedef struct {
    union {
        struct {
            BYTE    OverFlow    :1;
            BYTE    ParityErr   :1;
            BYTE    FramingErr  :1;
            BYTE    OverRunErr  :1;
            BYTE    Reserve     :4;
        } Bit;
        BYTE Byte;
    } ErrorStatus;
    union {
        struct {
            BYTE    CtsStatus   :1;
            BYTE    CtsHold    :1;
            BYTE    XoffHold    :1;
            BYTE    Reserve     :5;
        } Bit;
        BYTE    Byte;
    } ComStatus;
    WORD    SendSize;
    WORD    RecvSize;
} TRSSTATUS, *PRSSTATUS;
```

OverFlow : オーバーフローエラー
 ParityErr : パリティエラー
 FramingErr : フレミングエラー
 OverRunErr : オーバーランエラー
 CtsStatus :
 CtsHold :
 XoffHold :
 SendSize : 送信サイズ
 RecvSize : 受信サイズ

戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 DLL 内部の受信バッファに格納されたデータのサイズを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

受信中は不定値となります。

ACat_Sio_PurgeComm 関数

機能 SIO ゲートウェイユニットの送受信バッファに格納されたデータを削除します

書式 int ACat_Sio_PurgeComm(DWORD SlaveID, DWORD ch, DWORD QueSw)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-4)
 QueSw : キュー選択

キュー選択

QueSw	内容
SIO_CMD_CLEAR_SENDBUF	送信バッファ
SIO_CMD_CLEAR_RECVBUF	受信バッファ

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 SIO ゲートウェイユニットの送受信バッファに格納されたデータを削除します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Sio_SaveParam 関数

- 機能** SIO ゲートウェイユニットにパラメータを保存します
- 書式** int ACat_Sio_SaveParam(DWORD SlaveID)
- 引数** SlaveID : スレーブ ID
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** SIO ゲートウェイユニットにパラメータを保存します。
ACat_Sio_SetCommInfo 関数でパラメータを設定した後に実行してください。

ACat_Sio_LoadParam 関数

機能	SIO ゲートウェイユニットのパラメータを初期化します	
書式	int ACat_Sio_LoadParam(DWORD SlaveID)	
引数	SlaveID	: スレーブ ID
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)
説明	SIO ゲートウェイユニットのパラメータを初期化します	

ACat_Sio_SoftReset 関数

機能	SIO ゲートウェイユニットのソフトウェアリセットを行います	
書式	int ACat_Sio_SoftReset (DWORD SlaveID)	
引数	SlaveID	: スレーブ ID
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した SIO ゲートウェイユニットのソフトウェアリセットを行います。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。	

2-12 A-Link ゲートウェイユニット関数

ACat_Alink_Create 関数

機能	A-Link ゲートウェイユニット関数群を初期化します
書式	int ACat_Alink_Create(void)
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある A-Link ゲートウェイユニット関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 A-Link ゲートウェイユニットを使用する際には、必ずコールする必要があります。

ACat_Alink_Destroy 関数

機能	A-Link ゲートウェイユニット関数群を終了します	
書式	int ACat_Alink_Destroy(void)	
引数	なし	
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある A-Link ゲートウェイユニット関数の終了処理を行います。 この関数コール後は、A-Link ゲートウェイユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。	

ACat_Alink_Open 関数

- 機能** A-Link ゲートウェイユニットをオープンします
- 書式** int ACat_Alink_Open(DWORD SlaveID)
- 引数** SlaveID : スレーブ ID
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** スレーブ ID で指定した A-Link ゲートウェイユニットをオープンします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
この関数コール後、指定した A-Link ゲートウェイユニットにアクセス可能となります。
A-Link ゲートウェイユニットを使用する際には、必ずコールする必要があります。

ACat_Alink_Close 関数

機能	A-Link ゲートウェイユニットをクローズします
書式	int ACat_Alink_Close (DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した A-Link ゲートウェイユニットをクローズします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後は、指定した A-Link ゲートウェイユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_Alink_Start 関数

機能 A-Link ゲートウェイユニットの A-Link 通信を開始します

書式 int ACat_Alink_Start(DWORD SlaveID, DWORD ch, DWORD MaxSa, DWORD ScanSw)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 MaxSa : 接続 SA 数 (1-63)
 ScanSw : スキャンスイッチ

スキャンスイッチ

ScanSw	内容
ALINK_SINGLE_SCAN_START	単発スキャン開始
ALINK_CONSTANT_SCAN_START	継続スキャン開始

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した A-Link ゲートウェイユニットの A-Link 通信を開始します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_Stop 関数

- 機能** A-Link ゲートウェイユニットの A-Link 通信を停止します
- 書式** `int ACat_Alink_Stop(DWORD SlaveID, DWORD ch)`
- 引数**
- | | |
|---------|-----------------|
| SlaveID | : スレーブ ID |
| ch | : チャンネル番号 (1-2) |
- 戻り値**
- | | |
|---------------|-----------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード(エラーコード一覧を参照) |
- 説明** スレーブ ID で指定した A-Link ゲートウェイユニットの A-Link 通信を停止します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_Read 関数

機能

A-Link ゲートウェイユニットの A-Link 入力データを取得します

書式

```
int ACat_Alink_Read(DWORD SlaveID, DWORD ch, DWORD Sa, DWORD SaCount, WORD* pRead)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
Sa : 読込開始 SA 番号
SaCount : 読込 SA 数
pRead : 読込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定した A-Link ゲートウェイユニットから入力データを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_Write 関数

機能

A-Link ゲートウェイユニットの A-Link 出力データを出力します

書式

```
int ACat_Alink_Write(DWORD SlaveID, DWORD ch, DWORD Sa, DWORD SaCount, WORD* pWrite)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
Sa : 書込開始 SA 番号
SaCount : 書込 SA 数
pWrite : 書込データ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した A-Link ゲートウェイユニットヘデータを出力します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_ReadChk2occur 関数

機能 A-Link ゲートウェイユニットの A-Link 通信異常 (CHK2 発生状況) を取得します

書式 int ACat_Alink_ReadChk2occur (DWORD SlaveID, DWORD ch, TALINK_CHK2_OCCUR* pRead)

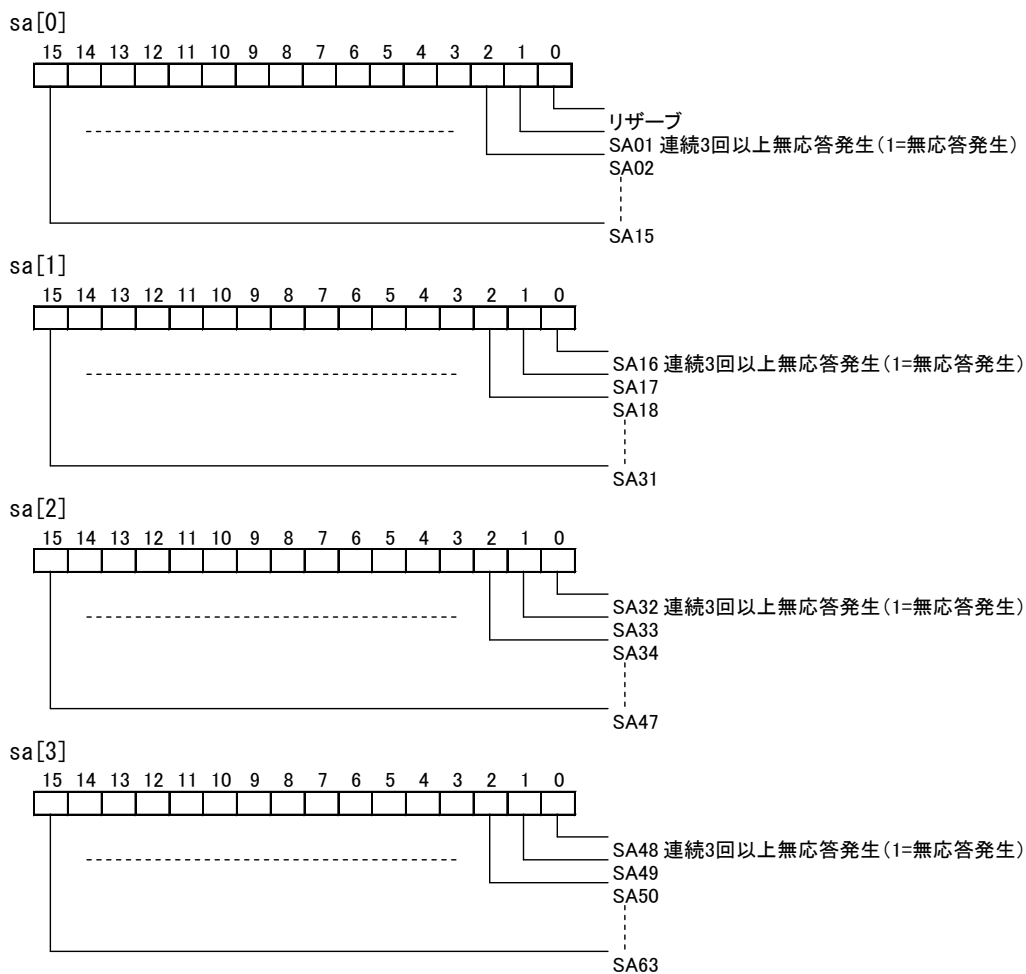
引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 pRead : 読込データ

通信状態データ

```
typedef union {
    unsigned short sa[ALINK_CHK2OCCUR_BUFMAX(=4)]
} TALINK_CHK2_OCCUR, *PALINK_CHK2_OCCUR;
```

sa : CHK2 発生状況



戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明	スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信異常 (CHK2 発生状況) を取得します。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
-----------	---

ACat_Alink_GetStatus 関数

機能

A-Link ゲートウェイユニットからステータスデータ取得を行います

書式

```
int ACat_Alink_GetStatus(DWORD SlaveID, DWORD ch, DWORD Sa, WORD* pRead)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
Sa : 読込開始 SA 番号
pRead : 読込ステータス

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した A-Link ゲートウェイユニットからステータスを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_GetSysSts 関数

機能 A-Link ゲートウェイユニットの A-Link システムステータスを取得します

書式 int ACat_Alink_GetSysSts(DWORD SlaveID, DWORD ch, TALINK_SYS_STATUS* pRead)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 pRead : A-Link システムステータス

通信パラメータ

```
typedef union {
    struct {
        WORD ScanTiming :6;
        WORD Reserve1 :1;
        WORD Scanning :1;
        WORD Reserve2 :1;
        WORD DataReq :1;
        WORD SaNotRes :1;
        WORD Reserve3 :5;
    } Bit;
    WORD Word;
} TALINK_SYS_STATUS, *PALINK_SYS_STATUS;
```

ScanTiming : スキャン詳細タイミング (0-63)
Scanning : スキャン中フラグ (0:未スキャン, 1:スキャン中)
DataReq : データ要求検知フラグ (0:未検知, 1:検知)
SaNotRes : 無応答サテライト検知フラグ (0:未検知, 1:検知)

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link システムステータスを取得します。
 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_GetChk1Count 関数

機能 A-Link ゲートウェイユニットの A-Link 通信異常 (CHK1) 発生回数を取得します

書式 int ACat_Alink_GetChk1Count (DWORD SlaveID, DWORD ch, WORD* pRead)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-2)
pRead	: CHK1 発生回数

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信異常 (CHK1 発生回数) を取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

無応答フラグ (CHK1) とは、1 回でも無応答があれば立ちます。
連続 3 回以上の無応答フラグ (CHK2) とは、無応答状態 (CHK1) が 3 回以上連続で発生した場合に立ちます。
最大 255 回までカウントし、クリアするには ACat_Alink_ClearChk1Count () を使用します。

ACat_Alink_GetChk2Count 関数

機能

A-Link ゲートウェイユニットの A-Link 通信状態 (CHK2) 発生回数を取得します

書式

```
int ACat_Alink_GetChk2Count(DWORD SlaveID, DWORD ch, WORD* pRead)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pRead : CHK2 発生回数

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信異常 (CHK2 発生回数) を取得します。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

無応答フラグ (CHK1) とは、1 回でも無応答があれば立ちます。

連続 3 回以上の無応答フラグ (CHK2) とは、無応答状態 (CHK1) が

3 回以上連続で発生した場合に立ちます。

最大 255 回までカウントし、クリアするには ACat_Alink_ClrChk2Count () を使用します。

ACat_Alink_ClrChk1Count 関数

- 機能** A-Link ゲートウェイユニットの A-Link 通信異常 (CHK1) 発生回数をクリアします
- 書式** `int ACat_Alink_ClrChk1Count(DWORD SlaveID, DWORD ch)`
- 引数**
- | | |
|---------|-----------------|
| SlaveID | : スレーブ ID |
| ch | : チャンネル番号 (1-2) |
- 戻り値**
- | | |
|---------------|------------------------|
| ACAT_ER_OK | : 正常 |
| ACAT_ER_OK 以外 | : エラーコード (エラーコード一覧を参照) |
- 説明** スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信異常 (CHK1 発生回数) をクリアします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_ClrChk2Count 関数

- 機能** A-Link ゲートウェイユニットの A-Link 通信状態 (CHK2) 発生回数をクリアします
- 書式** int ACat_Alink_ClrChk2Count (DWORD SlaveID, DWORD ch)
- 引数** SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)
- 説明** スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信異常 (CHK2 発生回数) をクリアします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_GetCommParam 関数

機能 A-Link ゲートウェイユニットの A-Link 通信設定を取得します

書式 int ACat_Alink_GetCommParam(DWORD SlaveID, DWORD ch, WORD* pDuplex,
WORD* pBaudrate, WORD* pHubNum)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 pDuplex : 通信方式
 pBaudrate : 転送レート
 pHubNum : HUB 挿入数 (0-7)

通信方式

pDuplex	内容
ALINK_DUPLEX_HALF	半 2 重
ALINK_DUPLEX_FULL	全 2 重

転送レート

pBaudrate	内容
ALINK_BAUDRATE_3M	3Mbps
ALINK_BAUDRATE_6M	6Mbps
ALINK_BAUDRATE_12M	12Mbps

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信設定を取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_SetCommParam 関数

機能 A-Link ゲートウェイユニットの A-Link 通信設定を行います

書式 int ACat_Alink_SetCommParam(DWORD SlaveID, DWORD ch, WORD Duplex,
WORD Baudrate, WORD HubNum)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 Duplex : 通信方式
 Baudrate : 転送レート
 HubNum : HUB 挿入数 (0-7)

通信方式

Duplex	内容
ALINK_DUPLEX_HALF	半 2 重
ALINK_DUPLEX_FULL	全 2 重

転送レート

Baudrate	内容
ALINK_BAUDRATE_3M	3Mbps
ALINK_BAUDRATE_6M	6Mbps
ALINK_BAUDRATE_12M	12Mbps

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した A-Link ゲートウェイユニットから A-Link 通信設定を行います。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

ACat_Alink_SoftReset 関数

- 機能** A-Link ゲートウェイユニットのソフトウェアリセットを行います
- 書式** int ACat_Alink_SoftReset (DWORD SlaveID)
- 引数** SlaveID : スレーブ ID
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** スレーブ ID で指定した A-Link ゲートウェイユニットのソフトウェアリセットを行います。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

2-13 Modbus ゲートウェイユニット関数

ACat_Mbg_Create 関数

機能	Modbus ゲートウェイユニット関数群を初期化します
書式	int ACat_Mbg_Create(void)
引数	なし
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある Modbus ゲートウェイユニット関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 Modbus ゲートウェイユニットを使用する際には、必ずコールする必要があります。

ACat_Mbg_Destroy 関数

- 機能** Modbus ゲートウェイユニット関数群を終了します
- 書式** int ACat_Mbg_Destroy(void)
- 引数** なし
- 戻り値** ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** ライブラリ内にある Modbus ゲートウェイユニット関数の終了処理を行います。
この関数コール後は、Modbus ゲートウェイユニットにアクセス不可になります。
アプリケーションの終了時に必ずコールする必要があります。

ACat_Mbg_Open 関数

機能	Modbus ゲートウェイユニットをオープンします
書式	int ACat_Mbg_Open(DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した Modbus ゲートウェイユニットをオープンします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後、指定した Modbus ゲートウェイユニットにアクセス可能となります。 Modbus ゲートウェイユニットを使用するには、必ずコールする必要があります。

ACat_Mbg_Close 関数

機能	Modbus ゲートウェイユニットをクローズします
書式	int ACat_Mbg_Close(DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した Modbus ゲートウェイユニットをクローズします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後は、指定した Modbus ゲートウェイユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_Mbg_ReadByte 関数

機能

Modbus ゲートウェイユニットから Byte 単位で入力データを読み込みます

書式

```
int ACat_Mbg_ReadByte(DWORD SlaveID, DWORD Offset, DWORD Len, BYTE* pRead)
```

引数

SlaveID : スレーブ ID
Offset : オフセット
Len : 読み込みデータ Byte サイズ
pRead : 読み込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットの入力データエリアから、入力データの読み込みを行います。
引数で与えられたオフセットからデータサイズ分のデータを Byte 単位で読み込みを行います。

ACat_Mbg_WriteByte 関数

機能

Modbus ゲートウェイユニットへ Byte 単位で出力データを書込みます

書式

```
int ACat_Mbg_WriteByte(DWORD SlaveID, DWORD Offset, DWORD Len, BYTE* pWrite)
```

引数

SlaveID : スレーブ ID
Offset : オフセット
Len : 書込みデータ Byte サイズ
pWrite : 書込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットの出力データエリアから、出力データの書込みを行います。
引数で与えられたオフセットからデータサイズ分のデータを Byte 単位で書込みを行います。

ACat_Mbg_ReadWord 関数

機能

Modbus ゲートウェイユニットから Word 単位で入力データを読み込みます

書式

```
int ACat_Mbg_ReadWord(DWORD SlaveID, DWORD Offset, DWORD Len, WORD* pRead)
```

引数

SlaveID : スレーブ ID
Offset : オフセット
Len : 読み込みデータ Word サイズ
pRead : 読み込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットの入力データエリアから、入力データの読み込みを行います。
引数で与えられたオフセットからデータサイズ分のデータを Word 単位で読み込みを行います。

ACat_Mbg_WriteWord 関数

機能

Modbus ゲートウェイユニットへ Word 単位で出力データを書込みます

書式

```
int ACat_Mbg_WriteWord(DWORD SlaveID, DWORD Offset, DWORD Len, WORD* pWrite)
```

引数

SlaveID : スレーブ ID
Offset : オフセット
Len : 書込みデータ Word サイズ
pWrite : 書込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットの出力データエリアから、出力データの書込みを行います。
引数で与えられたオフセットからデータサイズ分のデータを Word 単位で書込みを行います。

ACat_Mbg_Start 関数

機能

Modbus ゲートウェイユニットの指定したターゲット機器との通信を開始します

書式

```
int ACat_Mbg_Start(DWORD SlaveID, DWORD TargetID)
```

引数

SlaveID : スレーブ ID
TargetID : ターゲット機器 ID (1-8)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットのターゲット機器 ID で指定したターゲット機器との通信を開始します。

ACat_Mbg_Stop 関数

機能

Modbus ゲートウェイユニットの指定したターゲット機器との通信を停止します

書式

```
int ACat_Mbg_Stop(DWORD SlaveID, DWORD TargetID)
```

引数

SlaveID : スレーブ ID
TargetID : ターゲット機器 ID (1-8)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットのターゲット機器 ID で指定したターゲット機器との通信を停止します。

ACat_Mbg_GetTargetInfo 関数

機能 Modbus ゲートウェイユニットからターゲット機器設定パラメータの取得を行います

書式 int ACat_Mbg_GetTargetInfo(DWORD SlaveID, DWORD TargetID, WORD* TargetAddress, WORD* ComPort)

引数

SlaveID	: スレーブ ID
TargetID	: ターゲット機器 ID (1-8)
TargetAddress	: ターゲット機器アドレス
ComPort	: COM ポート

COM ポート

ComPort	内容
MBG_TARGET_PARAM_COM_NON	0: 未使用
MBG_TARGET_PARAM_COM_1CH	1: 1ch
MBG_TARGET_PARAM_COM_2CH	2: 2ch
MBG_TARGET_PARAM_COM_3CH	3: 3ch
MBG_TARGET_PARAM_COM_4CH	4: 4ch

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットからターゲット機器 ID で指定したターゲット機器の設定パラメータの取得を行います。

ACat_Mbg_SetTargetInfo 関数

機能 Modbus ゲートウェイユニットへターゲット機器設定パラメータの設定を行います

書式 int ACat_Mbg_SetTargetInfo(DWORD SlaveID, DWORD TargetID, WORD TargetAddress, WORD ComPort)

引数

SlaveID	: スレーブ ID
TargetID	: ターゲット機器 ID (1-8)
TargetAddress	: ターゲット機器アドレス
ComPort	: COM ポート

COM ポート

ComPort	内容
MBG_TARGET_PARAM_COM_NON	0: 未使用
MBG_TARGET_PARAM_COM_1CH	1: 1ch
MBG_TARGET_PARAM_COM_2CH	2: 2ch
MBG_TARGET_PARAM_COM_3CH	3: 3ch
MBG_TARGET_PARAM_COM_4CH	4: 4ch

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットへターゲット機器 ID で指定したターゲット機器のパラメータ設定を行います。

ACat_Mbg_GetCommInfo 関数

機能 Modbus ゲートウェイユニットからシリアル通信設定パラメータの取得を行います

書式 int ACat_Mbg_GetCommInfo(DWORD SlaveID, DWORD ComPort, WORD* BaudRate, WORD* DataLength, WORD* StopBit, WORD* Parity, WORD* FormatType, WORD* Interface)

引数

SlaveID : スレーブ ID
 ComPort : COM ポート
 BaudRate : 通信ボーレート
 DataLength : データ長
 StopBit : ストップビット
 Parity : パリティ
 FormatType : 伝送モード
 Interface : シリアルインターフェイス

COM ポート

ComPort	内容
MBG_TARGET_PARAM_COM_NON	0: 未使用
MBG_TARGET_PARAM_COM_1CH	1: 1ch
MBG_TARGET_PARAM_COM_2CH	2: 2ch
MBG_TARGET_PARAM_COM_3CH	3: 3ch
MBG_TARGET_PARAM_COM_4CH	4: 4ch

通信ボーレート

BaudRate	内容
MBG_COMM_BAUDRATE_1200	0: 1200 bps
MBG_COMM_BAUDRATE_2400	1: 2400 bps
MBG_COMM_BAUDRATE_4800	2: 4800 bps
MBG_COMM_BAUDRATE_9600	3: 9600 bps (初期値)
MBG_COMM_BAUDRATE_19200	4: 19200 bps
MBG_COMM_BAUDRATE_38400	5: 38400 bps
MBG_COMM_BAUDRATE_57600	6: 57600 bps
MBG_COMM_BAUDRATE_115200	7: 115200 bps

データ長

DataLength	内容
MBG_COMM_DATALEN_7BIT	0: 7bit
MBG_COMM_DATALEN_8BIT	1: 8bit (初期値)

ストップビット

StopBit	内容
MBG_COMM_STOPBIT_1BIT	0: 1bit (初期値)
MBG_COMM_STOPBIT_2BIT	1: 2bit

パリティ

Parity	内容
MBG_COMM_PARITY_NON	0: なし (初期値)
MBG_COMM_PARITY_EVEN	1: 偶数
MBG_COMM_PARITY_ODD	2: 奇数

伝送モード

FormatType	内容
MBG_COMM_PARAM_MODE_RTU	0: RTU (初期値)
MBG_COMM_PARAM_MODE_ASCII	1: ASCII

シリアルインターフェイス

Interface	内容
MBG_COMM_PARAM_IF_RS232C	0: RS232C (初期値)
MBG_COMM_PARAM_IF_RS422	1: RS422
MBG_COMM_PARAM_IF_RS485	2: RS485

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットからシリアル通信設定パラメータの取得を行います。

「シリアルインターフェイス」の RS422、RS485 の設定は、RS422/RS485 ユニットのみに有効となります。RS422/RS485 ユニット時、「2: RS485」以外は RS422 となります。

ACat_Mbg_SetCommInfo 関数

機能 Modbus ゲートウェイユニットからシリアル通信設定パラメータの設定を行います

書式 int ACat_Mbg_SetCommInfo(DWORD SlaveID, DWORD ComPort, WORD BaudRate, WORD DataLength, WORD StopBit, WORD Parity, WORD FormatType, WORD Interface)

引数

SlaveID : スレーブ ID
 ComPort : COM ポート
 BaudRate : 通信ボーレート
 DataLength : データ長
 StopBit : ストップビット
 Parity : パリティ
 FormatType : 伝送モード
 Interface : シリアルインターフェイス

COM ポート

ComPort	内容
MBG_TARGET_PARAM_COM_NON	0: 未使用
MBG_TARGET_PARAM_COM_1CH	1: 1ch
MBG_TARGET_PARAM_COM_2CH	2: 2ch
MBG_TARGET_PARAM_COM_3CH	3: 3ch
MBG_TARGET_PARAM_COM_4CH	4: 4ch

通信ボーレート

BaudRate	内容
MBG_COMM_BAUDRATE_1200	0: 1200 bps
MBG_COMM_BAUDRATE_2400	1: 2400 bps
MBG_COMM_BAUDRATE_4800	2: 4800 bps
MBG_COMM_BAUDRATE_9600	3: 9600 bps (初期値)
MBG_COMM_BAUDRATE_19200	4: 19200 bps
MBG_COMM_BAUDRATE_38400	5: 38400 bps
MBG_COMM_BAUDRATE_57600	6: 57600 bps
MBG_COMM_BAUDRATE_115200	7: 115200 bps

データ長

DataLength	内容
MBG_COMM_DATALEN_7BIT	0: 7bit
MBG_COMM_DATALEN_8BIT	1: 8bit (初期値)

ストップビット

StopBit	内容
MBG_COMM_STOPBIT_1BIT	0: 1bit (初期値)
MBG_COMM_STOPBIT_2BIT	1: 2bit

パリティ

Parity	内容
MBG_COMM_PARITY_NON	0: なし (初期値)
MBG_COMM_PARITY_EVEN	1: 偶数
MBG_COMM_PARITY_ODD	2: 奇数

伝送モード

FormatType	内容
MBG_COMM_PARAM_MODE_RTU	0: RTU (初期値)
MBG_COMM_PARAM_MODE_ASCII	1: ASCII

シリアルインターフェイス

Interface	内容
MBG_COMM_PARAM_IF_RS232C	0: RS232C (初期値)
MBG_COMM_PARAM_IF_RS422	1: RS422
MBG_COMM_PARAM_IF_RS485	2: RS485

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットへシリアル通信設定パラメータの設定を行います。

「シリアルインターフェイス」の RS422、RS485 の設定は、RS422/RS485 ユニットのみに有効となります。RS422/RS485 ユニット時、「2: RS485」以外は RS422 となります。

ACat_Mbg_GetMonCmd 関数

機能 Modbus ゲートウェイユニットからモニタデータコマンドを取得します

書式 int ACat_Mbg_GetMonCmd(DWORD SlaveID, DWORD CommandID, WORD* TargetID,
WORD* FunctionCode, WORD* StartAddress, WORD* ReadSize,
WORD* Offset, WORD* Cycle)

引数

SlaveID	: スレーブ ID
CommandID	: コマンド ID (1-32)
TargetID	: ターゲット機器 ID (1-8)
FunctionCode	: ファンクションコード
StartAddress	: 開始アドレス (0-65535)
ReadSize	: 読出しレジスタ数 (1-128)
Offset	: PDO オフセット (0-255)
Cycle	: モニタ周期 (1-60000)

ファンクションコード

FunctionCode	内容
MBG_CMD_FUNC_CODE_READ_COIL	0x01: コイル
MBG_CMD_FUNC_CODE_READ_INPUTSTA	0x02: 入力ステータス
MBG_CMD_FUNC_CODE_READ_HOLDREG	0x03: 保持レジスタ
MBG_CMD_FUNC_CODE_READ_INPUTREG	0x04: 入力レジスタ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットからモニタデータコマンドの取得を行います。

ACat_Mbg_SetMonCmd 関数

機能 Modbus ゲートウェイユニットへモニタデータコマンドの設定を行います

書式 int ACat_Mbg_SetMonCmd(DWORD SlaveID, DWORD CommandID, WORD TargetID,
WORD FunctionCode, WORD StartAddress, WORD ReadSize,
WORD Offset, WORD Cycle)

引数

SlaveID	: スレーブ ID
CommandID	: コマンド ID (1-32)
TargetID	: ターゲット機器 ID (1-8)
FunctionCode	: ファンクションコード
StartAddress	: 開始アドレス (0-65535)
ReadSize	: 読出しレジスタ数 (1-128)
Offset	: PDO オフセット (0-255)
Cycle	: モニタ周期 (1-60000)

ファンクションコード

FunctionCode	内容
MBG_CMD_FUNC_CODE_READ_COIL	0x01: コイル
MBG_CMD_FUNC_CODE_READ_INPUTSTA	0x02: 入力ステータス
MBG_CMD_FUNC_CODE_READ_HOLDREG	0x03: 保持レジスタ
MBG_CMD_FUNC_CODE_READ_INPUTREG	0x04: 入力レジスタ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットへモニタデータコマンドの設定を行います。

ACat_Mbg_GetSpotReqCmd 関数

機能 Modbus ゲートウェイユニットから即時要求データコマンドを取得します

書式 int ACat_Mbg_GetSpotReqCmd(DWORD SlaveID, DWORD CommandID, WORD* TargetID, WORD* FunctionCode, WORD* StartAddress, WORD* ReadSize, WORD* Offset)

引数

SlaveID	: スレーブ ID
CommandID	: コマンド ID (1-32)
TargetID	: ターゲット機器 ID (1-8)
FunctionCode	: ファンクションコード
StartAddress	: 開始アドレス (0-65535)
ReadSize	: 書込みレジスタ数 (1-128)
Offset	: PDO オフセット (0-255)

ファンクションコード

FunctionCode	内容
MBG_CMD_FUNC_CODE_FORCE_SINGLECOIL	0x05: コイル
MBG_CMD_FUNC_CODE_PRESET_SINGLEREG	0x06: 保持レジスタ
MBG_CMD_FUNC_CODE_FORCE_MULTICOIL	0x0F: 複数コイル
MBG_CMD_FUNC_CODE_FORCE_MULTIREG	0x10: 複数保持レジスタ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットから即時要求データコマンドの取得を行います。

ACat_Mbg_SetSpotReqCmd 関数

機能 Modbus ゲートウェイユニットへ即時要求データコマンドの設定を行います

書式 int ACat_Mbg_SetSpotReqCmd(DWORD SlaveID, DWORD CommandID, WORD TargetID,
WORD FunctionCode, WORD StartAddress, WORD ReadSize,
WORD Offset)

引数

SlaveID	: スレーブ ID
CommandID	: コマンド ID (1-32)
TargetID	: ターゲット機器 ID (1-8)
FunctionCode	: ファンクションコード
StartAddress	: 開始アドレス (0-65535)
ReadSize	: 書込みレジスタ数 (1-128)
Offset	: PDO オフセット (0-255)

ファンクションコード

FunctionCode	内容
MBG_CMD_FUNC_CODE_FORCE_SINGLECOIL	0x05: コイル
MBG_CMD_FUNC_CODE_PRESET_SINGLEREG	0x06: 保持レジスタ
MBG_CMD_FUNC_CODE_FORCE_MULTICOIL	0x0F: 複数コイル
MBG_CMD_FUNC_CODE_FORCE_MULTIREG	0x10: 複数保持レジスタ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットへ即時要求データコマンドの設定を行います。

ACat_Mbg_GetManReqCmd 関数

機能 Modbus ゲートウェイユニットから手動要求データコマンドを取得します

書式 int ACat_Mbg_GetManReqCmd(DWORD SlaveID, DWORD CommandID, WORD* TargetID, WORD* FunctionCode, WORD* StartAddress, WORD* ReadSize, WORD* Offset, WORD* WriteSw)

引数

SlaveID	: スレーブ ID
CommandID	: コマンド ID (1-8)
TargetID	: ターゲット機器 ID (1-8)
FunctionCode	: ファンクションコード
StartAddress	: 開始アドレス (0-65535)
ReadSize	: 書込みレジスタ数 (1-128)
Offset	: PDO オフセット (0-255)
WriteSw	: 手動書込みフラグ

ファンクションコード

FunctionCode	内容
MBG_CMD_FUNC_CODE_FORCE_SINGLECOIL	0x05: コイル
MBG_CMD_FUNC_CODE_PRESET_SINGLEREG	0x06: 保持レジスタ
MBG_CMD_FUNC_CODE_FORCE_MULTICOIL	0x0F: 複数コイル
MBG_CMD_FUNC_CODE_FORCE_MULTIREG	0x10: 複数保持レジスタ

手動書込みフラグ

WriteSw	内容
MBG_REQCMD_WRITEFLG_OFF	0x00: OFF
MBG_REQCMD_WRITEFLG_ON	0x01: ON

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットから手動要求データコマンドの取得を行います。

ACat_Mbg_SetManReqCmd 関数

機能 Modbus ゲートウェイユニットへ手動要求データコマンドの設定を行います

書式 int ACat_Mbg_SetManReqCmd(DWORD SlaveID, DWORD CommandID, WORD TargetID, WORD FunctionCode, WORD StartAddress, WORD ReadSize, WORD Offset, WORD WriteSw)

引数

SlaveID	: スレーブ ID
CommandID	: コマンド ID (1-8)
TargetID	: ターゲット機器 ID (1-8)
FunctionCode	: ファンクションコード
StartAddress	: 開始アドレス (0-65535)
ReadSize	: 書込みレジスタ数 (1-128)
Offset	: PDO オフセット (0-255)
WriteSw	: 手動書込みフラグ

ファンクションコード

FunctionCode	内容
MBG_CMD_FUNC_CODE_FORCE_SINGLECOIL	0x05: コイル
MBG_CMD_FUNC_CODE_PRESET_SINGLEREG	0x06: 保持レジスタ
MBG_CMD_FUNC_CODE_FORCE_MULTICOIL	0x0F: 複数コイル
MBG_CMD_FUNC_CODE_FORCE_MULTIREG	0x10: 複数保持レジスタ

手動書込みフラグ

WriteSw	内容
MBG_REQCMD_WRITEFLG_OFF	0x00: OFF
MBG_REQCMD_WRITEFLG_ON	0x01: ON

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットへ手動要求データコマンドの設定を行います。

ACat_Mbg_GetMonCmdErr 関数

機能 Modbus ゲートウェイユニットのモニタデータコマンドのエラーステータスを取得します

書式 int ACat_Mbg_GetMonCmdErr (DWORD SlaveID, DWORD CommandID, WORD* Status, WORD* CommError, WORD* ModbusError)

引数

SlaveID : スレーブ ID
 CommandID : コマンド ID (1-32)
 Status : ステータス
 CommError : シリアル通信エラー
 ModbusError : Modbus 例外レスポンス

ステータス

Status	内容
MBG_CMD_STATUS_OK	0: 正常
MBG_CMD_STATUS_MBSRES	3: Modbus 例外レスポンス
MBG_CMD_STATUS_TIMEOUT	4: 応答タイムアウト
MBG_CMD_STATUS_ERRCHECK	5: エラーチェックエラー (CRC、LRC エラー)
MBG_CMD_STATUS_FORMAT	6: 応答電文フォーマットエラー

シリアル通信エラー

CommError	内容
MBG_CMD_COMMERR_NON	0: なし
MBG_CMD_COMMERR_OVERFLOW	1: オーバーフロー
MBG_CMD_COMMERR_PARITY	2: パリティエラー
MBG_CMD_COMMERR_FLAMING	3: フレーミングエラー
MBG_CMD_COMMERR_OVERRUN	4: オーバーランエラー

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットからモニタデータコマンドのエラーステータスの取得を行います。
 「Modbus 例外レスポンス」は、接続するターゲット機器から異常時に返される「例外コード」になります。詳細はターゲット機器のマニュアルを参照してください。

ACat_Mbg_GetSpotReqCmdErr 関数

機能 Modbus ゲートウェイユニットの即時要求データコマンドのエラーステータスを取得します

書式 int ACat_Mbg_GetSpotReqCmdErr (DWORD SlaveID, DWORD CommandID, WORD* Status, WORD* CommError, WORD* ModbusError)

引数

SlaveID : スレーブ ID
 CommandID : コマンド ID (1-32)
 Status : ステータス
 CommError : シリアル通信エラー
 ModbusError : Modbus 例外レスポンス

ステータス

Status	内容
MBG_CMD_STATUS_OK	0: 正常
MBG_CMD_STATUS_MBSRES	3: Modbus 例外レスポンス
MBG_CMD_STATUS_TIMEOUT	4: 応答タイムアウト
MBG_CMD_STATUS_ERRCHECK	5: エラーチェックエラー (CRC、LRC エラー)
MBG_CMD_STATUS_FORMAT	6: 応答電文フォーマットエラー

シリアル通信エラー

CommError	内容
MBG_CMD_COMMERR_NON	0: なし
MBG_CMD_COMMERR_OVERFLOW	1: オーバーフロー
MBG_CMD_COMMERR_PARITY	2: パリティエラー
MBG_CMD_COMMERR_FLAMING	3: フレーミングエラー
MBG_CMD_COMMERR_OVERRUN	4: オーバーランエラー

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットから即時要求データコマンドのエラーステータスの取得を行います。
 「Modbus 例外レスポンス」は、接続するターゲット機器から異常時に返される「例外コード」になります。詳細はターゲット機器のマニュアルを参照してください。

ACat_Mbg_GetManReqCmdErr 関数

機能 Modbus ゲートウェイユニットの手動要求データコマンドのエラーステータスを取得します

書式 int ACat_Mbg_GetManReqCmdErr (DWORD SlaveID, DWORD CommandID, WORD* Status, WORD* CommError, WORD* ModbusError)

引数

SlaveID : スレーブ ID
 CommandID : コマンド ID (1-8)
 Status : ステータス
 CommError : シリアル通信エラー
 ModbusError : Modbus 例外レスポンス

ステータス

Status	内容
MBG_CMD_STATUS_OK	0: 正常
MBG_CMD_STATUS_MBSRES	3: Modbus 例外レスポンス
MBG_CMD_STATUS_TIMEOUT	4: 応答タイムアウト
MBG_CMD_STATUS_ERRCHECK	5: エラーチェックエラー (CRC、LRC エラー)
MBG_CMD_STATUS_FORMAT	6: 応答電文フォーマットエラー

シリアル通信エラー

CommError	内容
MBG_CMD_COMMERR_NON	0: なし
MBG_CMD_COMMERR_OVERFLOW	1: オーバーフロー
MBG_CMD_COMMERR_PARITY	2: パリティエラー
MBG_CMD_COMMERR_FLAMING	3: フレーミングエラー
MBG_CMD_COMMERR_OVERRUN	4: オーバーランエラー

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットから手動要求データコマンドのエラーステータスの取得を行います。
 「Modbus 例外レスポンス」は、接続するターゲット機器から異常時に返される「例外コード」になります。詳細はターゲット機器のマニュアルを参照してください。

ACat_Mbg_GetSpotReqCmdRes 関数

機能 Modbus ゲートウェイユニットの即時要求データコマンドのレスポンスを取得します

書式 int ACat_Mbg_GetSpotReqCmdRes(DWORD SlaveID, DWORD CommandID, WORD* Response, WORD* CommError, WORD* ModbusError)

引数

SlaveID : スレーブ ID
 CommandID : コマンド ID (1-32)
 Response : レスポンス
 CommError : シリアル通信エラー
 ModbusError : Modbus 例外レスポンス

レスポンス

Response	内容
MBG_CMD_STATUS_OK	0: 処理中/待機
MBG_CMD_STATUS_COMPLETE	1: 正常完了
MBG_CMD_STATUS_MBSRES	3: Modbus 例外レスポンス
MBG_CMD_STATUS_TIMEOUT	4: 応答タイムアウト
MBG_CMD_STATUS_ERRCHECK	5: エラーチェックエラー (CRC、LRC エラー)
MBG_CMD_STATUS_FORMAT	6: 応答電文フォーマットエラー

シリアル通信エラー

CommError	内容
MBG_CMD_COMMERR_NON	0: なし
MBG_CMD_COMMERR_OVERFLOW	1: オーバーフロー
MBG_CMD_COMMERR_PARITY	2: パリティエラー
MBG_CMD_COMMERR_FLAMING	3: フレーミングエラー
MBG_CMD_COMMERR_OVERRUN	4: オーバーランエラー

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットから即時要求データコマンドのレスポンスの取得を行います。

「Modbus 例外レスポンス」は、接続するターゲット機器から異常時に返される「例外コード」になります。詳細はターゲット機器のマニュアルを参照してください。

ACat_Mbg_GetManReqCmdRes 関数

- 機能** Modbus ゲートウェイユニットの手動要求データコマンドのレスポンスを取得します
- 書式** int ACat_Mbg_GetManReqCmdRes (DWORD SlaveID, DWORD CommandID, WORD* Response, WORD* CommError, WORD* ModbusError)
- 引数**
- SlaveID : スレーブ ID
 - CommandID : コマンド ID (1-8)
 - Response : レスポンス
 - CommError : シリアル通信エラー
 - ModbusError : Modbus 例外レスポンス

レスポンス

Response	内容
MBG_CMD_STATUS_OK	0: 処理中/待機
MBG_CMD_STATUS_COMPLETE	1: 正常完了
MBG_CMD_STATUS_MBSRES	3: Modbus 例外レスポンス
MBG_CMD_STATUS_TIMEOUT	4: 応答タイムアウト
MBG_CMD_STATUS_ERRCHECK	5: エラーチェックエラー (CRC、LRC エラー)
MBG_CMD_STATUS_FORMAT	6: 応答電文フォーマットエラー

シリアル通信エラー

CommError	内容
MBG_CMD_COMMERR_NON	0: なし
MBG_CMD_COMMERR_OVERFLOW	1: オーバーフロー
MBG_CMD_COMMERR_PARITY	2: パリティエラー
MBG_CMD_COMMERR_FLAMING	3: フレーミングエラー
MBG_CMD_COMMERR_OVERRUN	4: オーバーランエラー

- 戻り値**
- ACAT_ER_OK : 正常
 - ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

- 説明**
- スレーブ ID で指定した Modbus ゲートウェイユニットから手動要求データコマンドのレスポンスの取得を行います。
- 「Modbus 例外レスポンス」は、接続するターゲット機器から異常時に返される「例外コード」になります。詳細はターゲット機器のマニュアルを参照してください。

ACat_Mbg_ClrMonCmdErr 関数

機能

Modbus ゲートウェイユニットのモニタデータコマンドのエラーステータスをクリアします

書式

```
int ACat_Mbg_ClrMonCmdErr (DWORD SlaveID, DWORD CommandID)
```

引数

SlaveID : スレーブ ID
CommandID : コマンド ID (1-32)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットからモニタデータコマンドのエラーステータスのクリアを行います。

ACat_Mbg_ClrSpotReqCmdErr 関数

機能 Modbus ゲートウェイユニットの即時要求データコマンドのエラーステータスをクリアします

書式 int ACat_Mbg_ClrSpotReqCmdErr (DWORD SlaveID, DWORD CommandID)

引数
SlaveID : スレーブ ID
CommandID : コマンド ID (1-32)

戻り値
ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットから即時要求データコマンドのエラーステータスのクリアを行います。

ACat_Mbg_ClrManReqCmdErr 関数

- 機能** Modbus ゲートウェイユニットの手動要求データコマンドのエラーステータスをクリアします
- 書式** int ACat_Mbg_ClrManReqCmdErr (DWORD SlaveID, DWORD CommandID)
- 引数**
SlaveID : スレーブ ID
CommandID : コマンド ID (1-8)
- 戻り値**
ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
- 説明** スレーブ ID で指定した Modbus ゲートウェイユニットから手動要求データコマンドのエラーステータスのクリアを行います。

ACat_Mbg_ClrSpotReqCmdRes 関数

機能

Modbus ゲートウェイユニットの即時要求データコマンドのレスポンスをクリアします

書式

```
int ACat_Mbg_ClrSpotReqCmdRes(DWORD SlaveID, DWORD CommandID)
```

引数

SlaveID : スレーブ ID
CommandID : コマンド ID (1-32)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットから即時要求データコマンドのレスポンスのクリアを行います。

ACat_Mbg_ClrManReqCmdRes 関数

機能

Modbus ゲートウェイユニットの手動要求データコマンドのレスポンスをクリアします

書式

```
int ACat_Mbg_ClrManReqCmdRes (DWORD SlaveID, DWORD CommandID)
```

引数

SlaveID : スレーブ ID
CommandID : コマンド ID (1-8)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した Modbus ゲートウェイユニットから手動要求データコマンドのレスポンスのクリアを行います。

ACat_Mbg_GetErrorStatus 関数

機能 Modbus ゲートウェイユニットのエラー状況を取得します

書式 int ACat_Mbg_GetErrorStatus (DWORD SlaveID, TMODBUS_ERROR_STATUS* pErrorStatus)

引数
 SlaveID : スレーブ ID
 pErrorStatus : エラー状況

エラー状況

```
typedef struct {
    DWORD MonCmdErr1;
    DWORD MonCmdErr2;
    DWORD MonCmdErr3;
    DWORD MonCmdErr4;
    DWORD SpotCmdErr1;
    DWORD SpotCmdErr2;
    DWORD SpotCmdErr3;
    DWORD SpotCmdErr4;
    DWORD ManCmdErr;
} TMODBUS_ERROR_STATUS, *PMODBUS_ERROR_STATUS;
```

MonCmdErr1 : モニタデータコマンド エラー状況
 MonCmdErr2 : 予備 (未使用)
 MonCmdErr3 : 予備 (未使用)
 MonCmdErr4 : 予備 (未使用)
 SpotReqCmdErr1 : 即時要求データコマンド エラー状況
 SpotReqCmdErr2 : 予備 (未使用)
 SpotReqCmdErr3 : 予備 (未使用)
 SpotReqCmdErr4 : 予備 (未使用)
 ManCmdErr : 手動要求データコマンド エラー状況

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットからエラー状況の取得を行います。

ACat_Mbg_GetResponseStatus 関数

機能 Modbus ゲートウェイユニットのレスポンス状況を取得します

書式 int ACat_Mbg_GetResponseStatus (DWORD SlaveID, TMODBUS_RESPONSE_STATUS* pResStatus)

引数
 SlaveID : スレーブ ID
 pErrorStatus : エラー状況

エラー状況

```
typedef struct {
    DWORD MonCmdRes1;
    DWORD MonCmdRes2;
    DWORD MonCmdRes3;
    DWORD MonCmdRes4;
    DWORD SpotCmdRes1;
    DWORD SpotCmdRes2;
    DWORD SpotCmdRes3;
    DWORD SpotCmdRes4;
    DWORD ManCmdRes;
} TMODBUS_RESPONSE_STATUS, *PMODBUS_RESPONSE_STATUS;
```

MonCmdRes1 : 予備 (未使用)
 MonCmdRes2 : 予備 (未使用)
 MonCmdRes3 : 予備 (未使用)
 MonCmdRes4 : 予備 (未使用)
 SpotReqCmdRes1 : 即時要求データコマンド レスポンス状況
 SpotReqCmdRes2 : 予備 (未使用)
 SpotReqCmdRes3 : 予備 (未使用)
 SpotReqCmdRes4 : 予備 (未使用)
 ManCmdRes : 手動要求データコマンド レスポンス状況

戻り値
 ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットからレスポンス状況の取得を行います。

ACat_Mbg_SaveParam 関数

機能 Modbus ゲートウェイユニットにパラメータを保存します

書式 int ACat_Mbg_SaveParam(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 Modbus ゲートウェイユニットにパラメータを保存します。
保存されるパラメータは、ACat_Mbg_SetCommInfo 関数、ACat_Mbg_SetTargetInfo 関数、
ACat_Mbg_SetMonCmd 関数、ACat_Mbg_SetSpotReqCmd 関数、ACat_Mbg_SetManReqCmd 関数にて
設定されるパラメータのみになります。
入力データ、出力データは保存されません。

ACat_Mbg_LoadParam 関数

機能 Modbus ゲートウェイユニットのパラメータを初期化します

書式 int ACat_Mbg_LoadParam(DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 Modbus ゲートウェイユニットのパラメータを初期化します。
初期化されるパラメータは、ACat_Mbg_SetCommInfo 関数、ACat_Mbg_SetTargetInfo 関数、
ACat_Mbg_SetMonCmd 関数、ACat_Mbg_SetSpotReqCmd 関数、ACat_Mbg_SetManReqCmd 関数にて
設定されるパラメータのみになります。
入力データ、出力データは保存されません。

ACat_Mbg_SoftReset 関数

機能 Modbus ゲートウェイユニットのソフトウェアリセットを行います

書式 int ACat_Mbg_SoftReset (DWORD SlaveID)

引数 SlaveID : スレーブ ID

戻り値 ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した Modbus ゲートウェイユニットのソフトウェアリセットを行います。

2-14 CUnet ゲートウェイユニット関数

2-14-1 Utility

ACat_CN_Create 関数

機能	CUnet ゲートウェイユニット関数群を初期化します	
書式	int ACat_CN_Create(void)	
引数	なし	
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある CUnet ゲートウェイユニット関数の初期化を行います。 この関数コールは EtherCAT 通信開始後しか使用できません。 CUnet ゲートウェイユニットを使用する際には、必ずコールする必要があります。	

ACat_CN_Destroy 関数

機能	CUnet ゲートウェイユニット関数群を終了します	
書式	int ACat_CN_Destroy(void)	
引数	なし	
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_OK 以外	: エラーコード(エラーコード一覧を参照)
説明	ライブラリ内にある CUnet ゲートウェイユニット関数の終了処理を行います。 この関数コール後は、CUnet ゲートウェイユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。	

2-14-2 Open/Close

ACat_CN_open 関数

機能	CUnet ゲートウェイユニットをオープンします
書式	int ACat_CN_open(DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CUnet ゲートウェイユニットをオープンします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後、指定した CUnet ゲートウェイユニットにアクセス可能となります。 CUnet ゲートウェイユニットを使用する際には、必ずコールする必要があります。

ACat_CN_close 関数

機能	CUnet ゲートウェイユニットをクローズします
書式	int ACat_CN_close (DWORD SlaveID)
引数	SlaveID : スレーブ ID
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CUnet ゲートウェイユニットをクローズします。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 この関数コール後は、指定した CUnet ゲートウェイユニットにアクセス不可になります。 アプリケーションの終了時に必ずコールする必要があります。

ACat_CN_start 関数

機能

CUnnet ゲートウェイユニットの CUnet 通信を開始します

書式

```
int ACat_CN_start(DWORD SlaveID, DWORD ch, int StationAdr,  
                 int StationSize, int ByteBusSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
StationAdr : ステーションアドレス (0-63)
StationSize : ステーションサイズ (1-64)
ByteBusSize : 予約 (無視されます)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの CUnet 通信を開始します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
StationAdr で指定されたステーションアドレスから、StationSize で指定されたサイズ分
自己のグローバルメモリを占有します。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。
複数のアプリケーションでこの関数をコールした場合、現状での CUnet 通信が開始、停止
いずれの状態でも再度関数で指定されたパラメータで通信を開始します。
他のアプリケーションですでに通信開始されている場合は注意が必要です。

ACat_CN_start_lf 関数

機能

CUnet ゲートウェイユニットの CUnet 通信を LF モードで開始します

書式

```
int ACat_CN_start_lf(DWORD SlaveID, DWORD ch, int StationAdr,  
                    int StationSize, int ByteBusSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
StationAdr : ステーションアドレス (0-63)
StationSize : ステーションサイズ (1-64)
ByteBusSize : 予約 (無視されます)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの CUnet 通信を LF モードで開始します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

LF モードは、フレーム長定数 (LOF) が「256」となるモードです。LF モードによって、CUnet のネットワークへ HUB (通信ケーブル分岐ユニット) を挿入可能となります。ネットワークに HUB (通信ケーブル分岐ユニット) を挿入した CUnet は、以下のように通信ケーブルの敷設の自由度が高まり、適用可能なユーザシステムを拡張することができます。

- ① ネットワークの通信ケーブルの総延長を伸ばすことが可能。
- ② ネットワークの通信ケーブルを分岐することが可能。
- ③ 各 CUnet ステーション装置の終端抵抗への配慮を削減できる。
- ④ スター接続が可能となる。
- ⑤ 光ファイバへの対応が容易となる。

複数のアプリケーションでこの関数をコールした場合、現状態での CUnet 通信が開始、停止いずれの状態でも再度関数で指定されたパラメータで通信を開始します。他のアプリケーションですでに通信開始されている場合は注意が必要です。

ACat_CN_start_gmm 関数

機能

CUnet ゲートウェイユニットの CUnet 通信を GMM モードで開始します

書式

```
int ACat_CN_start_gmm(DWORD SlaveID, DWORD ch, int ByteBusSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
ByteBusSize : 予約 (無視されます)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード (エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの CUnet 通信を GMM モードで開始します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

GMM (Global Memory Monitor) モードは、他の CUnet ステーションと一切リンクせずに、継続的タイムシェアリングに基づいて他の CUnet ステーションが送信するパケットを受信するだけの、グローバルメモリデータモニタ (覗き見) モードです。他の CUnet ステーション同士が共有しているグローバルメモリのデータをモニタリングすることが可能です。

ACat_CN_stop 関数

機能	CUnet ゲートウェイユニットの CUnet 通信を停止します
書式	int ACat_CN_stop(DWORD SlaveID, DWORD ch)
引数	SlaveID : スレーブ ID ch : チャンネル番号 (1-2)
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CUnet ゲートウェイユニットの CUnet 通信を停止します。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_reset 関数

機能 CUnet ゲートウェイユニットの自己の占有エリアのグローバルメモリをクリアします

書式 int ACat_CN_reset (DWORD SlaveID, DWORD ch, WORD Data)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-2)
Data	: データ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した CUnet ゲートウェイユニットに Data で指定された値で、自己の占有エリアのグローバルメモリをクリアします。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_GetCnectStat 関数

機能

CUnet ゲートウェイユニットの接続ステータスを取得します

書式

```
int ACat_CN_GetCnectStat (DWORD SlaveID, DWORD ch, int * StationAdr, int * StationSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
StationAdr : ステーションアドレス
StationSize : ステーションサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの自己ステーションアドレスとステーションサイズを取得します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_config 関数

機能

CUnet ゲートウェイユニットの通信ボーレートを設定します

書式

ACat_CN_config(DWORD SlaveID, DWORD ch, int Baudrate)

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
Baudrate : 通信ボーレート (3:3Mbps, 6:6Mbps, 12:12Mbps)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの通信ボーレートを設定します。
設定した通信ボーレートは設定後の通信開始時に有効になります。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_Status 関数

機能 CUnet ゲートウェイユニットの動作ステータスを取得します

書式 int ACat_CN_Status(DWORD SlaveID, DWORD ch, PCN_STATUS pCnStatus)

引数

SlaveID	: スレーブ ID
ch	: チャンネル番号 (1-2)
pCnStatus	: ステータス

通信状態データ

```
typedef struct _CN_STATUS {  
    int status;  
    int sa;  
    int slen;  
} CN_STATUS, *PCN_STATUS;
```

status : ステータス [0:停止中, 1:運転中]
sa : グローバルメモリアドレス
slen : グローバルメモリサイズ

戻り値

ACAT_ER_OK	: 正常
ACAT_ER_OK 以外	: エラーコード (エラーコード一覧を参照)

説明 スレーブ ID で指定した CUnet ゲートウェイユニットの動作ステータスを取得します。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_Put 関数

機能

CUnet ゲートウェイユニットの占有ステーションアドレスに出力します

書式

```
int ACat_CN_Put(DWORD SlaveID, DWORD ch, unsigned long long * pData)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pData : データを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの自己占有ステーションアドレスにデータを出力します。
出力するデータは8バイトサイズです。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_Get 関数

機能

CUnet ゲートウェイユニットの指定したステーションアドレスを取得します

書式

```
int ACat_CN_Get(DWORD SlaveID, DWORD ch, short StationAdr, unsigned long long *pData)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
StationAdr : ステーションアドレス (1-63)
pData : データを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの指定したステーションアドレスからデータを取得します。
取得するデータは 8 バイトサイズです。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

2-14-3 メール

ACat_CN_MailOpen 関数

機能	CUnet ゲートウェイユニットのメールをオープンします	
書式	int ACat_CN_MailOpen(DWORD SlaveID, DWORD ch)	
引数	SlaveID	: スレーブ ID
	ch	: チャンネル番号 (1-2)
戻り値	ACAT_ER_OK	: 正常
	ACAT_ER_CN_ALREADYMAILOPEN	: メールが既にオープンしています
	上記以外	: エラーコード(エラーコード一覧を参照)

説明 スレーブ ID で指定した CUnet ゲートウェイユニットのメールをオープンします。
この関数コール後、その他のメール機能が使用可能となります。
メール機能を使用の際には、必ずコールする必要があります。
メール機能は複数アプリケーションでの使用はできません。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_MailClose 関数

機能 CUnet ゲートウェイユニットのメールをクローズします

書式 int ACat_CN_MailClose(DWORD SlaveID, DWORD ch)

引数
SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)

戻り値
ACAT_ER_OK : 正常
ACAT_ER_CN_MAILNOTOPEN : メールがオープンされていない
上記以外 : エラーコード(エラーコード一覧を参照)

説明
スレーブ ID で指定した CUnet ゲートウェイユニットのメールをクローズします。
この関数コール後、その他メール機能が使用不可になります。
アプリケーションの終了時に必ずコールする必要があります。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_MailSend 関数

機能

CUnet ゲートウェイユニットのメールを送信します
(256 バイト以下のデータを送信します)

書式

int ACat_CN_MailSend(DWORD SlaveID, DWORD ch, int StationAdr, void *pMail, int Len)

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
StationAdr : 送信先のステーションアドレス (0-63)
pMail : 送信するデータのポインタ
Len : 送信するデータのバイトサイズ (1-256)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_CN_MAILSIZE : メールサイズのエラーです
ACAT_ER_CN_MAILSENDING : メール送信中の為送信できません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットから指定したステーションアドレスに指定したサイズのデータを送信します。
Len に 256 を超える値を指定した場合、関数はメール送信せずに [ACAT_ER_CN_MAILSIZE] を返します。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_MailReceive 関数

機能

CUnet ゲートウェイユニットのメールの受信データを取得します

書式

```
int ACat_CN_MailReceive(DWORD SlaveID, DWORD ch, void *pMail, int Len,
                        int *pStationAdr)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pMail : 受信したデータを格納するポインタ
Len : 受信したデータを取得するバイトサイズ (1-256)
pStationAdr : 送信元のステーションアドレスを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_CN_MAILSIZE : メールサイズのエラーです
ACMST_ER_CN_MAILNOTRECEIVE : メールを受信していません
上記以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットから指定したサイズのデータを受信します。Len に 256 を超える値を指定した場合、関数はメール受信せずに [ACAT_ER_CN_MAILSIZE] を返します。

メール着信のシグナルは [ACat_CN_MailStatus] 関数より取得できます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。

チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_MailStatus 関数

機能

CUnet ゲートウェイユニットのメールのステータスを取得します

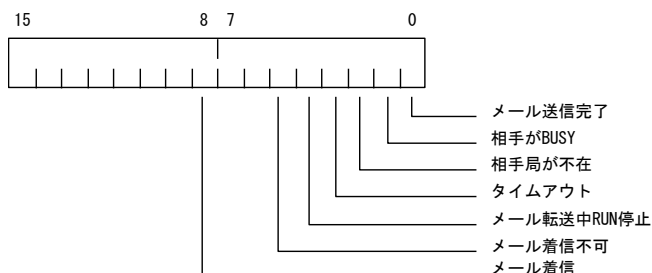
書式

int ACat_CN_MailStatus(DWORD SlaveID, DWORD ch, short *Status)

引数

SlaveID : スレーブ ID
 ch : チャンネル番号 (1-2)
 Status : ステータスを格納するポインタ

Status のフォーマット



ステータス

Status	値	内容
MKY43_MAIL_STATUS_SENDEND	0x0001	メール送信完了
MKY43_MAIL_STATUS_BUSY	0x0002	相手が BUSY
MKY43_MAIL_STATUS_NOEXIST	0x0004	相手局が不在
MKY43_MAIL_STATUS_TIMEOUT	0x0008	タイムアウト
MKY43_MAIL_STATUS_TRANSSTOP	0x0010	メール転送中 RUN 停止
MKY43_MAIL_STATUS_NOTRECEIVE	0x0020	メール着信不可
MKY43_MAIL_STATUS_RECEIVE	0x0100	メール着信

戻り値

ACAT_ER_OK : 正常
 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットからメールのステータスを取得します。メール送信をすると、送信関連のステータスはクリアされます。メール送信完了 [MKY43_MAIL_STATUS_SENDEND] は送信完了すると 1 が読み取れます。エラーによる送信中断の場合、値は不定です。メールを着信すると、[MKY43_MAIL_STATUS_RECEIVE] は 1 が読み取れます。スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。メール着信不可発生時は、CONET ラインが一度切断された可能性がある為、メールをクローズし、再度メールをオープンする必要があります。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

2-14-4 レジスタアクセス

ACat_CN_GetReg64 関数

機能	64 ビットアクセスで指定したレジスタを読みみます
書式	<pre>int ACat_CN_GetReg64(DWORD SlaveID, DWORD ch, unsigned short OffsetAdr, unsigned long long * pReg)</pre>
引数	SlaveID : スレーブ ID ch : チャンネル番号 (1-2) OffsetAdr : レジスタのアドレス pReg : レジスタデータを格納するポインタ
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CUnet ゲートウェイユニットの OffsetAdr で指定したレジスタを 64 ビットアドレスで読みみます。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_GetReg32 関数

機能

32 ビットアクセスで指定したレジスタを読みみます

書式

```
int ACat_CN_GetReg32(DWORD SlaveID, DWORD ch, unsigned short OffsetAdr,  
                    unsigned long * pReg)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
OffsetAdr : レジスタのアドレス
pReg : レジスタデータを格納するポインタ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの OffsetAdr で指定したレジスタを 32 ビットアドレスで読みみます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_GetReg16 関数

機能	16 ビットアクセスで指定したレジスタを読み込みます
書式	<pre>int ACat_CN_GetReg16(DWORD SlaveID, DWORD ch, unsigned short OffsetAdr, unsigned short * pReg)</pre>
引数	SlaveID : スレーブ ID ch : チャンネル番号 (1-2) OffsetAdr : レジスタのアドレス pReg : レジスタデータを格納するポインタ
戻り値	ACAT_ER_OK : 正常 ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
説明	スレーブ ID で指定した CUnet ゲートウェイユニットの OffsetAdr で指定したレジスタを 16 ビットアドレスで読み込みます。 スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。 チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetReg64 関数

機能

64 ビットアクセスで指定したレジスタを書込みます

書式

```
int ACat_CN_SetReg64(DWORD SlaveID, DWORD ch, unsigned short OffsetAdr,  
                    unsigned long long Data)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
OffsetAdr : レジスタのアドレス
Data : 書込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの OffsetAdr で指定したレジスタに 64 ビットアドレスで書込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetReg32 関数

機能

32 ビットアクセスで指定したレジスタを書込みます

書式

```
int ACat_CN_SetReg32(DWORD SlaveID, DWORD ch, unsigned short OffsetAdr,  
                    unsigned long Data)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
OffsetAdr : レジスタのアドレス
Data : 書込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの OffsetAdr で指定したレジスタに 32 ビットアドレスで書込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetReg16 関数

機能

16 ビットアクセスで指定したレジスタを書込みます

書式

```
int ACat_CN_SetReg16(DWORD SlaveID, DWORD ch, unsigned short OffsetAdr,  
                    unsigned short Data)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
OffsetAdr : レジスタのアドレス
Data : 書込みデータ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの OffsetAdr で指定したレジスタに 16 ビットアドレスで書込みます。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

2-14-5 レジスタアクセス

ACat_CN_GetMemByte 関数

機能

Byte 単位でグローバルメモリからデータを読みみます

書式

```
int ACat_CN_GetMemByte(DWORD SlaveID, DWORD ch, unsigned char *pBuf, short StationAdr,
short StationOffsetAdr, unsigned short ByteSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 8 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
ByteSize : データを取得するバイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、Byte 単位で指定したサイズのデータを読みみます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_GetMemShort 関数

機能

Word 単位でグローバルメモリからデータを読みみます

書式

```
int ACat_CN_GetMemShort(DWORD SlaveID, DWORD ch, unsigned short * pBuf,  
                        short StationAdr, short StationOffsetAdr, unsigned short ShortSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 16 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
ShortSize : データを取得する 2 バイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、Word 単位で指定したサイズのデータを読みみます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_GetMemLong 関数

機能

Long 単位でグローバルメモリからデータを読み込みます

書式

```
int ACat_CN_GetMemLong(DWORD SlaveID, DWORD ch, unsigned long * pBuf, short
                        StationAdr, short StationOffsetAdr, unsigned short LongSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 32 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
LongSize : データを取得する 4 バイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、Long 単位で指定したサイズのデータを読み込みます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_GetMemDLong 関数

機能

DLong 単位でグローバルメモリからデータを読みみます

書式

```
int ACat_CN_GetMemDLong(DWORD SlaveID, DWORD ch, unsigned long long * pBuf,  
                        short StationAdr, short StationOffsetAdr, unsigned short DLongSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 64 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
DLongSize : データを取得する 8 バイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、DLong 単位で指定したサイズのデータを読みみます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetMemByte 関数

機能

Byte 単位でグローバルメモリにデータを書込みます

書式

```
int ACat_CN_SetMemByte(DWORD SlaveID, DWORD ch, unsigned char *pBuf, short StationAdr,
                       short StationOffsetAdr, unsigned short ByteSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 8 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
ByteSize : データをセットするバイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、Byte 単位で指定したサイズのデータを書込みます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetMemShort 関数

機能

Word 単位でグローバルメモリにデータを書込みます

書式

```
int ACat_CN_SetMemShort(DWORD SlaveID, DWORD ch, unsigned short * pBuf,  
                        short StationAdr, short StationOffsetAdr, unsigned short ShortSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 16 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
ShortSize : データをセットする 2 バイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、Word 単位で指定したサイズのデータを書込みます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetMemLong 関数

機能

Long 単位でグローバルメモリにデータを書込みます

書式

```
int ACat_CN_SetMemLong(DWORD SlaveID, DWORD ch, unsigned long *pBuf, short StationAdr,
                       short StationOffsetAdr, unsigned short LongSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 32 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
LongSize : データをセットする 4 バイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、Long 単位で指定したサイズのデータを書込みます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_SetMemDLong 関数

機能

DLong 単位でグローバルメモリにデータを書込みます

書式

```
int ACat_CN_SetMemDLong(DWORD SlaveID, DWORD ch, unsigned long long * pBuf,  
                        short StationAdr, short StationOffsetAdr, unsigned short DLongSize)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)
pBuf : 64 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
DLongSize : データをセットする 8 バイトサイズ

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットの StationAdr で指定したステーションアドレスのグローバルメモリの StationOffsetAdr で指定したオフセットから、DLong 単位で指定したサイズのデータを書込みます。

スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

ACat_CN_HardReset 関数

機能

CUnet ゲートウェイユニットのハードウェアリセットを行います

書式

```
int ACat_CN_HardReset (DWORD SlaveID, DWORD ch)
```

引数

SlaveID : スレーブ ID
ch : チャンネル番号 (1-2)

戻り値

ACAT_ER_OK : 正常
ACAT_ER_OK 以外 : エラーコード(エラーコード一覧を参照)

説明

スレーブ ID で指定した CUnet ゲートウェイユニットのハードウェアリセットを行います。
スレーブ ID はユニット本体の DipSW により設定したものを指定しなければなりません。
チャンネル番号はユニット上に CUnet 通信 IC が複数ある場合の番号です。

2-15 付録

2-15-1 通知コード

通知コード一覧

エラーコード	名称	内容
0x00000001	ACAT_NOTIFY_STATECHANGED	動作状態の変更を通知
0x00000002	ACAT_NOTIFY_ETH_LINK_CONNECTED	Ethernet ケーブルが接続時に通知
0x00000010	ACAT_NOTIFY_DC_STATUS	Distributed clocks (DC) の初期化を通知
0x00000011	ACAT_NOTIFY_DC_SLV_SYNC	DC の同期通信が開始、または逸脱を通知
0x00000012	ACAT_NOTIFY_DCM_SYNC	DCM 同期が発生時に通知
0x00000021	ACAT_NOTIFY_SLAVE_PRESENCE	非表示のスレーブが表示された
0x00000100	ACAT_NOTIFY_SB_STATUS	バススキャンが正常に終了
0x00000101	ACAT_NOTIFY_SB_MISMATCH	バススキャンの結果が不一致
0x00000102	ACAT_NOTIFY_SB_DUPLICATE_HC_NODE	重複した HC グループが製品コード、ベンダーID と等しいことが検出
0x00000200	ACAT_NOTIFY_MBOXRCV	メールボックスを受信
0x00000300	ACAT_NOTIFY_AUTO_RETURN	自動復帰機能が動作
0xFFFFFFFF	ACAT_NOTIFY_ILLEGAL_CODE	例外コードが発生しました

2-15-2 EtherCAT エラー通知コード

EtherCAT エラー通知コード一覧

エラーコード	名称	内容
0x80000001	ACAT_NOTIFY_ER_CYGCMD_WKC	サイクルコマンドエラー (WKC)
0x80000002	ACAT_NOTIFY_ER_MASTER_INITCMD_WKC	マスター初期化コマンドエラー (WKC)
0x80000003	ACAT_NOTIFY_ER_SLAVE_INITCMD_WKC	スレーブ初期化コマンドエラー (WKC)
0x80000008	ACAT_NOTIFY_ER_COE_MBXSNW_WKC	CoE メールボックス送信エラー (WKC)
0x80000009	ACAT_NOTIFY_ER_FOE_MBXSNW_WKC	FoE メールボックス送信エラー (WKC)
0x8000000A	ACAT_NOTIFY_ER_FRAME_RESPONSE	送信されたイーサネットフレームで応答がありません
0x8000000B	ACAT_NOTIFY_ER_SLAVE_INITCMD_RESPONSE	スレーブから送信された初期化コマンドで応答がありません
0x8000000C	ACAT_NOTIFY_ER_MASTER_INITCMD_RESPONSE	送信されたマスター初期化コマンドで応答がありません
0x8000000E	ACAT_NOTIFY_ER_MBSLAVE_INITCMD_TIMEOUT	メールボックスの初期化コマンドの応答を待機しているときのタイムアウト
0x8000000F	ACAT_NOTIFY_ER_NOT_ALL_DEVICES_OPERATIONAL	すべてのスレーブが OP 状態にありません
0x80000010	ACAT_NOTIFY_ER_ETH_LINK_NOT_CONNECTED	イーサネットケーブルが接続されていません
0x80000012	ACAT_NOTIFY_ER_RED_LINEBRK	Cable Redundancy は未サポート
0x80000013	ACAT_NOTIFY_ER_STATUS_SLAVE	フレーム受信時に少なくとも 1 つのスレーブがエラー状態になっています
0x80000014	ACAT_NOTIFY_ER_SLAVE_STATUS_INFO	スレーブのエラーステータス情報を通知
0x80000015	ACAT_NOTIFY_ER_SLAVE_NOT_ADDRESSABLE	スレーブのステーションアドレスが異常
0x80000016	ACAT_NOTIFY_ER_MBSLAVE_COE_SDO_ABORT	COE メールボックスの SDO アボートが発生
0x80000017	ACAT_NOTIFY_ER_CLIENTREGISTRATION_DROPPED	クライアント登録が削除されました
0x80000018	ACAT_NOTIFY_ER_RED_LINEFIXED	Cable Redundancy は未サポート
0x80000019	ACAT_NOTIFY_ER_FOE_MBSLAVE	FoE メールボックスのアボートが発生
0x8000001A	ACAT_NOTIFY_ER_PDIWATCHDOG	PDI Watchdog がスレーブで発生しました
0x8000001B	ACAT_NOTIFY_ER_SLAVE_NOTSUPPORTED	スレーブはサポートされていません
0x8000001C	ACAT_NOTIFY_ER_SLAVE_UNEXPECTED_STATE	スレーブが予期しない状態 (ESM) です
0x8000001D	ACAT_NOTIFY_ER_ALL_DEVICES_OPERATIONAL	全スレーブが Operational になった
0x8000001E	ACAT_NOTIFY_ER_EEPROM_CHECKSUM	EEPROM のチェックサム異常が発生
0x8000001F	ACAT_NOTIFY_ER_LINE_CROSSED	ライン交差が検出されました
0x80000020	ACAT_NOTIFY_ER_JUNCTION_RED_CHANGE	Cable Redundancy は未サポート
0x80000028	ACAT_NOTIFY_ER_FRAMELOSS_AFTER_SLAVE	スレーブ後がフレームロスしています
0x80000101	ACAT_NOTIFY_ER_HC_DETECTADDGROUPS	HotConnect は未サポート
0x80000102	ACAT_NOTIFY_ER_HC_PROBEALLGROUPS	HotConnect は未サポート
0x80000103	ACAT_NOTIFY_ER_HC_TOPOCHGDONE	HotConnect は未サポート

2-15-3 エラーコード

エラーコード一覧

エラーコード	名称	内容
0x0000000	ACAT_ER_OK	正常
0x9000001	ACAT_ER_ALREADYOPEN	既にオープンしている
0x9000002	ACAT_ER_NOTOPEN	オープンされていない
0x9000003	ACAT_ER_INVALIDPARAM	無効な引数
0x9000004	ACAT_ER_OPENDEVICE	デバイスドライバ起動失敗
0x9000005	ACAT_ER_CREATETHREAD	スレッド作成失敗
0x9000006	ACAT_ER_CREATESEMAPH	セマフォ作成失敗
0x9000007	ACAT_ER_CREATEMAP	マップファイル作成失敗
0x9000008	ACAT_ER_CREATEMAIL	メールスロット作成失敗
0x9000009	ACAT_ER_INICONFIG	設定ファイルの記述エラー
0x900000A	ACAT_ER_ALREADYSTART	スタートしている
0x900000B	ACAT_ER_NOTSTART	スタートしていない
0x900000C	ACAT_ER_DEVICE	デバイスアクセスエラー
0x900000D	ACAT_ER_LOADDEVICE	デバイスロードエラー
0x900000E	ACAT_ER_MAILCODE	メール受信コードエラー
0x900000F	ACAT_ER_NOTFOUND_SLAVE	スレーブ検出失敗
0x9000010	ACAT_ER_ETC_COMLINK	EC マスタ通信エラー
0x9000011	ACAT_ER_NOTSERVOON	サーボ ON ではない
0x9000012	ACMST_ER_MOVING_MODECHANGE	動作中のオペレーションモード設定変更
0x9000013	ACAT_ER_TIMEOUT	コマンドタイムアウト
0x9000014	ACMST_ER_NOTPDOADDR	PDO 割付にない
0x9000015	ACAT_ER_BUSY	SDO がビジー状態
0x9000016	ACAT_ER_STATE_FAULT	ステートが FAULT 状態
0x9000017	ACAT_ER_NOTCIA402	CiA402 デバイスではない
0x9000020	ACAT_ER_NOTLICENSE	マスタライセンスがない
0x9000030	ACAT_ER_CREATEENI	ENI ファイル作成失敗(ソケットの生成に失敗)
0x9000031	ACAT_ER_CREATEENI1	ENI ファイル作成失敗(サーバの接続に失敗)
0x9000032	ACAT_ER_CREATEENI2	ENI ファイル作成失敗(サーバとのデータの送受信に失敗)
0xFFFFFFFF	ACAT_ER_UNITYTYPE	ユニットタイプエラー

2-15-4 EtherCAT マスタエラーコード

EtherCAT マスタエラーコード(1/7)

エラーコード	名称	内容
0x90000101	ACAT_ER_MST_NOTSUPPORT	未サポートエラーが発生しました
0x90000102	ACAT_ER_MST_INVALIDINDEX	無効なインデックスを指定しました
0x90000103	ACAT_ER_MST_INVALIDOFFSET	無効なオフセットを指定しました
0x90000104	ACAT_ER_MST_CANCEL	キャンセルが発生しました
0x90000105	ACAT_ER_MST_INVALIDSIZE	無効なサイズを指定しました
0x90000106	ACAT_ER_MST_INVALIDDATA	無効なデータを指定しました
0x90000107	ACAT_ER_MST_NOTREADY	準備ができていない
0x90000108	ACAT_ER_MST_BUSY	現在ビジー状態であり、API リクエストを処理できません
0x90000109	ACAT_ER_MST_ACYC_FRM_FREEQ_EMPTY	EtherCAT 非同期コマンドを送信できません
0x9000010A	ACAT_ER_MST_NOMEMORY	メモリ不足が発生しました
0x9000010B	ACAT_ER_MST_INVALIDPARM	無効なパラメータを指定しました
0x9000010C	ACAT_ER_MST_NOTFOUND	コンフィグファイルが見つからないか、無効な SlaveID で呼び出されました。
0x9000010D	ACAT_ER_MST_DUPLICATE	重複した固定アドレスが検出されました
0x9000010E	ACAT_ER_MST_INVALIDSTATE	初期化されていないか、構成されていません
0x9000010F	ACAT_ER_MST_TIMER_LIST_FULL	タイマーリストにスレーブを追加できない
0x90000110	ACAT_ER_MST_TIMEOUT	タイムアウトが発生しました
0x90000111	ACAT_ER_MST_OPENFAILED	オープンに失敗しました
0x90000112	ACAT_ER_MST_SENDFAILED	フレームの送信に失敗しました
0x90000113	ACAT_ER_MST_INSERTMAILBOX	メールボックスコマンドを登録できませんでした
0x90000114	ACAT_ER_MST_INVALIDCMD	不明なメールボックスコマンドを検出しました
0x90000115	ACAT_ER_MST_UNKNOWN_MBX_PROTOCOL	未定義なメールボックスプロトコルを検出しました
0x90000116	ACAT_ER_MST_ACCESSDENIED	マスタ内部ソフトウェアが発生しました
0x90000117	ACAT_ER_MST_IDENTIFICATIONFAILED	識別コマンドが失敗しました
0x90000118	ACAT_ER_MST_LOCK_CREATE_FAILED	Lock の生成に失敗しました
0x9000011A	ACAT_ER_MST_PRODKEY_INVALID	プロダクトキーが無効です
0x9000011B	ACAT_ER_MST_WRONG_FORMAT	コンフィグファイルの形式が正しくありません
0x9000011C	ACAT_ER_MST_FEATURE_DISABLED	無効な機能を実行しようとして失敗しました
0x9000011D	ACAT_ER_MST_SHADOW_MEMORY	間違ったモードでメモリが要求されました
0x9000011E	ACAT_ER_MST_BUSCONFIG_MISMATCH	コンフィグファイルと現在接続されているバス topology が一致しません
0x9000011F	ACAT_ER_MST_CONFIGDATA_READ	コンフィグファイルを読むことができません
0x90000120	ACAT_ER_MST_ENI_NO_SAFEOP_OP_SUPPORT	構成は SAFEOP および OP 要求状態をサポートしていません
0x90000121	ACAT_ER_MST_XML_CYCCMDS_MISSING	コンフィグファイルに周期コマンドが含まれていません
0x90000122	ACAT_ER_MST_XML_ALSTATUS_READ_MISSING	コンフィグファイルの周期処理に ALSTATUS レジスタの読み取りがありません
0x90000123	ACAT_ER_MST_MCSM_FATAL_ERROR	マスタ制御状態は未定義状態です
0x90000124	ACAT_ER_MST_SLAVE_ERROR	スレーブエラーが検出されました 下記、通知エラーコードを参照ください ACAT_NOTIFY_ER_STATUS_SLAVE ACAT_NOTIFY_ER_SLAVE_STATUS_INFO

EtherCAT マスタエラーコード(2/7)

エラーコード	名称	内容
0x90000125	ACAT_ER_MST_FRAME_LOST	EtherCAT フレームがバスセグメントで失われたため、応答が受信されませんでした このエラーが頻繁に表示される場合、配線に問題がある可能性があります
0x90000126	ACAT_ER_MST_CMD_MISSING	不完全な EtherCAT フレームを受信しました
0x90000127	ACAT_ER_MST_CYCCMD_WKC_ERROR	サイクルコマンドエラー (WKC) が発生しました
0x90000128	ACAT_ER_MST_INVALID_DCL_MODE	DC ラッチが [Auto Read] モードで実行されている場合、この機能は使用できません
0x90000129	ACAT_ER_MST_AI_ADDRESS	コンフィグファイルとバストポロジが一致しなくなりました エラーは、すでに認識されているスレーブがもう存在しない場合にのみ表示されます
0x9000012A	ACAT_ER_MST_INVALID_SLAVE_STATE	メールボックスコマンドは、現在のスレーブ状態では許可されていません
0x9000012B	ACAT_ER_MST_SLAVE_NOT_ADDRESSABLE	ステーションアドレスが失われました スレーブはパワーサイクル状態にあります
0x9000012C	ACAT_ER_MST_CYC_CMDS_OVERFLOW	構成ユーティリティ内でコンフィグファイルを作成中にエラーが発生しました
0x9000012D	ACAT_ER_MST_LINK_DISCONNECTED	Ethernet ケーブルが切断されました EtherCAT バスセグメントがネットワークインターフェイスに接続されていません
0x9000012E	ACAT_ER_MST_MASTERCORE_INACCESSIBLE	マスタにアクセスできません
0x9000012F	ACAT_ER_MST_COE_MBXSEND_WKC_ERROR	CoE メールボックス送信エラー (WKC) CoE メールボックスはスレーブで読み取れません スレーブは最後の書き込み以降、メールボックスを読み取りませんでした
0x90000130	ACAT_ER_MST_COE_MBXRCV_WKC_ERROR	CoE メールボックス受信エラー (WKC) CoE メールボックスをスレーブから読み取りできませんでした
0x90000131	ACAT_ER_MST_NO_MBX_SUPPORT	スレーブはメールボックスアクセスをサポートしていません
0x90000132	ACAT_ER_MST_NO_COE_SUPPORT	スレーブは CoE をサポートしていません
0x90000133	ACAT_ER_MST_NO_EOE_SUPPORT	スレーブは EoE をサポートしていません
0x90000134	ACAT_ER_MST_NO_FOE_SUPPORT	スレーブは FoE をサポートしていません
0x90000135	ACAT_ER_MST_NO_SOE_SUPPORT	スレーブは SoE をサポートしていません
0x90000136	ACAT_ER_MST_NO_VOE_SUPPORT	スレーブは VoE をサポートしていません
0x90000137	ACAT_ER_MST_EVAL_VIOLATION	評価の構成に違反がありました
0x90000138	ACAT_ER_MST_EVAL_EXPIRED	評価は時間リミットに達しました
0x90000139	ACAT_ER_MST_LICENSE_MISSING	ライセンスキーが無効または欠落しています

EtherCAT マスタエラーコード(3/7)

エラーコード	名称	内容
0x90000201	ACAT_ER_MST_SDO_ABORTCODE_TOGGLE	トグルビットは交替されませんでした
0x90000202	ACAT_ER_MST_SDO_ABORTCODE_TIMEOUT	SDO プロトコルのタイムアウトが発生しました
0x90000203	ACAT_ER_MST_SDO_ABORTCODE_CGS_SCS	Client/Server コマンドが失敗または無効でした
0x90000204	ACAT_ER_MST_SDO_ABORTCODE_BLK_SIZE	無効なブロックサイズを指定しました
0x90000205	ACAT_ER_MST_SDO_ABORTCODE_SEQNO	無効なシーケンス番号を指定しました
0x90000206	ACAT_ER_MST_SDO_ABORTCODE_CRC	CRC エラー (BLOCK メモリのみ)が発生しました
0x90000207	ACAT_ER_MST_SDO_ABORTCODE_MEMORY	指定したメモリがメモリ範囲外でした
0x90000208	ACAT_ER_MST_SDO_ABORTCODE_ACCESS	オブジェクトへのサポートされていないアクセスを指定しました
0x90000209	ACAT_ER_MST_SDO_ABORTCODE_WRITEONLY	書き込み専用オブジェクトを読取りました
0x9000020A	ACAT_ER_MST_SDO_ABORTCODE_READONLY	読取り専用オブジェクトを書込みました
0x9000020B	ACAT_ER_MST_SDO_ABORTCODE_INDEX	オブジェクトがオブジェクトディクショナリに存在しません
0x9000020C	ACAT_ER_MST_SDO_ABORTCODE_PDO_MAP	オブジェクトが PDO マッピングできません
0x9000020D	ACAT_ER_MST_SDO_ABORTCODE_PDO_LEN	マッピングされるオブジェクトの数と長さが PDO の長さを超えています
0x9000020E	ACAT_ER_MST_SDO_ABORTCODE_P_INCOMP	一般的なパラメータと互換性がありません
0x9000020F	ACAT_ER_MST_SDO_ABORTCODE_I_INCOMP	デバイスの一般的な内部と互換性がありません
0x90000210	ACAT_ER_MST_SDO_ABORTCODE_HARDWARE	ハードウェアエラーが原因でアクセスに失敗しました
0x90000211	ACAT_ER_MST_SDO_ABORTCODE_DATA_SIZE	データ型が一致しません サービスパラメータの長さが一致しない
0x90000212	ACAT_ER_MST_SDO_ABORTCODE_DATA_SIZE1	データ型が一致しません サービスパラメータが長すぎます
0x90000213	ACAT_ER_MST_SDO_ABORTCODE_DATA_SIZE2	データ型が一致しません サービスパラメータが短すぎます
0x90000214	ACAT_ER_MST_SDO_ABORTCODE_OFFSET	サブインデックスがありません
0x90000215	ACAT_ER_MST_SDO_ABORTCODE_DATA_RANGE	パラメータ値が範囲外で書き込みに失敗しました
0x90000216	ACAT_ER_MST_SDO_ABORTCODE_DATA_RANGE1	パラメータ値が上限超で書き込みに失敗しました
0x90000217	ACAT_ER_MST_SDO_ABORTCODE_DATA_RANGE2	パラメータ値が下限超で書き込みに失敗しました
0x90000218	ACAT_ER_MST_SDO_ABORTCODE_MINMAX	最大値が最小値よりも低いです
0x90000219	ACAT_ER_MST_SDO_ABORTCODE_GENERAL	一般的なエラーが発生しました
0x9000021A	ACAT_ER_MST_SDO_ABORTCODE_TRANSFER	アプリケーションにデータを転送または保存できません
0x9000021B	ACAT_ER_MST_SDO_ABORTCODE_TRANSFER1	ローカル制御のため、アプリケーションにデータを転送または保存できません
0x9000021C	ACAT_ER_MST_SDO_ABORTCODE_TRANSFER2	現在のデバイスの状態では、アプリケーションにデータを転送または保存できません
0x9000021D	ACAT_ER_MST_SDO_ABORTCODE_DICTIONARY	オブジェクトディクショナリの動的生成に失敗したか、オブジェクトディクショナリがありません
0x9000021E	ACAT_ER_MST_SDO_ABORTCODE_UNKNOWN	不明なコード
0x9000021F	ACAT_ER_MST_SDO_ABORTCODE_MODULE_ID_LIST_NOT_MATCH	検出されたモジュール ID リスト [0xF030] と構成されたモジュール ID リスト [0xF050] が一致しません

EtherCAT マスタエラーコード(4/7)

エラーコード	名 称	内 容
0x90000220	ACAT_ER_MST_FOE_ERRCODE_NOTDEFINED	ベンダー固有の FoE エラー
0x90000221	ACAT_ER_MST_FOE_ERRCODE_NOTFOUND	FoE スレーブが見つかりません
0x90000222	ACAT_ER_MST_FOE_ERRCODE_ACCESS	FoE スレーブへのアクセスが拒否されました
0x90000223	ACAT_ER_MST_FOE_ERRCODE_DISKFULL	ディスク一杯です
0x90000224	ACAT_ER_MST_FOE_ERRCODE_ILLEGAL	違法なアクセスです
0x90000225	ACAT_ER_MST_FOE_ERRCODE_PACKENO	誤ったパケット番号です
0x90000226	ACAT_ER_MST_FOE_ERRCODE_EXISTS	すでに存在しています
0x90000227	ACAT_ER_MST_FOE_ERRCODE_NOUSER	ユーザーがありません
0x90000228	ACAT_ER_MST_FOE_ERRCODE_BOOTSTRAPONLY	BootStrap のみアクセスできます
0x90000229	ACAT_ER_MST_FOE_ERRCODE_NOTINBOOTSTRAP	ダウンロードしたファイル名は BootStrap 状態では無効です
0x9000022A	ACAT_ER_MST_FOE_ERRCODE_INVALIDPASSWORD	パスワードが間違っています
0x9000022B	ACAT_ER_MST_FOE_ERRCODE_PROGERROR	書き込みプログラムが異常です
0x9000022C	ACAT_ER_MST_FOE_ERRCODE_INVALID_CHECKSUM	チェックサムが異常です
0x9000022D	ACAT_ER_MST_FOE_ERRCODE_INVALID_FIRMWARE	ファームウェアが異常です
0x9000022E	ACAT_ER_MST_FOE_ERRCODE_NO_FILE	ファームウェアがハードウェアに適合しません

EtherCAT マスタエラーコード(5/7)

エラーコード	名称	内容
0x90000301	ACAT_ER_MST_CFGFILENOTFOUND	マスター構成ファイルへのパスが間違っていたか、ファイルが使用できません
0x90000302	ACAT_ER_MST_EEPROMREADERROR	スレーブ EEPROM から読み込みできません
0x90000303	ACAT_ER_MST_EEPROMWRITEERROR	スレーブ EEPROM へ書き込みできません
0x90000304	ACAT_ER_MST_XML_CYCCMDS_SIZEMISMATCH	周期コマンドの作成中にエラーが発生しました マスター構成ファイルで定義されたサイズとプロセスデータのサイズが一致しません
0x90000305	ACAT_ER_MST_XML_INVALID_INP_OFF	サイクルコマンドの入力オフセットが無効です。 InputOffs を確認してください
0x90000306	ACAT_ER_MST_XML_INVALID_OUT_OFF	サイクルコマンドの出力オフセットが無効です。 OutputOffs を確認してください
0x90000307	ACAT_ER_MST_PORTCLOSE	ポートのクローズに失敗しました
0x90000308	ACAT_ER_MST_PORTOPEN	ポートのオープンに失敗しました
0x90000309	ACAT_ER_MST_SLAVE_NOT_PRESENT	スレーブが消えた、または存在しなかった
0x9000030A	ACAT_ER_MST_NO_FOE_SUPPORT_BS	FoE プロトコルは Boot Strap をサポートしていません
0x9000030B	ACAT_ER_MST_EEPROMRELOADERROR	EEPROM のリロードに失敗しました
0x9000030C	ACAT_ER_MST_SLAVECTRLRESETERROR	スレーブコントローラのリセットに失敗しました
0x9000030D	ACAT_ER_MST_SYSDRIVERMISSING	システムドライバのオープンに失敗しました
0x9000030E	ACAT_ER_MST_BUSCONFIG_TOPOCHANGE	バスのスキャン中にトポロジが変更されました
0x9000030F	ACAT_ER_MST_EEPROMASSIGNERROR	EEPROM の割り当てに失敗しました
0x90000310	ACAT_ER_MST_MBX_ERROR_TYPE	メールボックスで不明なメールボックスを受信
0x90000311	ACAT_ER_MST_REDLINEBREAK	スレーブ間またはマスターと最初のスレーブ間のケーブルが切断されました
0x90000312	ACAT_ER_MST_XML_INVALID_CMD_WITH_RED	Redundancy の無効な EtherCAT コマンドを受信
0x90000313	ACAT_ER_MST_XML_PREV_PORT_MISSING	自動インクリメントアドレスがバス上の最初のスレーブではない場合、以前のポートが利用可能かどうかを確認します
0x90000314	ACAT_ER_MST_XML_DC_CYCCMDS_MISSING	DC 有効で、DC の周期コマンドが欠落しています
0x90000315	ACAT_ER_MST_DLSTATUS_IRQ_TOPOCHANGED	トポロジー変更によるデータリンク (DL) ステータス割り込みが発生しました
0x90000316	ACAT_ER_MST_FOE_MBX_WKC_ERROR	メールボックスで受信した FoE プロトコルのワーキングカウンターエラー
0x9000031C	ACAT_ER_MST_DC_REF_CLOCK_SYNC_OUT_UNIT_DISABLED	DC リファレンスクロックが無効です スレーブは ENI ファイルで基準クロックとして選択されていますが、スレーブには SYNC ユニットがありません
0x9000031D	ACAT_ER_MST_DC_REF_CLOCK_NOT_FOUND	DC リファレンスクロックが見つからない 基準クロックがネットワークから削除された場合に発生する可能性があります
0x9000031E	ACAT_ER_MST_XML_DC_REF_CLOCK_NOT_FIRST	[Sync Window Monitoring] 設定が有効で、最初のスレーブのリファレンスクロックが有効にならない
0x9000031F	ACAT_ER_MST_MBX_CMD_WKC_ERROR	MailBox コマンドのワーキングカウンターエラー メールボックスの初期化コマンドがリトライ回数を超えました

EtherCAT マスタエラーコード(6/7)

エラーコード	名 称	内 容
0x90000401	ACAT_ER_MAX_BUS_SLAVES_EXCEEDED	スレーブ数が最大数を越えています
0x90000402	ACAT_ER_MBX_SYNTAX	メールボックスヘッダーの構文が間違っています Slave error mailbox return value: 0x01
0x90000403	ACAT_ER_MBX_UNSUPPORTEDPROTOCOL	メールボックスプロトコルはサポートされていません Slave error mailbox return value: 0x02
0x90000404	ACAT_ER_MBX_INVALIDCHANNEL	無効なチャンネルが選択されています Slave error mailbox return value: 0x03
0x90000405	ACAT_ER_MBX_SERVICENOTSUPPORTED	メールボックスプロトコルヘッダーが間違えています Slave error mailbox return value: 0x04
0x90000406	ACAT_ER_MBX_INVALIDHEADER	メールボックスプロトコルヘッダーが間違えています Slave error mailbox return value: 0x05
0x90000407	ACAT_ER_MBX_SIZE_TOO_SHORT	受信したメールボックスの長さが短すぎます Slave error mailbox return value: 0x06
0x90000408	ACAT_ER_MBX_NO_MORE_MEMORY	リソースが限られているため、メールボックスプロトコルを処理できません Slave error mailbox return value: 0x07
0x90000409	ACAT_ER_MBX_INVALID_SIZE	無効なサイズです Slave error mailbox return value: 0x08
0x90000411	ACAT_ER_DC_SLAVES_BEFORE_REF_CLOCK	基準クロックの前にバス上に DC が構成されたスレーブがあります
0x90000412	ACAT_ER_DATA_TYPE_CONVERSION_FAILED	データ型の変換に失敗しました
0x90000413	ACAT_ER_LINE_CROSSED	ラインが交差しました
0x90000414	ACAT_ER_LINE_CROSSED_SLAVE_INFO	スレーブでラインが交差しました
0x90000415	ACAT_ER_SOCKET_DISCONNECTED	ソケットが切断されました
0x90000416	ACAT_ER_ADO_NOT_SUPPORTED	ADO はサポートされていません
0x90000417	ACAT_ER_FRAMELOSS_AFTER_SLAVE	スレーブ後にフレームロスが発生しています
0x90000418	ACAT_ER_FOE_ERRCODE_MAX_FILE_SIZE	最大ファイルサイズよりもファイルが大きい
0x90000501	ACAT_ER_DCM_NOT_INITIALIZED	DC モード初期化に失敗しました
0x90000502	ACAT_ER_DCM_MAX_CTL_ERROR_EXCEED	DC 同期通信に失敗しました
0x90000503	ACAT_ER_DCM_NO_MEMORY	十分なメモリが確保できません
0x90000504	ACAT_ER_DCM_INVALID_HW_LAYER	無効なハードウェアレイヤー
0x90000505	ACAT_ER_DCM_TIMER_MODIFY_ERROR	ハードウェアレイヤーエラー (タイマ補正)
0x90000506	ACAT_ER_DCM_TIMER_NOT_RUNNING	ハードウェアレイヤーエラー (タイマ未動作)
0x90000507	ACAT_ER_DCM_WRONG_CPU	ハードウェアレイヤーエラー (Wrong CPU)
0x90000508	ACAT_ER_DCM_INVALID_SYNC_PERIOD	DC 同期周期の長さが無効です
0x90000509	ACAT_ER_DCM_INVALID_SETVAL	DCM コントローラの SetVal が小さすぎます
0x9000050A	ACAT_ER_DCM_DRIFT_TO_HIGH	タイマとリファレンスクロックの間のドリフトが高い
0x9000050B	ACAT_ER_DCM_E_BUS_CYCLE_WRONG	バスサイクル時間 (dwBusCycleTimeUseC) が実際のサイクルと一致しない
0x9000050C	ACAT_ER_DCX_E_NO_EXT_CLOCK	外部同期クロックが見つかりません
0x9000050D	ACAT_ER_DCM_E_INVALID_DATA	DCM コントローラに無効なデータがあります

EtherCAT マスタエラーコード(7/7)

エラーコード	名 称	内 容
0x90000601	ACAT_ER_FOE_ERRCODE_FILE_HEAD_MISSING	ファイルヘッダが存在しません
0x90000602	ACAT_ER_FOE_ERRCODE_FLASH_PROBLEM	フラッシュプログラムが異常です
0x90000603	ACAT_ER_FOE_ERRCODE_FILE_INCOMPATIBLE	互換性のないファイルです
0x90000604	ACAT_ER_SDO_ABORTCODE_SI_NOT_WRITTEN	サブインデックスを書き込むことができません
0x90000605	ACAT_ER_SDO_ABORTCODE_CA_TYPE_MISM	ENUM オブジェクトタイプなどの可変長のオブジェクトでは、完全なアクセスはサポートされていません
0x90000606	ACAT_ER_SDO_ABORTCODE_OBJ_TOO_BIG	オブジェクトの長さがメールボックスのサイズを超えています
0x90000607	ACAT_ER_SDO_ABORTCODE_PDO_MAPPED	RxPDO にマッピングされたオブジェクトのため、SDO ダウンロードはブロックされました
0x90000608	ACAT_ER_OEM_SIGNATURE_MISMATCH	メーカー署名の不一致が発生しました
0x90000609	ACAT_ER_ENI_ENCRYPTION_WRONG_VERSION	ENI 暗号化アルゴリズムバージョンはサポートされていません
0x9000060A	ACAT_ER_ENI_ENCRYPTED	暗号化された ENI の読み込みには OEM キーセットが必要です
0x9000060B	ACAT_ER_OEM_KEY_MISMATCH	OEM キーが不一致です
0x9000060C	ACAT_ER_OEM_KEY_MISSING	OEM キーアクセスには OEM キーセットが必要です
0x90000620	ACAT_ER_S2SMBX_NOT_CONFIGURED	S2S メールボックスが設定されていません
0x90000621	ACAT_ER_S2SMBX_NO_MEMORY	S2S メールボックスにメモリがありません
0x90000622	ACAT_ER_S2SMBX_NO_DESCRIPTOR	S2S メールボックスに記述子がありません
0x90000623	ACAT_ER_S2SMBX_DEST_SLAVE_NOT_FOUND	S2S メールボックスの宛先スレーブが見つかりません
0x90000626	ACAT_ER_JUNCTION_RED_LINE_BREAK	ジャンクション冗長がラインブレイクしました
0x90000627	ACAT_ER_VALIDATION_ERROR	検証エラー (検証データの不一致) が発生しました
0x90000628	ACAT_ER_TIMEOUT_WAITING_FOR_DC	DC を待っているタイムアウトが発生しました
0x90000629	ACAT_ER_TIMEOUT_WAITING_FOR_DCM	DCM を待っているタイムアウトが発生しました
0x9000062A	ACAT_ER_SIGNATURE_MISMATCH	署名の不一致が発生しました
0x9000062B	ACAT_ER_PDIWATCHDOG	PDI ウォッチドッグの有効期限が切れました
0x90000701	ACAT_ER_CN_ALREADYMAILOPEN	[CUnet] メールが既にオープンされている
0x90000702	ACAT_ER_CN_MAILNOTOPEN	[CUnet] メールがオープンされていない
0x90000703	ACAT_ER_CN_MAILSIZE	[CUnet] メールサイズのエラー
0x90000704	ACAT_ER_CN_MAILSENDING	[CUnet] メール送信中のため受信できない
0x90000705	ACAT_ER_CN_MAILNOTRECEIVE	[CUnet] メールを受信していない
0x90000706	ACAT_ER_CN_MAILSENDRESULT	[CUnet] メール送信結果取得
0x90000707	ACAT_ER_CN_MAILRECEIVE	[CUnet] メール受信
0x90000708	ACAT_ER_CN_MAILCLOSE	[CUnet] メールのクローズが発行された

2-15-5 EtherCAT 用語説明

用語説明一覧

EtherCAT Technology Group (ETG)	: EtherCAT をオープン化し普及促進をはかる団体
EtherCAT State Machine (ESM)	: EtherCAT の通信状態遷移
EtherCAT Specification	: EtherCAT 仕様
EtherCAT Slave Information (ESI)	: EtherCATスレーブの設定情報が記述されたXML形式のファイル
EtherCAT Notification	: EtherCAT マスタ通知
ベンダーID	: ETG に登録される EtherCAT のベンダーID
プロダクトコード	: EtherCAT ユニットの製品コード
スレーブアドレス(スレーブ ID)	: ステーションエイリアスアドレス。 スレーブによって DipSW 等で設定されるアドレス。
フィジカルアドレス	: スレーブユニットのステーションアドレス。 マスタから設定されるアドレス。
ネットワーク情報ファイル (config.xml)	: 接続される全スレーブユニットの設定情報が記述された XML 形式のファイル (EtherCAT Network Information ファイルと同意)
ディストリビュートクロック (DC)	: EtherCAT スレーブとマスタを同期させるための分配クロック
オブジェクト	: デバイス内の特定の構成体
オブジェクトディクショナリ (OD)	: オブジェクトの記述が入ったデータ構造
サービスデータオブジェクト (SDO)	: ODにアクセスできるCoEの非同期メールボックス通信
インデックス	: ODに含まれるオブジェクトのインデックス
サブインデックス	: ODに含まれるオブジェクトのサブインデックス
プロセスデータ	: 周期的あるいは非周期的に転送されるアプリケーションオブジェクト
プロセスデータオブジェクト (PDO)	: 1 つ以上のプロセスデータエンティティを持つパラメータをマッピングすることによって記述される構造
入力 PDO (Receive PDO)	: EtherCAT スレーブで受信する PDO
出力 PDO (Transmit PDO)	: EtherCAT スレーブで送信する PDO
CAN application over EtherCAT (CoE)	: SDOまたはPDOによる通信を行い、オブジェクトディクショナリを参照
File access over EtherCAT (FoE)	: FoEプロトコルでのメールボックス通信でファイルのダウンロード、アップロードを行う

このリファレンスマニュアルについて

- (1)本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2)本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3)本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社もしくは、営業所までご連絡下さい。その際、巻末記載の書籍番号も併せてお知らせ下さい。

76DLH0047J

2020年 10月 第10版

 **株式会社アルゴシステム**

本社

〒587-0021 大阪府堺市美原区小平尾656番地

TEL (072) 362-5067

FAX (072) 362-4856

ホームページ <http://www.algosystem.co.jp/>