

【6】サンプリング・クロック値の設定

書式	int AD_Set_Clock(int clk_source, int set_mode, int *time_unit, int *clk_period)
引数	<p>clock__source: クロック源 / 0: 内部クロック源 0 (10.00MHz) 1: 内部クロック源 1 (8.192MHz) 2: 内部クロック源 2 (オプション) 3: 内部クロック源 3 (未使用) 4: 外部クロック源 (有効極性 =) 5: 外部クロック源 (有効極性 =)</p> <p>set__mode: クロック値の指定方法 / 0: クロック周期の値、 1: 分周比 time__unit: クロック周期の単位 / 0: s、 1: ms、 2: μs、 3: ns clk__period: クロック周期の値、または分周比</p>
戻り値	<p>正常終了時: = 0 エラー時: エラーコード / 負の値 (エラーコード表)</p>
機能・動作	サンプリングクロック値を設定する。なお、設定できないクロック周期の値を指定すると設定可能な長い方の近似値が設定される。また、クロック源に 2, 4, 5 を指定し、クロック周期の値で設定するときは前記【5】ad__set__exclckでクロック源周波数値を定義しておく。

【7】サンプリング・スタート

書式	int AD_Start_Samp (int no_samp)
引数	no__samp: サンプリング・データ点数 (1チャンネル当りの点数)
戻り値	<p>正常終了時: = 0 エラー時: エラーコード / 負の値 (エラーコード表)</p>
機能・動作	<p>サンプリングを実行する。 前記【4】トリガ設定でトリガ動作モードを《即トリガ》としてあるときは即スタート、また《ポストトリガ》が選択されているときはトリガ待ちスタートとなる。前記【4】で設定するレンジトリガのときは全てトリガ待ちスタートとなる。 なお前記【3】でサンプリングモードを自動に設定した場合、DLL内に確保されるデータバッファサイズは当関数で指定する1チャンネル当りサンプリング・データ点数と前記【2】で指定されるサンプリング実行チャンネル数の積となる。(1データ = 2バイト)</p>

【8】ステータス取得

書式	int AD_Get_Status (int *sampled, int status[])
引数	<p>sampled: サンプリング済みデータ点数 (1チャンネル当り) / DLL内部バッファ status[0]: ADボードのステータス1 / (3-13項) status[1]: ADボードのステータス2 / (未使用: 本ボードでは無い)</p>
戻り値	<p>正常終了時: = 0: 連続サンプリング実行中、 = 1: 停止中。 エラー時: エラーコード / 負の値 (エラーコード表)</p>
機能・動作	<p>当該時点でのサンプリング済みデータ点数 (DLLバッファ内)、 およびADボードのステータス生データを得る。(FIFOメモリ状態を含む)</p>

【9】本ハンドラの終了

書式	int AD_Close_ADSystem (void)
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	本ハンドラの終了、確保したメモリ領域の開放等を行う。

【10】DLL内バッファからユーザプログラム内バッファにADデータを転送

書式	int AD_Read_DLLData(int no_data, data_pos, WORD *bufptr, int bufsize)
引数	no_data : DLL内バッファから読み出すデータ点数 (1チャンネル当り) data_pos : DLL内バッファから読み出すデータの先頭位置 (サンプリング順番号) bufptr : 転送先ADデータバッファのポインタ bufsize : 転送先ADデータバッファの大きさ (バイト指定 = 総データ数の2倍)
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	サンプリングされたADデータがDLL内のバッファに格納されているとき、任意の部分をユーザプログラム内のADデータバッファに転送する。読み出すデータの末尾はステータス関数で得る<サンプリング済みデータ点数>の値以内でなくてはならない。 【注】当方法はサンプリング動作モード設定【3】で自動モードを指定したときだけ有効。

【11】ADボードからユーザプログラム内バッファに直接、ADデータを読み込む

書式	int AD_Read_DirectFifo (WORD *bufptr, int bufsize)
引数	bufptr : 転送先ADデータバッファのポインタ bufsize : 転送先ADデータバッファの大きさ (バイト指定 = 総データ数の2倍)
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	ADボード上のFIFOメモリから直接、ユーザプログラム内のデータバッファに転送する。一般的に当関数は割り込みイベント (【3】参照)、またはFIFOメモリ状態をポーリングして実行タイミングを得る。サンプリング動作モード設定【3】でADデータ転送の参照フラグにNot-Emptyを指定したときは1回サンプリングスキャン分のADデータを転送する。また、Half-Fullを指定したときはADボード上のFIFOメモリ容量の半分を転送する。 【注】当方法はサンプリング動作モード設定【3】でマニュアルモードを指定したときだけ有効。

【12】マニュアル（１回）サンプリング・スキャン

書式	int AD_Get_OneScan (WORD *bufptr, int bufsize)
引数	bufptr : ADデータバッファのポインタ bufsize : ADデータバッファの大きさ（バイト指定 = 総データ数の２倍）
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値（エラーコード表）
機能・動作	前記【２】で指定したチャンネル群に対して各１回だけサンプリングを実行する。

【13】汎用デジタル（ラッチ）出力

書式	int AD_Out_Aux (int out_data)
引数	out_data : 汎用デジタル（ラッチ）出力データ。 / ４ビット、
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値（エラーコード表）
機能・動作	汎用（ラッチ）出力データの更新。 当出力ポートは電源投入（ハードウェア・リセット）時：00H となるが、ボードの制御部リセット（ソフト的リセット）では変化しない。

【14】汎用デジタル（現在値）入力

書式	int AD_Inp_Aux (void)
引数	なし
戻り値	正常終了時 : = 汎用デジタル（現在値）入力データ。 / ４ビット、 エラー時 : エラーコード / 負の値（エラーコード表）
機能・動作	汎用デジタル（現在値）入力データの読み込み。

【15】データバッファをリング状 / エンドレスに設定

書式	int AD_Set_SampLoop (void)
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値（エラーコード表）
機能・動作	DLL内のADデータバッファをリング状・エンドレスで使用するか否かの設定。（トグル） リング状・エンドレスに設定した場合、自動モードでDLL内データバッファが満杯になると次は同バッファ先頭を上書きする形となる。この時、ステータス取得関数【８】で得るサンプリング済みデータ点数の値も先頭値（ゼロ）に戻る。 本設定によりADボードは無限サンプリング・モードとなる。 長時間の監視システム等に利用できる。

【16】アナログ入力モードの設定

書式	int AD_Set_Inpmode (int inp_mode, int ad_code, int ad_reso)
引数	inp_mode : アナログ入力信号形式 / 未使用 (本ボードはシングルエンド = 普通の2線式) ad_code : ADデータコード指定 / 0: バイナリ、1: 2の補数 ad_reso : ダミーデータ / 無視される。(他ドライバとの互換性を保つためにある)
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	アナログ入力信号形式、変換出力データ・コード (符号化の型) を設定する。

【17】サンプリング動作の強制停止

書式	int AD_Stop_Samp (void)
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	サンプリングを強制的に中止する。 本ハンドラではADデータ転送には割り込みを使用していないが、ユーザプログラム中で割り込みイベントを利用している場合は連続サンプリング終了後、必ず当関数を実行すること。

【18】残りADデータの読み込み

書式	int AD_Read_RestData ()
引数	なし
戻り値	正常終了時 : = 読み込んだ (残り) データ数 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	データロス・エラー発生 (当ハンドラが検出するとサンプリングを中止させる) したとき、ADボード上のFIFOバッファメモリに残りデータがあれば、これを当関数で読み込むことができる。

【19】(ADボード) 各フラグのクリア

書式	int AD_Clear_Flags ()
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	ADボードの制御部をリセットし、各フラグをクリアする。

エラー 本ハンドラの各関数は実行前後（または実行中）に不適当なパラメータや動作状態を検出するとエラーコードを返してきます。

表5 - 4 B . エラーコード一覧

戻り値	不具合の内容、または因果情報	適用関数、引数、等
- 1	ボードを検出できない。	AD_Open_ADSys ()
- 2	ドライバファイルを検出できない。	AD_Open_ADSys ()
- 3	ドライバファイルのバージョンが不適当。	AD_Open_ADSys ()
- 4	初期化が未実行。	
- 5	連続サンプリング開始前の設定不足。	AD_Start_Samp ()
- 6	Ownerパラメータの不適当。	AD_Open_ADSys ()
- 7	サンプリング実行チャンネル数 指定パラメータの不適当	no_ch
- 8	サンプリング・スキャン順 指定パラメータの不適当	scan_order []
- 9	アナログ入力レンジ 指定パラメータの不適当	range []
- 10	ADデータ転送時の参照フラグ 指定パラメータの不適当	trs_trig
- 11	ADデータ転送先 指定パラメータの不適当	buf_area
- 12	割り込みイベント発生要因 指定パラメータの不適当	intr_sw、
- 13	トリガ動作モード 指定パラメータの不適当	trg_mode
- 14	トリガ源 指定パラメータの不適当	trg_source
- 15	トリガ極性 指定パラメータの不適当	trg_pol
- 16	トリガ源選択 指定パラメータの不適当	trg_sel
- 17	トリガレベル（レンジトリガ以外） 指定パラメータの不適当	trg_level
- 18	トリガレベル（レンジトリガ） 指定パラメータの不適当	trg_level_hi/_lo
- 19	トリガレベル上限値・下限値の大小関係不適当	trg_level
- 20	クロック源 指定パラメータの不適当	clk_source
- 21	サンプリング・クロックの指定方法 パラメータの不適当	set_mode
- 22	クロック周期の単位 指定パラメータの不適当	time_unit
- 23	クロック周期の値、または分周比 指定パラメータの不適当	clk_period
- 24	アナログ入力信号形式 指定パラメータの不適当	inp_mode
- 25	ADデータ・コード 指定パラメータの不適当	data_code
- 26	AD変換分解能 指定パラメータの不適当	ad_reso
- 27	外部クロック源周波数を設定していない。	AD_Set_Clock ()
- 28	クロック分周比 指定パラメータの不適当	clk_period
- 29	ADデータバッファ・サイズが小さすぎる。	bufsize
- 30	1回サンプリングスキャン実行時、チャンネル設定の未実行	AD_Get_OneScan ()
- 31	bufptrエラー	bufptr
- 32	サンプリングされたADデータ数を越えて読もうとした。	AD_Read_Data ()
- 33	割り込みリソースが割り当てられていない。	AD_Set_SampMode ()
- 34	割り込み設定エラー（重複など）	AD_Set_SampMode ()
- 35	メモリ確保エラー	
- 36	ユーザプログラム内のADデータバッファが指定されている。	AD_Read_Dlldata ()
- 37	本ハンドラDLL内のADデータバッファが指定されている。	AD_Read_DirectFifo ()
- 38	連続サンプリング中なのに各種条件設定等しようとした。	
- 39	連続サンプリング中に残りデータ読み込み関数関数は不可。	AD_Read_RestData ()

WINDOWSハンドラ使用例

表 6 - 4 C . ハンドラ使用例

ファイル名	動作の概要	備考
Sample.DSW	チャンネル数、サンプリングクロック、サンプリング点数を指定し、即トリガ（即スタート）での動作例。 その他にADデータ転送先、割り込み要因の指定。	VC++ (5.0) 用
Sample.VBP		VB (5.0) 用

ハンドラ使用上の注意

- (1) 本ハンドラは不適切な使い方をするとエラーを返してきますから、原因が除去されるようにデバッグしてください。特に問題を起こしやすい点は、

ハード（ADボード）とソフト（ハンドラ）設定の不整合。

ADボードと他のハードの設定重複。： I/Oアドレス

割り込み番号

特にハード同志の重複設定による不具合は本ハンドラで検出できませんから、システムの構築時に充分な確認が必要です。

- (2) 本ハンドラは【AD_Open__ADSys ()】で使用開始、
【AD_Close__ADSys ()】で使用終了とします。

特に終了手続きは本ハンドラ内で確保したメモリの開放等を含むので重要です。

- (3) 本ハンドラの管理する割り込みは1系統のみです。
サンプリング実行時の《ボード上のFIFOバッファ》～《パソコン側データバッファ》間データ転送の実行タイミング検出には使用しておらず、単独の指定イベントによるメッセージ通知機能だけです。

- (4) データロスト・エラー

ADボードの自動サンプリング動作でFIFOバッファメモリにADデータが流入する速度よりパソコン側から読み出す速度が遅く、FIFOバッファメモリが満杯になったうえに次のデータが書き込まれようとする、そのデータはこぼれ（失われ）てしまいます。

これがデータロスト・エラーで、構築されたシステム全体の処理速度を超えたクロック値を指定して実行したときに起ります。なおデータロスト・エラーの発生を本ハンドラが検出すると連続サンプリングを強制停止させますが、それ以前のデータ（FIFOバッファメモリ中にある）は残りデータとして専用関数【18】で有効に読み出すことができます。

システムパフォーマンスはCPUを含むパソコン本体の実行速度や周辺機器の状態に左右されますが、（ユーザが記述する）本ハンドラの応用プログラムがサンプリング実行&データ読み込み（ブロック転送）に専念した場合は本ADボード自体の最高サンプリング速度が実現可能です。

- (5) 直接I/O操作を行うとき

本ハンドラを介さずADボードに直接I/O操作（OUT/INP）を行うと、本ハンドラの管理が行き届きませんから不本意な動作となることがあります。