技術ノート

Windowsにおける計測プログラミング（第2回）

広島市立大学情報科学部　藤原久志・石渡　孝

1. はじめに

今回を含めた2回を応用編とし，コンピュータ内に組み込んだインタフェースボードを利用した計測プログラムの解説を行う．まず今回は，タイマボードそしてAD変換ボードを用いた計測プログラムを紹介する．

解説に用いるインタフェースボードは，株式会社インタフェース[1]の製品を採用した．その理由は，同社の豊富な実績に加え，本社が広島にあり綿密な情報交換が可能なためである．そして，これまでの「仮想計測」とは異なり，実際の計測を取り扱うので，解説に特定の製品に依存した箇所が含まれることをご了承いただきたい．

2. 今回の解説概要

今回の解説記事では，まずインタフェースボードに関する基礎知識を概説する．次に，タイマボードを用いた計測プログラミングを話題とする．まず用いたボードと作成したサンプルプログラムを紹介し，続いて「正確な時間間隔」を保った計測について考察する．最後に，AD変換ボードを用いた計測プログラミングに言及する．まず用いたボードと作成したサンプルプログラムを紹介し，続いてデータの連続サンプリングのコーディングを論じる．特に後者を通じて，イベントオブジェクトとWaitForSingleObject関数を用いたプログラム動作の制御（同期）を，実例で解説する．

今回紹介するサンプルプログラム(ソースコード等)は，我々の用意したホームページ[2]よりダウンロード可能である．プログラミングの詳細については，簡潔性の観点から記事では扱えないので，ソースコードを読み込む事で掴んでいただきたい．このとき，インタフェース社のホームページより入手できる関連資料[3]（ユーザーズマニュアルやヘルプ[4]など）が必携である．また，実際に購入・使用を検討されている方は，事前の評価の為に同社の製品貸出サービスを利用すると良い．

3. インタフェースボードに関する基礎知識

3.1 インタフェースボード

インタフェースボード（図1）とは，狭義には「コンピュータと外部周辺機器（ハードディスクや計測装置など）との接続の仲立ちをする部品」と定義できる．しかし，今回用いるタイマボードもこれに含めるとすれば，拡張ボード（コンピュータに新しい機能を追加するための部品）と同義と考えられる．インタフェースボードは，次節で説明する拡張スロットに挿入して使用する．

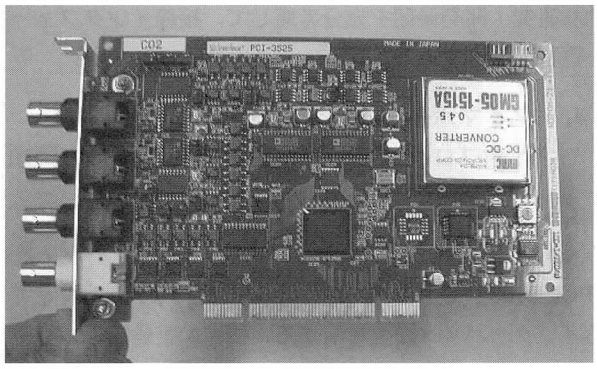

図1　インタフェースボード（写真は，今回の記事で用いたAD変換ボード）．

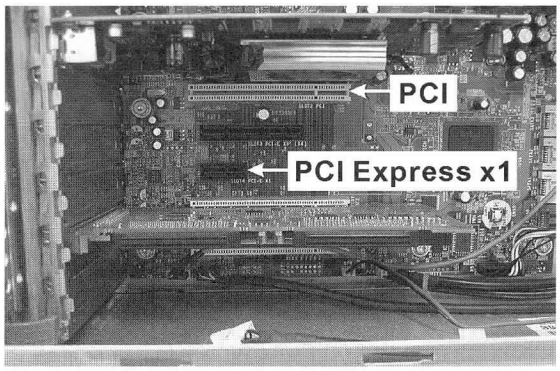

図2　本解説記事執筆時点で，最新のコンピュータにおける拡張スロットの状況例．

3.2 拡張スロットの動向

インタフェースボードと拡張スロットの接続インタフェースとして，PCI及びPCI Expressの二つが重要である[5,6]．

PCIは，1990年代初頭に登場した規格で，汎用コンピュータ向けには，32ビット幅，33 MHzで最大133 MB/sのデータ転送速度の規格が使われている[5,7]．一方，PCI Expressは2002年に次世代のPCI規格として決定された規格である[6]．汎用コンピュータ向け（インタフェースボードとの接続用）には，PCI Express×1（双方向で最大500 MB/sのデータ転送速度）の実装が増えてきている[8]．

本記事執筆時点で，PCIからPCI Expressへの移行過渡期であり，二つのインタフェースが混在するコンピュータも珍しくない（図2参照）．一方，インタフェースボードも，これまで主流であったPCI接続のものに加えて，PCI Express接続のボードが販売され始めている[1,8]．このような過渡期では，インタフェースボード或いはコンピュータの購入において，現有資産との整合性あるいは将来

的な研究・開発展開を充分考慮して，接続インタフェースを選択すべきである．

4. タイマボードを用いた計測プログラム：計測プログラミングにおける「時間」の問題

4.1 使用したボードについて

今回使用したタイマボードは，インタフェース社の「PCI-6103」である[9]．このボードは，1 μs から設定できるインターバルタイマを有する．後述の実験結果より明らかなように，正確な時間間隔で計測を実行するためには，このようなタイマボードが必要である．またボード内のフリーランカウンタを利用すれば，正確な時間計測も可能である．

なお，このボードは Hyper-Threading Technology (Intel) を有効にしている環境では，正常に動作しない可能性があり，注意が必要である[9,10]．

4.2 サンプルプログラムについて

今回用意したサンプルプログラムは，前回のサンプルプログラムと同一の実行画面と機能を有する．唯一の違いは，時間に関する諸処理（時間の計測とタイマ）を前回のサンプルプログラムは Win32 API 関数で処理し，今回のサンプルプログラムはタイマボードのライブラリ関数で処理する点である．従って双方のプログラムで，ソースコードを記述するファイルの数と名前は同一で，ファイルの中身も「time.h」と「time.cpp」のみ内容が異なっている．タイマボードのプログラミングは，比較的簡単であり，ソースコードを読む事での理解は容易と考えている．

4.3 一定の時間間隔での計測処理

計測プログラミングでは，一定の時間間隔で計測処理（データの取得・解析）をしたい場合がある．そのような場面を想定し，前回および今回のサンプルプログラムを同一のコンピュータ上で実行し，それぞれのタイマの動作状況を調べた．実験に用いたコンピュータの CPU は，Pentium 4 (Intel, 3.4 GHz) で，OS として Windows XP Professional Service Pack 2 (Microsoft) をインストールしている．

得られた結果を表1に示す．表より明らかなように，正確な時間間隔で計測処理を行うためには，タイマボードの使用が必要である．

表1 タイマ（Windows 及びタイマボード）による周期実行性．データは100回の測定結果を統計処理して得た．表中の数値の単位は，全てミリ秒

指定周期	Windows		タイマボード	
	平均	標準偏差	平均	標準偏差
100	109	3	100.00	0.01
10	16	5	10.00	0.04

4.4 短い時間間隔（10 ms）での計測処理

今回用いたタイマボードの注意事項として，「Windows では10 ms より短い周期でインターバルタイマ割り込みを使用するとアプリケーションが処理しきれずフリーズした状態に陥る場合があります」との記述がある[9]．これは，Windows ではプリエンプティブマルチタスキング[11-13]（OS が CPU 時間の各プログラムへの割り当てを配分・管理する）を採用しており，その OS の管理と割り込みとの「衝突」が生じていると考えられる．

さらに，Windows はリアルタイム OS ではないために，あるイベントから特定の時間内（例えば 1 ms 以内）で確実に特定のスレッドを実行させる方法は存在しない[12]．

以上より Windows 上では，10 ms より短い時間間隔を正確に保っての計測処理は難しいと考えられる．そのような計測処理を行うためには，リアルタイム OS[14] または DOS[15] の利用が考えられる．興味のある方は，提示した参考文献・資料を手がかりに自習いただきたい．

4.5 高速現象の計測（前節の補足）

前節で述べた結論は「10 ms より短い周期でのコンピュータによる計測処理は難しい」ということであり，高速現象の計測に Windows を搭載したコンピュータを使用できないという意味ではない．

例えば「10 Hz 発振レーザーで励起した分子の蛍光観測」の場合，蛍光減衰（～100 ns）曲線を光電子増倍管＋オシロスコープで取得し，これをコンピュータに転送し解析（例：最小二乗法を用いた寿命の算出）することは充分可能である．この場合「現象（蛍光減衰）は速いが，その発生周波数（10 Hz）は低い点」が肝心であり，コンピュータは低い発生周波数（＝長い発生周期）に合わせて計測処理（データ転送・解析）をすれば良いのである．

以上をまとめると，高速現象に対しては「高速なデータ取得を外部機器（インタフェースボードを含む）に任せ，そのデータをコンピュータに転送・処理する」という形で対応可能である．このとき「コンピュータの計測処理周期を短くしない事」が重要である．従って，観測対象の現象の発生周期が短い（～10 ms）場合には，「外部機器内部でのデータ保持または積算」などの工夫で，計測処理周期を長くする必要がある．そして，諸般の事情で計測処理周期を短くせざるを得ない場合には，前節で述べたように他の OS の検討が必要になる．

5. AD 変換ボードを用いた計測プログラム

計測プログラミングの一番の目的は「データをデジタルの形で取得・解析・保存すること」である．この観点から，AD 変換ボードは計測プログラミングで最も基本的なインタフェースボードと言える．

5.1　使用したボードについて

今回使用したAD変換ボードは，インタフェース社の「PCI-3525（図1参照）」である[16]．このボードは，12ビットAD変換（2チャンネル入力）で最高10 MHzのサンプリングが可能である（通常モードの場合）．また，入力端子がBNCコネクタであり，信号伝達にBNCケーブルを用いる事が多い計測プログラミングに好適である．さらに，ボードからコンピュータ内に確保したバッファへのデータ転送は，DMA（Direct Memory Access）転送[13]で行われるため，その間CPUは他の処理を行う事が可能である．

なお，このボードもHyper-Threading Technology（Intel）を有効にしている環境では，正常に動作しない可能性があり，注意が必要である[10,16]．

5.2　サンプルプログラムについて

今回のAD変換ボードに対しては，三種類のプログラムを作成した．

最初のプログラムは，先述のタイマボード用サンプルプログラムを拡張したもので，指定時間間隔毎にアナログ入力信号（初期設定：チャンネル1，以後二つのプログラムも同様）のAD変換データを1件取得し，表示・記録するものである．インタフェース社のサンプルプログラム（AdInputAD.c）[17]を利用し作成した．このプログラムは，ボードの初期化・設定・終了処理の分かり易い例となっている．

二番目のプログラムは，上記のプログラムでAD変換データを1件取得する箇所を，500件取得（サンプリング周波数10 MHz）に変更し，これに対応してデータ表示・記録箇所にも変更を加えたものである．一回に複数件数のデータを連続サンプリングする習作であり，ヘルプ[3,18]内のC言語のプログラム例とインタフェース社のサンプルプログラム（AdSmpl_C.c[19]内のR_SamplingDlgProc関数）を参考に作成した．

最後に三番目のプログラム（図3）は，二番目のプログラムをさらに改良し，一定時間間隔毎にトリガ機能を利用してデータ取得・表示・記録を行うプログラムである．その改良において，上述のヘルプ[18]の「トリガによるサンプリング開始」及び「非同期のサンプリング」に関する記述を参考にした．特に，非同期サンプリングについては，インタフェース社のサンプルプログラム（AdSmpl_C.c[19]内のE_SamplingDlgProc関数）も参考にした．図3のプログラムまで理解すれば，今回のボードの利用に必要な要素技術は習得可能と考えている．

5.3　データの連続サンプリング

今回のボードでのデータ連続サンプリングの手順（コーディング）は，次の通りである．

まずAdStartSampling関数[18]により，サンプリングバッファに対し，アナログ入力信号のAD変換データを指定件数サンプリングする．このとき，ボードからサンプリングバッファへのデータ転送は，DMA転送で行われる．

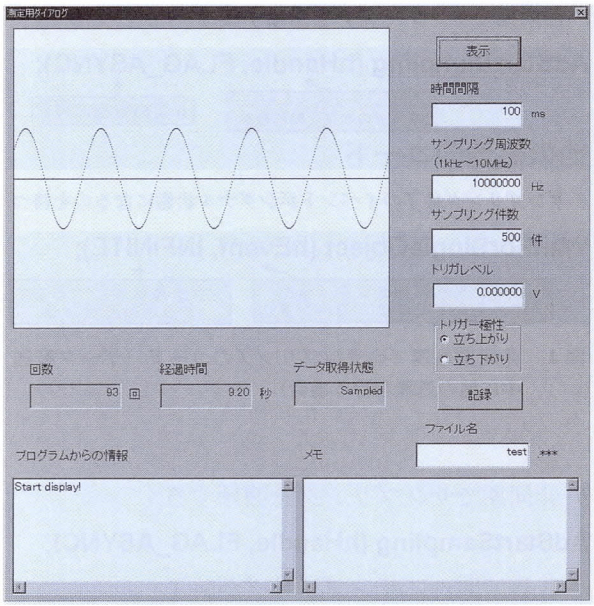

図3　AD変換ボードを用いたデータ取得プログラム（トリガ機能付）の実行画面．ファンクションジェネレータにより発生した振幅1 V，周波数100 kHzの正弦波を入力（50 Ω終端抵抗を有効）している．

次に，AdGetSamplingData関数[18]により，サンプリングバッファ内の12ビットAD変換データをWORD型（16ビット）の配列に写し取る．このWORD型配列内のデータは，計測者側のプログラミングによる解析処理（表示・保存を含む）が可能となる．

5.4　連続サンプリングにおける非同期処理（その1）

AdStartSampling関数によるサンプリングは，第2引数の識別子をFLAG_SYNCまたはFLAG_ASYNCとすることで，同期処理または非同期処理となる．同期処理では，関数呼び出し後に，サンプリング終了まで制御が戻らない．一方，非同期処理では，関数呼び出し後に直ちに制御が戻る．そして，サンプリングはDMA転送であるため，プログラム（CPU）はこれと並行して他の処理を行う事が可能である．

こうして非同期処理では，ボードの特長を生かしたデータ取得・処理を行うことができる．その実装例は，図4のようになる．AdStartSampling関数呼び出し後に，直ちに制御が戻り，「他の処理」を行う．これが終了すると，WaitForSingleObject関数によりサンプリングの終了（予め設定したイベントオブジェクトがシグナル状態になる）を検出する．すなわち「他の処理」終了時点で，サンプリングが終了していれば，WaitForSingleObject関数は直ちにこれを検出し，以降のコードが実行される．一方，サンプリングが終了していなければ，WaitForSingleObject関数はその終了を待ち続ける．

```
// 非同期でサンプリングを開始する
AdStartSampling (hHandle, FLAG_ASYNC);
                  デバイスハンドル（ボードの初期化時に取得）
                                    非同期処理の指定
他の処理のコード
// サンプリング終了のイベントがシグナル状態になるのを待つ
WaitForSingleObject (hEvent, INFINITE);
                  イベントオブジェクト（予めプログラム内で作成）のハンドル
                                    シグナル状態になるまでずっと待ち続ける
```

図4 非同期処理でのサンプリングのコード（データ転送中に他の処理を行う場合）.

```
// 非同期でサンプリングを開始する
AdStartSampling (hHandle, FLAG_ASYNC);
// 許容時間内にサンプリングが終了
// しなければ、サンプリング停止   許容待機時間
if (WaitForSingleObject(hEvent, time_out)
                == WAIT_TIMEOUT){
        // サンプリングを即刻停止
        AdStopSampling (hHandle);
}
else { // サンプリング終了した場合
        // サンプリングデータを配列に
        AdGetSamplingData 関数;
}
```

図5 非同期処理でのサンプリングのコード（サンプリング時間に制約がある場合）.

図4の例は，サンプリングおよび「他の処理」のそれぞれに要する時間に制約が無い場合に有効である．サンプリングに要する時間に制約がある場合（図3のプログラム）は，次節で取り扱う．

5.5 連続サンプリングにおける非同期処理（その2）

図3のプログラムでは，指定時間間隔（図では，100 msに設定されている）毎に，AdStartSampling関数を非同期処理指定で呼び出し，トリガによりデータ取得を行っている（トリガの設定方法については，ヘルプ[18]および我々のプログラムのソースコードを参照の事）．従って，100 ms以内にサンプリングが終了しないと（例えば，トリガがかからなかった場合），AdStartSampling関数を重ねて呼び出す事になり具合が悪い．

そこで，図3のプログラムでは許容待機時間（時間間隔の半分に指定）が経過した時点で，終了していないサンプリングを強制的に停止している．その実装の要約を図5に示す．WaitForSingleObject関数の第2引数をINFINITEでなく，許容待機時間を代入した変数（time_out）に指定し，関数の戻り値に応じて条件分岐する事で，プログラムの動作を制御している．

前節及び本節を総括すると，この二節では今回用いたAD変換ボードでの連続サンプリングにおける非同期処理を解説した．そして，より一般的な話題として，「イベントオブジェクトとWaitForSingleObject関数による同期」を，実例を通じて解説した．特に後者について，その「本質」を掴み取ることで，個々の案件で要請される「時間的制約」を満たすよう応用して頂ければ幸いである．

6. おわりに

今回は，タイマボードとAD変換ボードを用いた計測プログラムについて紹介した．ソースコードを読んで頂くと明白であるが，前回の仮想計測プログラムで紹介した要素技術以外は，MS-DOS時代と大きな変わりは無い．従って，特にその時代に活躍された方々には，必要とあらば大いに挑戦していただく事を願っている．

次回（最終回）は，コンピュータの外部の計測機器を制御する例として，IEEE-488（GPIB）を介したオシロスコープのデータ取得プログラムについて紹介する．

謝辞

本解説記事の作成に対し，広島市立大学特定研究費（一般研究）より助成を頂きました．また，株式会社インタフェースの関係者各位ならびに広島市立大学の高井博之先生には，様々な技術支援を頂きました．ここに深く感謝いたします．

参考文献・注釈

1) http://www.interface.co.jp/.
2) http://regulus.mtrl1.im.hiroshima-cu.ac.jp/~mtrl1/spsj/index.html. このホームページにアクセスできない場合は，管理責任者（石渡： ）に問い合わせて頂きたい．
3) ユーザID登録（無料）により，ダウンロード可能．
4) ボードに対応したソフトウェアのセットアップ（「Utility Disk」をダウンロード・利用して行う）により生成する．
5) 本当に役立つPC自作の基礎知識300（日経BP社，東京，2006）.
6) 伊勢雅英，パソコンの仕組み—最新テクノロジーと将来の動向（ソフトバンクパブリッシング，東京，2002）.
7) 主としてサーバー向けに，PCI-X（データバス幅が64ビット，動作周波数は最大133 MHz）と呼ばれる拡張規格もある．
8) グラフィックボード用には，PCI Express×16を用いるコンピュータ（マザーボード）が増えている．また，高速かつ大量なデータを扱う用途（例：高速・高解像度カメラ画像の取得）で，PCI Express×4を接続インタフェースとするボードも販売され始めている．
9) http://www.interface.co.jp/catalog/prdc.asp?name=pci-6103.

10) http://www.interface.co.jp/catalog/hyper_threading.asp.
11) Marshall Brain, Ron Reeves（訳：ドキュメントシステム），Win32システムサービスプログラミング（ピアソン・エデュケーション，東京，2002）.
12) Jeffrey Richter（訳：株式会社ロングテール，長尾高弘），Advanced Windows 改訂第4版（アスキー，東京，2001）.
13) Andrew S. Tanenbaum（訳：水野忠則，太田剛，最所圭三，福田晃，吉沢康文）モダンオペレーティングシステム（ピアソン・エデュケーション，東京，2004）.
14) 森友一朗，薬師輝久，馬場秀忠（著），木村新，岡本教佳（監修），RTLinux リアルタイム処理プログラミングハンドブック（秀和システム，東京，2000）.
15)「DOS プログラミングガイド～入門編～」（http://www.interface.co.jp/catalog/tutorial/tutorial_dos.asp）.
16) http://www.interface.co.jp/catalog/prdc.asp?name＝pci-3525.
17) AD 変換用ソフトウェアのセットアップ（注釈4参照）を行うと，C:¥Program Files¥Interface¥GPC3100¥samples¥vc¥InputAD に，AdInputAD.c ファイルは生成する.
18) GPC-3100［AD(PCI/C-PCI)］オンライン HELP.
19) C:¥Program Files¥Interface¥GPC3100¥samples¥vc¥AdSmpl_C.