

使えるインターフェースを活用しよう

レガシ・インターフェースを使った外部制御

シリアル・ポートとパラレル・ポート、さらにISAバスなどは、旧のPC/ATパソコンのインターフェースの総称としてレガシ・インターフェースと呼びます。これらのインターフェースを利用しないパソコンや周辺機器をレガシ・フリーと呼びます。ですが、デスクトップや一部のノート・パソコンではCOMポートとパラレル・ポートは採用されて続けています。また、部屋の隅で眠っている旧型機をちょっとした制御専用で再利用する場合などにはレガシ・インターフェースの利用を考慮する価値があります。本章では、COMポートとパラレル・ポートを使って簡単に実現できる外部制御について紹介します。

5.1 COMポート

PC/ATパソコンのシリアル・ポートには、25ピンの標準的ないわゆるRS-232Cポートでなく独自の9

表5.1⁽¹⁾ シリアル・ポートのコントロール線を操作するAPI

パラメータ		説明
<pre> BOOL GetCommModemStatus(HANDLE hFile, LPDWORD lpModemStat); BOOL EscapeCommFunction(HANDLE hFile, DWORD dwFunc); </pre>		
hFile	CreateFile() で得られたシリアル・ポートのハンドル	
lpModemStat	MS_CTS_ON	CTSのチェック
	MS_DSR_ON	DSRのチェック
	MS_RING_ON	RING(RI)のチェック
	MS_RLSD_ON	CDのチェック
dwFunc	制御コード	
	CLRDTR	DTRのクリア
	CLRRTS	RTSのクリア
	SETDTR	DTRのセット
	SETRTS	RTSのセット
	SETXOFF	XOFF制御を行う
	SETXON	XON制御を行う
	SETBREAK	BREAK制御を行う
CLRBREAK	BREAK制御を解除する	
戻り値	成功すると、0以外を返す	
解 説		
GetCommModemStatus() はシリアル・ポートのコントロール線の状態を得る。 EscapeCommFunction() はシリアル・ポートのコントロール線とフロー制御を操作する		

ピンのポートが用意されています。このポートには、シリアル・データ線以外に6本の制御線と呼ばれる入出力線があります。シリアル通信を使用しないのなら、この6本の制御線は独立した入出力線として利用できます。

この制御線は表5.1に示すWindows APIから制御することができます。ですから、見方を変えると、この制御線はWindowsから直接制御できる唯一の外部入出力線ということになります。

制御線と入出力ボードとの接続

シリアル・ポートは、データ線、制御線とも一般的なデジタル機器と異なり $\pm 15V$ の電圧で0と1を表します。ですから、この変換のための仕組みが必要です。この信号レベル変換にはトランジスタやHC-TTLを利用するという方法もありますが、ここで紹介する変換ボードでは一般的なレベル変換ICを利用します。レベル変換ICは変換用インバータが四つ入っているものが標準ですから、六つある制御線のう

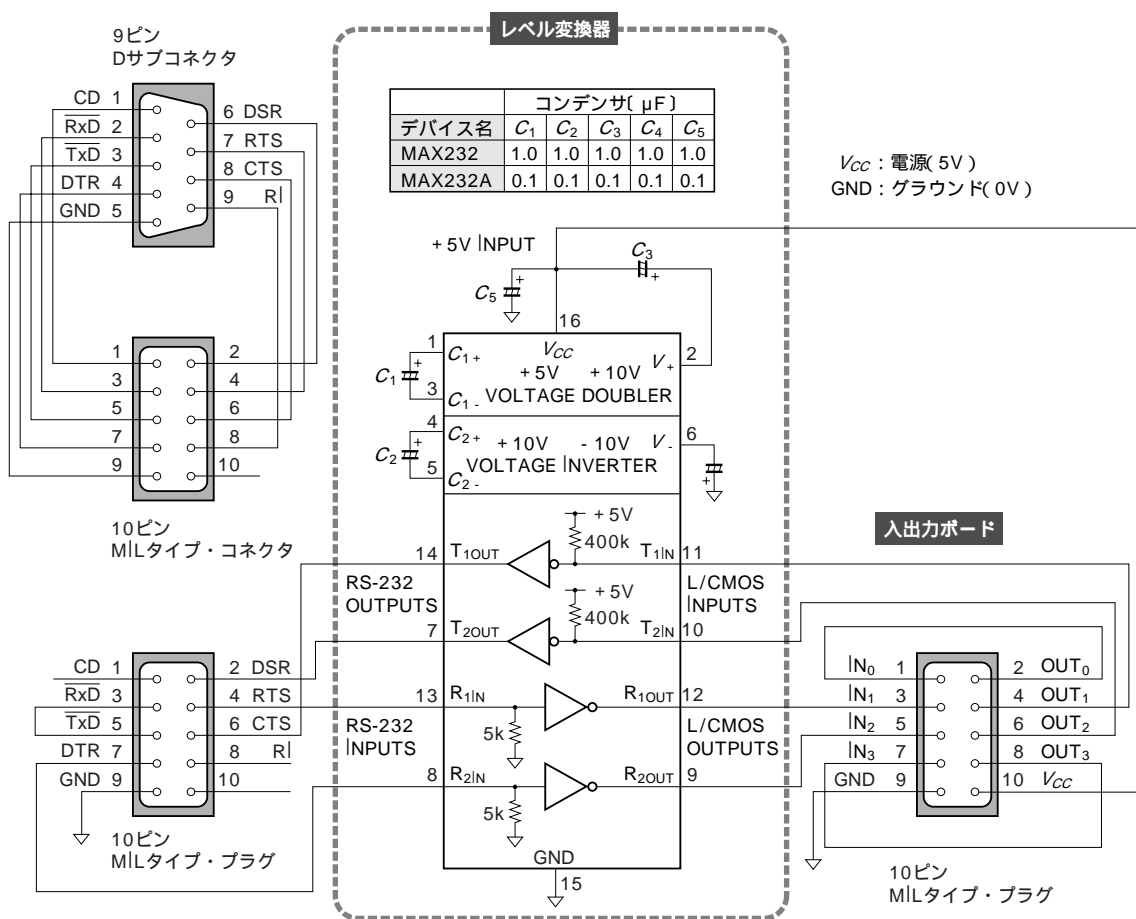


図5.1 変換ボードの回路図

Dサブコネクタは10芯のリボン・ケーブルを使いDサブコネクタ側でリボン・ケーブルの線を1本未接続にすることで、10ピンのMILコネクタと接続し変換ケーブルとしている(写真5.4参照)。

図の回路ではレベル変換ICを一つ使っている。残りのTxD, RxD, CD, RIの各信号も利用するのであれば変換ICを増やせば対応できる。シリアル通信や入出力ボードのチェック用に、RxD-TxD, IN₀-OUT₀, IN₃-OUT₃は直接接続している。

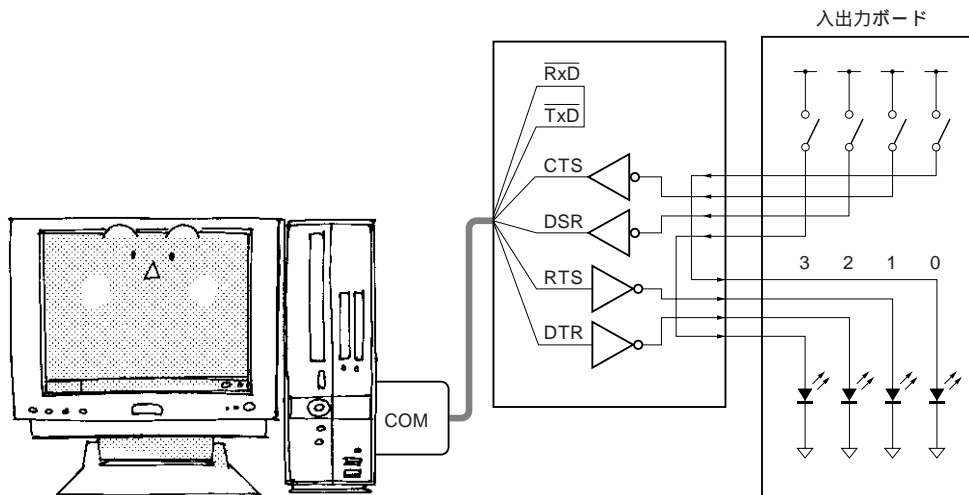


図 5.2 COM ポートの制御線を使って構成された実験用システム

変換ボードを使用するとインバータを介することになるので，入出力の論理は負論理となる．

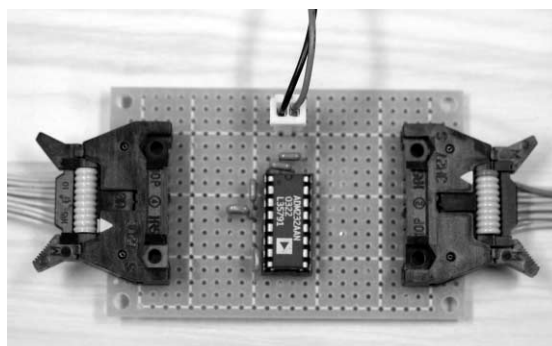


写真 5.1 製作した変換ボード

ち RTS, DTR, CTC, DSR の 4 本を変換することになります．レベル変換 IC をもう一つ使えば，残りの 2 本の入力線も利用可能になります．

変換ボードの回路図を 図 5.1 に示します．この変換ボードを利用して入出力ボードを COM ポートに接続すると，パソコンからは 図 5.2 のようなシステムとして見えます．製作した変換ボードを 写真 5.1 に示します．

LED 点滅とスイッチのエコー動作プログラム

前章の拡張インターフェースの使用と同様に，簡単な入出力プログラムを試すことにします．COM ポートの制御線を使用する場合も，デバイスをオープンして，そのハンドルを使って入出力関数で実際の入出力を行うという処理の流れの大枠は同じです．COM ポートの場合，デバイスのオープンには Windows API の `CreateFile()` を使います．その後の入出力は 表 5.1 の `GetCommModemStatus()` と `EscapeCommFunction()` を利用します．`EscapeCommFunction()` はこれまで紹介したインターフェースの入出力関数と異なり，1 回の操作で 1 ビットのみでの出力操作となります．

リスト5.1 COMポートによるLED点滅とスイッチのエコー動作プログラム：list0501.cpp

```

//
// list0501.cpp LEDの点滅, スイッチのエコー
// ~ COMポートの制御線を使った外部制御の基礎
//
#include<windows.h>
#include<stdio.h>
#include<conio.h>

main()
{
    HANDLE h;
    DWORD x;
    int i;

    printf("COMポートのテスト1 ¥n");

    h = CreateFile( "COM4", // ポート名
                   0, 0, // アクセス・モード, 非共有
                   0, // セキュリティ属性, 使用しない
                   OPEN_EXISTING, // 既存ファイルのオープン
                   0, 0 ); // 属性, テンプレート

    if( h == INVALID_HANDLE_VALUE ){
        printf("COMポートがオープンできません¥n");
        return 0;
    }

    printf("点滅動作¥n");
    for( i = 0 ; i < 10 ; i++ ){ // 10回点滅, 周期1秒
        printf( "%d In", i );
        EscapeCommFunction( h, CLRRTS );
        EscapeCommFunction( h, CLRDTR );
        Sleep( 500 ); // 時間待ち500ms

        EscapeCommFunction( h, SETRTS );
        EscapeCommFunction( h, SETDTR );
        Sleep( 500 ); // 時間待ち500ms
    }
    printf("エコー動作¥n");
    while( 1 ){
        GetCommModemStatus( h, &x );
        if( x & MS_CTS_ON )
            EscapeCommFunction( h, SETRTS );
        else
            EscapeCommFunction( h, CLRRTS );
        if( x & MS_DSR_ON )
            EscapeCommFunction( h, SETDTR );
        else
            EscapeCommFunction( h, CLRDTR );
    }
}

```