

DOSの入出力機器をWindowsで使用する 入出力教材の作成

神奈川県立産業技術短期大学校 制御技術科 田巻 愛

1. はじめに

現在使用されているパソコンのOSは、そのほとんどがDOSからWindowsに移行した。機器の制御にパソコンを使用する場合においてもWindows上で作業する機会が増加している。

一方Windowsで入出力を行うとき、シリアルポートなどパソコンに標準装備のデバイスを利用した方法が存在するが、それ以外の入出力方法はサポートされていない。このため、DOSで使用してきた機器のようにインターフェースボードを介してパソコンから入出力を行う方法が明確でない。そこでDOSからWindowsに移行する場合を想定し、開発ツールとしてMicrosoft社のVisual C++(以下、「VC」という)を使用して、パソコンからの入出力を2日間で学習する教材を作成した。さらに、この教材を活用した講習の結果についてあわせて報告する。

2. デバイスドライバについて

2.1 Windowsにおける入出力の方法について

WindowsではI/Oへアクセスする方法はサポートされていない。そのため、基本的にはデバイスドライバを通してI/Oにアクセスする。例外としてWindows95であればI/O命令が可能である。しかし、この方法では、Windows95上のほかのアプリケーションが同一のI/Oを使用すると、競合が発生する問題がある。また、Windows NT上ではデバイスドライバなしでは入出力を行うことはできない。

これらより、入出力アプリケーションの作成方法

として、デバイスドライバを使用した方法を取り上げることにした。

図1に示すようにデバイスドライバは常駐しI/Oとメモリのやりとりを行う。出力を行う場合、アプリケーションがデバイスドライバへデータを一度送り、デバイスドライバは対応するI/Oへデータを出力する。入力も基本的に同様に行うことができる。このI/Oを、インターフェースボードのI/Oと一致させることで入出力が行えるようになる。

2.2 実際の入出力方法

教材作成のため表1に示す2製品について、デバイスドライバの比較を行った。

両製品とも入出力や割り込みをサポートしており、互換機、PC98シリーズに対応している。両者の違いは、API-DIOがContec社製インターフェー

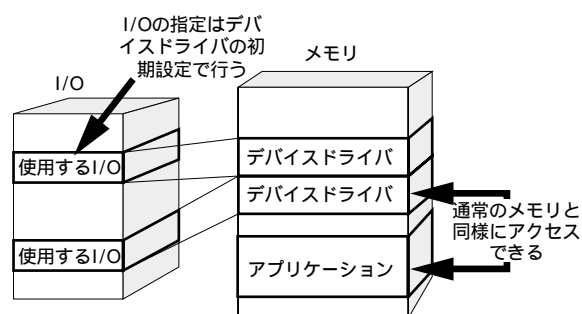


図1 デバイスドライバ動作イメージ

表1 比較したデバイスドライバ

製品名	メーカー名	動作機種
API-DIO	CONTEC	互換機, PC98
Win-RT	ToolCraft	互換機, PC98

スボード用であるのに対して、Win-RTは汎用のデバイスドライバであるという点である。つまり、Win-RTは自作インターフェースボードなどにも対応可能である。

次に、実際に入出力アプリケーションの作成過程を説明する。全体の流れは図2のとおりである。

デバイスドライバの設定では、I/Oポートアドレスと割り込みの使用に関する設定を行い、デバイスドライバを常駐させる。

各開発ツールでのアプリケーション作成では、Windowsアプリケーション開発ツール上でデバイスドライバの使用に必要な環境設定を行う。具体的にはインクルードファイル、ライブラリファイルの設定である。これらの設定を終了するとアプリケーションの作成を行えるようになる。

2.3 アプリケーション作成の一例

次にデバイスドライバを使用したアプリケーション作成の具体例について説明する。ここでは開発ツールとしてVC、デバイスドライバとしてAPI-DIOを使用した場合について説明する。この例ではアプリケーションの作成方法にWindows API関数を使用する。そのため、プロジェクトの新規作成でApplicationを選択し、プロジェクト名を入力する(図3参照)。

次に、ダイアログ(図4)、Cソース(図6)を作成する。

さらに先ほど作成したプロジェクトにこれらのファイルと、API-DIO用関数のライブラリであるApidio.libを挿入する(図5参照)。

これら設定後、ビルドを行うことが可能となる。

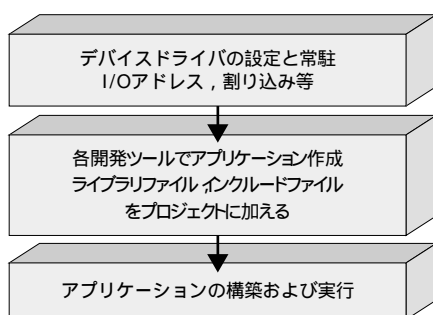


図2 デバイスドライバを用いたアプリケーション作成の流れ

このアプリケーションは、入出力確認装置AB10-EXE(後述)のボタンを押して、アプリケーションのOKボタンを押すと、押しているボタンに対応する位置のLEDを点灯する機能を持つ。

Win-RTを使用した場合も同様の流れである。Win-RTにはプリコンパイラが付属しており、よりDOSに近い形で記述が可能である(図7参照)。

2.4 デバイスドライバの選定

選定するうえで機能と価格がポイントになる。機能的にWin-RTは汎用性が高いという点で優れている。一方で価格的にはAPI-DIOのほうが廉価である。講習で使用するインターフェースボードでAPI-DIO、Win-RT両者の動作が可能であることを確認



図3 プロジェクトの作成

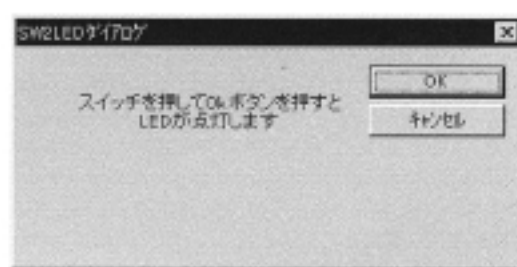


図4 作成したダイアログ



図5 プロジェクトに含まれるファイル

した。そのため、講習では廉価なAPI-DIOを使用することにした。

3. 教材の作成

講習を実施する環境は表2に示すとおりである。

受講対象者をDOSからWindowsへの移行を考えている方とした。そのため、開発言語としてDOS, Windowsで共通して使用されるC言語を選択した。

入出力の確認装置としてパソコンに接続するハードウェアにはAB10-EXEを使用した。これはLED, 押しボタンスイッチ, ステッピングモータ, DCモータ, ポテンショメータが実装されており, パソコンから入出力を直接確認できる装置である。

作成するアプリケーションの内容を使用頻度の高いダイアログベースに絞り, 時間内にプログラミングと入出力の両方を学習できるように配慮した。

講習会全体の流れは図8に示すように4つのステップから構成されている。全体の流れはDOS上の

知識を確認した後, Windowsプログラムの基礎的な内容の学習, 入出力に必要なデバイスドライバを使用した学習を行う。

ステップ1では, DOSのCプログラムを組みながら, ポインタや構造体の知識の確認を行った。

ステップ2では, シンプルなWindowsアプリケ

```

%%
        outp(PPI1CREG, 0x90);

%%
        ユーザ記述のコード

WINRT_CONTROL_ITEM_WinRTpp01]=
//      command  param1  param2
        {OUTP_B, 0,      0x00,
};

_WinRTpp01[0].port = PPI1CWRG;

(void)WinRTProcessIo3Buffer(hWinRT, _WinRTpp01,
sizeof( _WinRTpp01), &iWinRTlength);

プリコンパイル後のコード

```

図7 Win-RTを使用した場合の記述

```

//SW2LED.c
//AB10-EXEの押しボタンスイッチを押すと対応する位置のLEDが点灯する
//API Configurationで
//P1048x(98)x
//DrvNo = 1,
//GrpNo = 1 PortAddress = 0xd0 NonInterrupt
//GrpNo = 2 PortAddress = 0xd8 NonInterrupt
//を設定した後実行してください。
#include <windows.h>
#include <stdio.h>
#include "resource.h"
#include "a:\contecw95dio\vsamples\vc\c\api_dio.h" //API-DIO用インポートファイル

HINSTANCE hInst; //現在のインスタンスのハンドル
BOOL CALLBACK dlgProc(HWND hwndDlg, UINT msg, WPARAM wParam, LPARAM lParam);

//main関数
int PASCAL WinMain(HINSTANCE hinstCurrent, HINSTANCE hinstPrevious,
LPSTR lpszCmdLine, int nCmdShow)
{
    DLGPROC pProc;
    pProc = (DLGPROC)MakeProcInstance(dlgProc, hinstCurrent);
    if ( pProc != NULL )
    {
        int retco = DialogBox(hinstCurrent, MAKEINTRESOURCE(IDD_DIALOG1),
            NULL, pProc);
        FreeProcInstance((FARPROC)pProc);
    }
}

BOOL CALLBACK dlgProc(HWND hDlg, UINT msg, WPARAM wParam, LPARAM lParam)
{
    //I/O関連の宣言
    static HANDLE hDrv1, hDrv2; //デバイスドライバへのハンドル
    static WORD DrvNo, GrpNo;
    static DWORD dwRet;
    static WORD CNx, CtrlWord;
    static DOUT LpDOut; //出力用構造体
    static DIMP LpInp; //入力用構造体
    static BYTE InBuf[2], OutBuf[2]; //データバッファ配列
    static WORD InPortNo[2], OutPortNo[2]; //ポート番号格納用配列

    switch(msg)
    {
        //ダイアログの初期処理
        case WM_INITDIALOG:
            //DIOの初期化
            //デバイスドライバへのアクセス方法の確認と
            //ハードウェア(8255-1, 8255-2)へのハンドルの初期化
            DrvNo = 1;
            GrpNo = 1;
            dwRet = DioOpen(&hDrv1, DrvNo, GrpNo);
            //ドライバ番号1のグループ番号1をhDrv1とする
            //これは8255-1へのハンドルになります
            GrpNo = 2;
            dwRet = DioOpen(&hDrv2, DrvNo, GrpNo);
            //ドライバ番号1のグループ番号2をhDrv1とする
            //これは8255-2へのハンドルになります
            CNx = 0;
            CtrlWord = 0x90; //
            dwRet = Dio8255Mode(hDrv1, CNx, CtrlWord);
            //8255-1のコントロールワードを0x90にする
            CtrlWord = 0x80; //
            dwRet = Dio8255Mode(hDrv2, CNx, CtrlWord);
            //8255-2のコントロールワードを0x80にする
            }
        }

        }
        return(TRUE);
    }
    // コマンドの処理
    case WM_COMMAND:
        switch(wParam)
        {
            case IDOK: //OKボタンが押されたときの処理を記述
                //スイッチからのデータを入力し
                //LEDに出力します
                {
                    //8255-1 Aportから入力(押しボタンスイッチ)
                    //入力用関数で使用する構造体のメンバ
                    //typedef struct tagDIMP {
                    // LPWORD InpPortNo: 入力ポート番号格納配列の先頭アドレス
                    // WORD PortNum; 入力ポートの数
                    // LPBYTE Buf; 入力データの格納配列の先頭アドレス
                    //} DIMP;
                    //DWORD FAR PASCAL DioInp(HANDLE, DIMP FAR *);
                    //Aポートを選択
                    InPortNo[0] = 0;
                    LpInp.InpPortNo = &InPortNo[0];
                    //論理ポート総数
                    LpInp.PortNum = 1;
                    //入力データの格納アドレス指定
                    LpInp.Buf = &InBuf[0];
                    //実行
                    DioInp(hDrv1, &LpInp);
                }
            //8255-1 Bportへ出力(LED)
            //出力用関数で使用する構造体のメンバ
            //typedef struct tagDOUT {
            // LPWORD OutPortNo: 出力ポート番号格納配列の先頭アドレス
            // WORD PortNum; 出力ポートの数
            // LPBYTE Buf; 出力データ格納配列の先頭アドレス
            //} DOUT;
            //HANDLE hDrv;
            //DOUT LpDOut;
            //DWORD dwRet;
            //dwRet = DioOut(hDrv, &LpDOut);
            //Bポートを選択
            OutPortNo[0] = 1;
            LpDOut.OutPortNo = &OutPortNo[0];
            //論理ポート総数
            LpDOut.PortNum = 1;
            //出力データを設定
            OutBuf[0] = InBuf[0];
            LpDOut.Buf = &OutBuf[0];
            //出力の実行
            DioOut(hDrv1, &LpDOut);
            } break;
            case WM_CLOSE:
            case IDCANCEL:
                DioClose(hDrv1); //8255-1のハンドルのクローズ
                DioClose(hDrv2); //8255-2のハンドルのクローズ
                EndDialog(hDlg, TRUE);
                return TRUE;
            }
        }
    default:
        return(DefWindowProc(hDlg, msg, wParam, lParam));
    }
    return FALSE;
}

```

図6 Cソースファイル

表2 訓練実施環境

コンピュータ	PC9821-Xv13
OS	Windows95
開発言語	MS-VisualC++Ver4.0
デバイスドライバ	ContecAPI-DIO Win95(PC98用)
ハードウェア	AB10-EXE

表3 作成したテキストの概要

概要	内容
C言語基礎	Cの演算子等をまとめたもの
Windowsアプリケーション作成	基礎プログラミングとI/Oを含めたもの
タイマの使用方法	DOSタイマ割り込みとWindowsマルチメディアタイマの使用方法

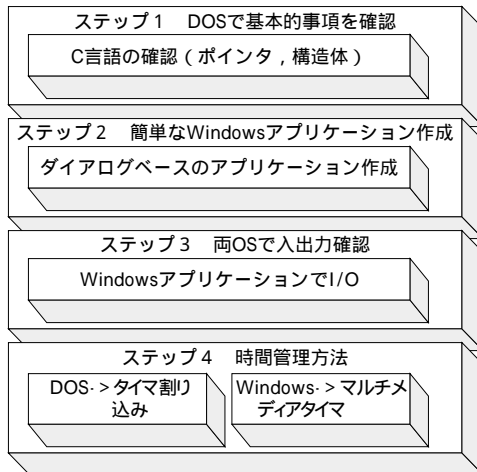


図8 カリキュラム



図9 テキストの一例

ーションを作成し、Windowsアプリケーションの基本的な構造を知ることができる。

ステップ3では、デバイスドライバを使用してI/O入出力を確認するアプリケーションを作成し、入出力に必要なデバイスドライバへのアクセス方法を知ることができる。

ステップ4では、信号処理で大きな要素となる時間管理方法を学習する。DOSではTimer割り込み、Windowsではマルチメディアタイマを使用し1[ms]単位で時間を管理する。具体的にはステップモータの駆動を取り上げ、回転速度を変える内容を題材とした。

作成したテキストは表3に示すような内容になっている。

テキストは図9に示すようにMS-PowerPointで作成し、同時にサンプルプログラムをテキスト形式で配布した。これにより、必要なときにデータを参照し、自学自習形式でプログラムを作成できる。受講者は配布したサンプルプログラムに手を加えること

でプログラムを作成していくことが可能である。

テキストはプログラミングの注意点および、プログラミングに必要なデータを中心に構成した。

4. 講習の実施と改善点

2日間の講習を実施したが、予定の内容を修了した講習生はいなかった。これは2日という講習時間に対して内容が多いためであると思われる。よって、以下のような内容の改善が必要である。

- ・講習日数を2日から4日へ変更する
- ・プログラミングに必要な労力を減らすために、開発言語を検討する

5. おわりに

DOSからWindowsへの移行で最も問題になるのがI/Oへの直接アクセスである。

DOSで比較的手軽にできたI/OへのアクセスをWindowsでいかに簡単に実現できるかに的を絞ったつもりである。しかし、まだまだ私自身のWindowsプログラミングの経験が足りず、テキスト、プログラムともに改善が必要であると考えている。各方面からご意見等をいただければ幸いである。