

TR-H-269

シリアルポートを使った機器の制御

川村真宣

1999.3.10

ATR人間情報通信研究所

〒619-0288 京都府相楽郡精華町光台2-2 TEL: 0774-95-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

Telephone: +81-774-95-1011

Fax : +81-774-95-1008

シリアルポートを使った 機器の制御

川村真宣

99.2.26

概要

このレポートは、SONYのパンティルタカメラ(EVI-G20,EVI-D30)を例に、シリアルポートを使用して通信や機器の制御をするための、基本的な方法を紹介します。「アプリケーションの作成」の項目に示したとおりにプログラムを書き込むだけで、簡単に制御プログラムを作ることが出来ます。Borland C++ Builderの使い方が分からない人でもプログラムが出来るような構成にしているつもりですが、もし分からない場合は、『すぐわかるC++ Builder』（参考文献1）などの入門書を参考にしてください。

目次

概要	1
目次	1
必要な機器を準備する	2
機器を接続する	2
アプリケーションを作成する	3
カメラが動かない場合	7
VISCAケーブルを作る	8
RS-422規格のシリアルポート	8
サンプルプログラム	8
参考文献	9
インターフェースコンバータの回路図	10



必要な機器を準備する

初めに、必要な機器を準備します。ここでは、SONYのパンティルタ(EDI-D30)を例に、コントロールするプログラム作成の手順を説明します。コンピュータのセットアップや、コンパイラのインストールなどは、それぞれのマニュアルを参照して下さい。

表1 準備するもの

コンピュータ	MS-Windows 95/98/NT をインストールしたPC/AT互換機
制御の対象	パンティルタ
ケーブル	VISCAケーブル
コンパイラ	Borland C++ Builder

✂ 機器を接続する

下の図のように、コンピュータとカメラを接続します。

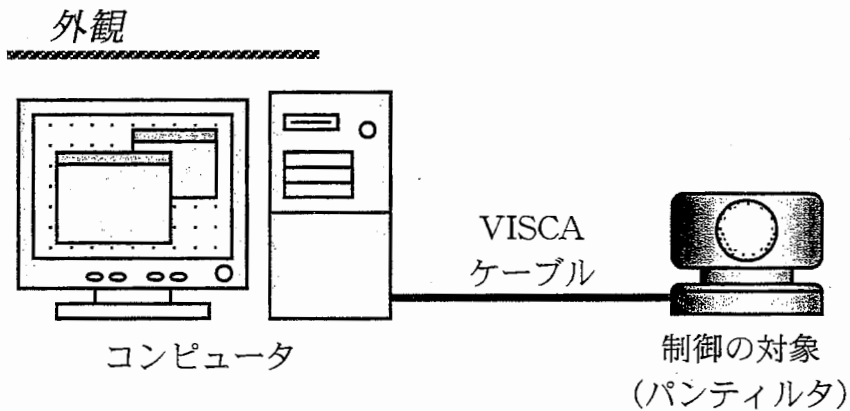


図1 機器の接続

☞ アプリケーションを作成する

まずはBorland C++ Builderを起動します。Borland C++ Builderを起動すると、中央にコードエディタとフォームが、一番上にメインメニューウインドウ、左側にオブジェクトインスペクタがあらわれます。

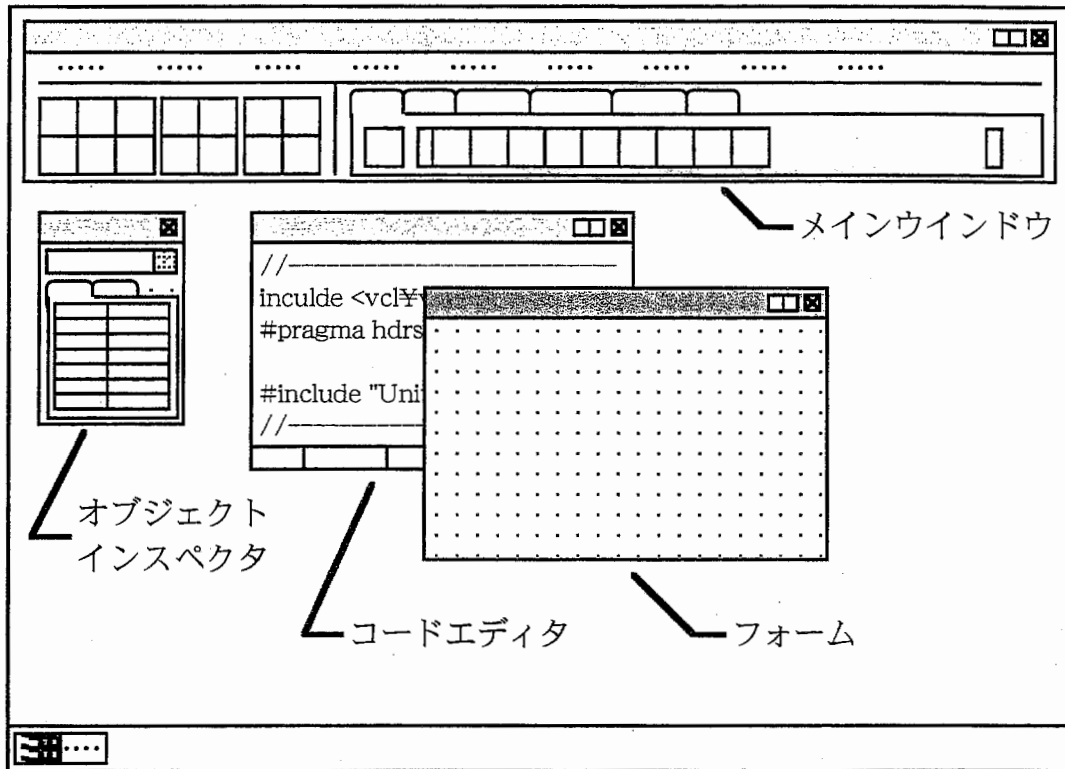


図2 起動直後の画面

○フォームとコードエディタ

フォームは、アプリケーションを実行した時にウインドウになる部分です。プログラム作成の際、このフォームの上にコンポーネントと呼ばれる部品を配置して、アプリケーションの見た目を組み立てていきます。プログラムはコードエディタを使って書きます。

○メインウインドウ

フォームの上に置く様々な部品（コンポーネントと呼ばれる）を置いているところです。ここから使用する部品を選択し、フォームの上に設置します。また、ファイルのセーブやプログラムを実行する時も、このウインドウを使います。

○オブジェクトインスペクタ

コンポーネントの初期設定をしたり、イベント（マウスがクリックされたなど）に対してどのようなプログラムを実行するかを決めるためのものです。

Borland C++ Builder では、フォームの上にコンポーネントと呼ばれる部品を置いて、その部品の動作をプログラムするといった順番でアプリケーションを作成します。

いよいよ、シリアルポートからデータを送信するプログラムを作成します。画面左側にあるオブジェクトインスペクタのイベントと書かれたタブをクリックし、OnCreateと書かれた部分の右の空白をダブルクリックします。すると、コードエディタ(プログラムが書かれているウインドウ)に切り替わり、以下のようなOnCreateのイベントハンドラが作成されます。

```
void _fastcall TForm1::FormCreate(TObject *Sender)
{
}

```

括弧の中に、次のプログラムを書き込みます。

```
DCB tDCB;
COMMTIMEOUTS TIMEOUT;

hCom = CreateFile( //ハンドルを取得
    "COM1",
    GENERIC_READ | GENERIC_WRITE,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL);

SetupComm(hCom, 1024, 1024); //バッファサイズの設定

GetCommState(hCom, &tDCB); //通信パラメータ設定
tDCB.BaudRate = CBR_9600;
tDCB.StopBits = ONESTOPBIT;
tDCB.Parity = NOPARITY;
tDCB.ByteSize = 8;
tDCB.fDtrControl = DTR_CONTROL_HANDSHAKE;
SetCommState(hCom, &tDCB);

EscapeCommFunction(hCom, SETDTR); //DTR信号をONにする

SetCommMask(hCom, 0); //イベントマスクの設定

GetCommTimeouts(hCom, &TIMEOUT); //タイムアウト値の設定
TIMEOUT.ReadIntervalTimeout = 200;
TIMEOUT.ReadTotalTimeoutMultiplier = 0;
TIMEOUT.ReadTotalTimeoutConstant = 200;
TIMEOUT.WriteTotalTimeoutMultiplier = 0;
TIMEOUT.WriteTotalTimeoutConstant = 10000;
SetCommTimeouts(hCom, &TIMEOUT);

```

制御の対象がSONYのEVI-D30かG20の場合は、続けてカメラの初期化をするプログラムを書きます。そのほかのデバイスを使う場合は、それぞれのデバイスのマニュアルを参照してください。

```
//インターフェイスとコマンドバッファのクリア
unsigned char Init[]
    = {0x88,0x30,0x01,0xFF,0x88,0x01,0x00,0x01,0xFF};
unsigned char reply[10];
WriteFile(hCom,Init,10,&BytesWritten,NULL);
ReadFile(hCom,reply,10,&BytesRead,NULL);
```

オブジェクトインスペクタのOnCloseと書かれた項目の右の空白をダブルクリックして、フォームのOnCloseイベントハンドラに、以下のプログラムを書き込みます。

```
CloseHandle(hCom); //ハンドルのクローズ
```

コードエディタの右側のスクロールバーを一番上までドラッグして、次の部分に、グローバル変数を設定します。

```
#pragma resource "*.dfm"
TForm1 *Form1;
HANDLE hCom;
unsigned long BytesWritten;
unsigned long BytesRead;
```

この行を書き込む
この行を書き込む
この行を書き込む

フォームをクリックし、メインウインドウ (図3) のStandardタブをクリックし、ボタンコンポーネント (OKと書かれたボタン) をダブルクリックします。すると、フォーム上にボタンのコンポーネントがセットされます。

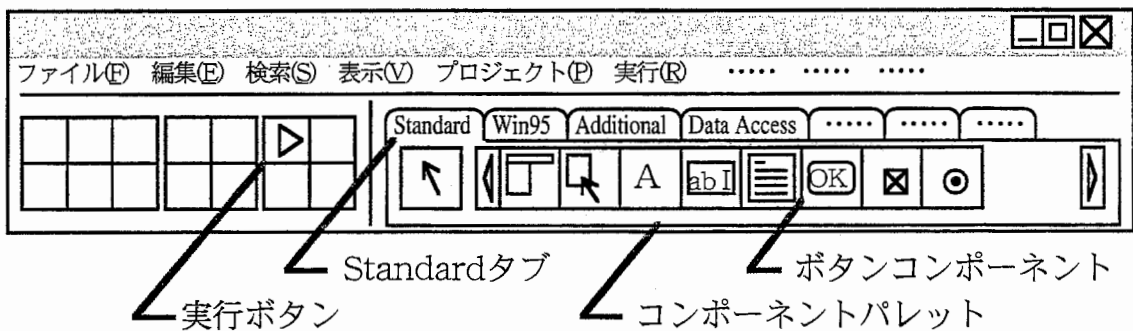


図3 メインウインドウ

フォーム上にセットされたボタンをクリックし、オブジェクトインスペクタのプロパティタブをクリックします。Captionと書かれた項目の右の空白を選択し、キーボードから” RESET” と入力します。

最後に、データをシリアルポートから出力する部分のプログラムを書きます。フォームに置いたボタンをダブルクリックすると、ボタンのOnClickイベントハンドラが生成されます。ここに、以下のプログラムを書き込みます。

```

unsigned char Reset[]
    = {0x81,0x01,0x06,0x05,0xff};    ☞ここに送信するデータを記入する
int nData = 5;                        ☞ここに送信するデータの数を記入する

//COMポートに出力
WriteFile(hCom,Reset,,nData&BytesWritten,NULL);

```

完成したプログラムを実行してみます。メインウィンドウの左側にある実行ボタン(▶)をクリックするか、F9ボタンを押します。カメラが首を振れば完成です。

フォームにボタンを追加して、それぞれのボタンに別のコマンドを出力するようにプログラムを記述すれば、様々なコマンドを送信することが出来ます。下のコマンドリストを参考に、パンティルタリモコンを作ってみましょう。

表2 パンティルタのコマンドリスト

Drive命令	Up	8X 01 06 01 VV WW 03 01 FF	X : カメラ番号
	Down	8X 01 06 01 VV WW 03 02 FF	
	Left	8X 01 06 01 VV WW 01 03 FF	VV : PanSpeed (D30 → 0x01~0x18) (G20 → 0x01~0x20)
	Right	8X 01 06 01 VV WW 02 03 FF	
	Stop	8X 01 06 01 VV WW 03 03 FF	
	Home	8X 01 06 04 FF	WW : TiltSpeed
	Reset	8X 01 06 05 FF	
電源	Power On	8X 01 04 00 02 FF	(D30 → 0x01~0x14)
	Power Off	8X 01 04 00 03 FF	(G20 → 0x01~0x20)
Zoom命令	Stop	8X 01 04 07 00 FF	Address Set ネットワークを初期化
	Tele	8X 01 04 07 02 FF	
	Wide	8X 01 04 07 03 FF	
初期化	Address Set	88 30 01 FF	IF-Clear
	IF-Clear	8X 01 00 01 FF	コマンドバッファのクリア

Drive命令は、コマンドを送信すると、Stopコマンドを受け取るか、機械的に動けなくなるまで止まりません。そこでOnMouseDownイベントハンドラにDrive命令を、OnMouseUpイベントハンドラにStop命令を書き込みます。

Xの部分には、カメラ番号を記入します。通常は1を記入し、81 ... となります。カメラを2台接続し、2台目のカメラに送信する場合は、82 ... となります。

例1) カメラ1をスピード5で左に向ける時のコマンド

```
unsigned char Left = {0x81,0x01,0x06,0x01,0x05,0x05,0x03,0x01};
```

例2) カメラ2をスピード6で右に向ける時のコマンド

```
unsigned char Left = {0x81,0x01,0x06,0x01,0x06,0x06,0x03,0x01};
```

もっと詳しいコマンドリストは、

<http://www.sony.co.jp/soj/BizPartners/ISP/dtp/pdf/#commandList>
からダウンロードできます。

ここで使用しているパナティルタは、カメラにコマンドを送信すると、ACKかエラーメッセージを送り返してきます。データを受信するためには、イベントマスクの設定や受信まで待機するプログラムを書く必要があります。ここでは、受信データが一杯にならない様に、以下のプログラムを送信する部分のWriteFile関数の後ろに書き込みます。

```
unsigned char reply[100];  
unsigned long BytesRead;  
  
//COMポートから入力  
ReadFile(hCom,reply,5,&BytesRead,NULL);
```

カメラが動かない場合

コンパイル時にエラーメッセージがでたら、もう一度プログラムを見直してみます。エラーメッセージをダブルクリックすると、エラーのある行にカーソルが移動します。次の点に注意して、エラーを取り除きます。

- 1 関数および変数の間違いがないか
- 2 文末のセミコロン(;)を忘れていないか

コンパイルに成功したが、カメラが動かない場合は、以下の点をチェックします。

- 1 ACアダプタがきちんと接続されているか
- 2 電源スイッチがONになっているか
- 3 VISCAケーブルがきちんと接続されているか
- 4 VISCAケーブルの配線が間違っていないか

図4を見て、テスタを使って接続を確かめてください

- 5 送信するコマンドが間違っていないか
- 6 COMポートが使用可能な状態か

BIOSの設定が必要になってきます

それでも動かない場合は、販売店に相談してください。



VISCAケーブルを作る

VISCAケーブルが手に入らない場合は、図4を参考にして作ります。パンティルタ側のケーブルは、Macintoshのシリアルケーブルを流用することができます。

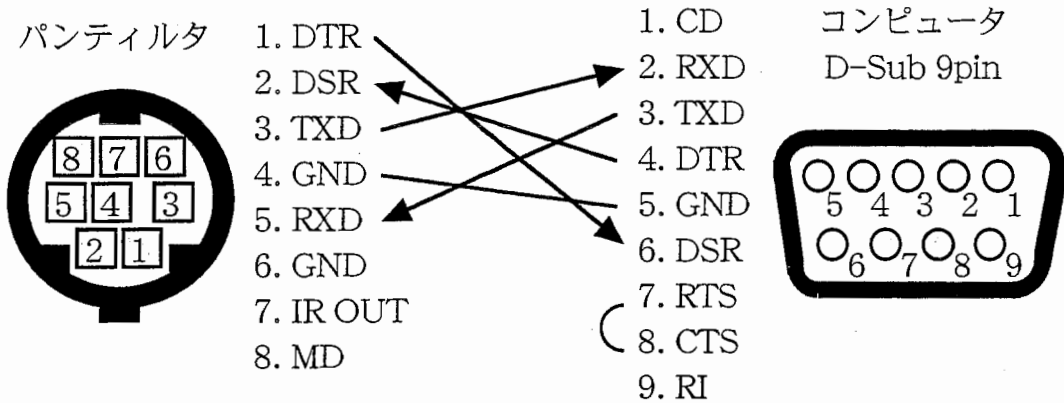


図4 VISCAケーブルのコネクタ背面と接続



RS-422規格のシリアルポート

通常、PC/AT互換機のシリアルポートには、RS-232Cの規格が採用されています。しかし、中にはMacintoshのようにシリアルポートにRS-422を採用しているものも存在します。また、制御したい機器がRS-422規格のものとしたら、そのままコンピュータのシリアルポートに接続しても動きません。このような場合、RS-232CをRS-422に変換する回路が必要になります。回路図を、このレポートの最後に示します。



サンプルプログラム

シリアルポートを使ったアプリケーションのサンプル4つの使い方を説明します。

○ パンティルタ リモコン1

パンティルタに付属しているリモートコントローラと同じです。方向ボタンを押すと、カメラが動きます。プログラムの構成は、“アプリケーションの作成”で説明したものとほとんど同じです。

- 接続 -

左カメラをCOM1に、右カメラをCOM2に接続します。

- 使い方 -

操作したいカメラの方向、ズーム、電源のボタンを押すと、カメラが動きます。

○ パンティルタ リモコン2

パンティルタリモコン1をコンパクトにしたタイプです。

- 接続 -

左カメラをCOM1に、右カメラをCOM2に接続します。

- 使い方 -

操作したいカメラのチェックボックスをチェックし、方向、ズーム、電源のボタンを押すと、チェックしたカメラを操作できます。

○ VISCA Script

2台のパンティルタを、プログラムの通りに動かすことが出来ます。D30、G20のどちらのカメラを接続しても、同じアプリケーションで動作します。

- 接続 -

左カメラをCOM1に、右カメラをCOM2に接続します。

- 使い方 -

テキストエディタでプログラムを書き、実行ボタンを押します。

関数は、以下のものが使用可能です。

表3 VISCA Script コマンドリスト

drive [LまたはR] [水平方向の角度] [垂直方向の角度]	片眼を指定した位置に動かす
driveLR [水平方向の角度] [垂直方向の角度]	両眼を指定した位置に動かす
home [LまたはR]	ホームポジションに戻る
reset [LまたはR]	カメラをリセットする
verg [距離] [横方向の角度] [縦方向の角度]	距離と角度を指定して、その点で輻輳する
wait [時間]	動作と動作の間のウェイトを設定する
speed [%]	カメラの速度を設定する

- サンプルファイル -

sokudoku.VSC 速読をしている人の目の動きです。

sokudokuj.VSC 速読をしている人の目の動きです。

car.VSC 車が通り過ぎる所を見ている人の目の動きです。

doku.VSC 速読をして、どこまで読んだか分からなくなった人の目の動きです。

○ ビデオカセットレコーダ リモコン

2台のVCRを同時に操作するリモコンです。2眼式立体ディスプレイを使う場合などのような、同時に画像を再生する必要がある場合に便利です。VCRにはSONYのEVO-9800Aを使用します。このVCRは、RS-422を使用しているため、インターフェイスコンバータが必要になります。(図5)

🔍 参考文献

[1] 塚越一雄,『すぐわかる C++ Builder』,技術評論社,1997年

[2] 柳沢正明,『DelphiによるRS-232C制御と通信プログラム』,ソフトバンク,
InsideWindows 1998年2月号

RS-232

RS-422

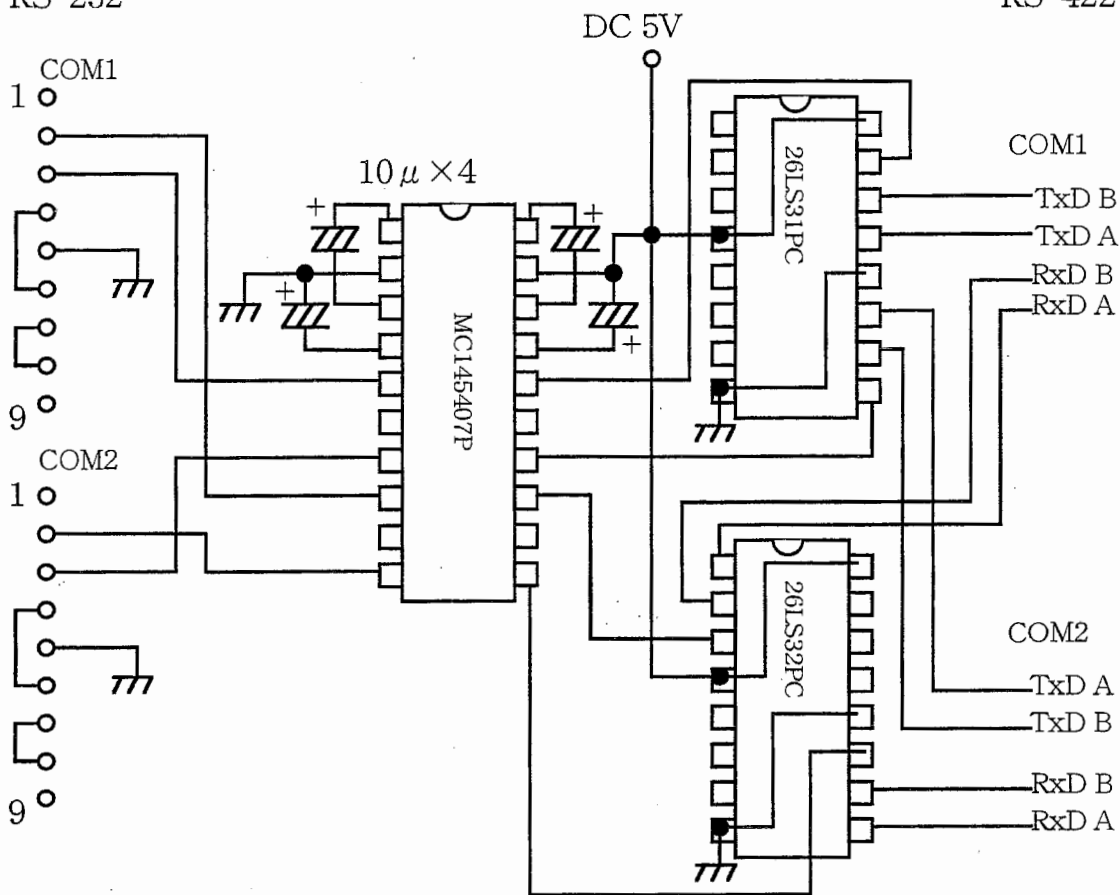


図5 RS-232C <-> RS-422インターフェースコンバータの回路図