

ServoCenter3.1 Direct Serial I/O Examples – Visual C++ 6.0

1. Overview

This document provides a simple Visual C++ 6.0 example program illustrating how to access the serial port and directly communicate with the ServoCenter 3.1 controller board to control a servo. Note that the program is provided to illustrate simple raw serial communication using the ServoCenter 3.1 protocol and does not demonstrate the full feature set of the ServoCenter 3.1 controller. Refer to section 4.1 of the ServoCenter 3.1 manual for a description of the entire ServoCenter 3.1 protocol and feature set.

2. Sample Code

```

/*****\
* This demo program illustrates how to move servo motors *
* using raw serial communication access to the *
* Yost Engineering, Inc. ServoCenter 3.1 controller board *
* using Visual C++ 6.0. *
* *
* (c) 2001-2004 Yost Engineering, Inc. *
* www.YostEngineering.com *
* *
\*****/

#include <windows.h>
#include <stdio.h>
#define PORTNUM 1
#define BAUDRATE 9600

void moveservo(HANDLE *,int,int,int,int);
int InitPort(unsigned int, unsigned int, HANDLE *, DCB *);

int main(int argc, char *argv[])
{
    DCB dcb;
    HANDLE hCom;
    int i=0, BoardNum, ServoNum, Position, Speed;

    printf(" ServoCenter 3.1 Demonstration Program \n");
    printf(" (c)2000-2004 Yost Engineering, Inc. \n");
    printf(" www.YostEngineering.com \n");
    printf(" \n");

    if((InitPort(PORTNUM,BAUDRATE,&hCom,&dcb))!=0)//open serial port
    {
        printf("\tCould not initialize Comm Port!\n");
        return (1);
    }
}

```

```

else
{
    while(1)
    {
        printf("\n  Enter Board Number (0-15):");
        scanf("%d",&BoardNum);
        printf("\n  Enter Servo Number (0-15):");
        scanf("%d",&ServoNum);
        printf("\n  Enter Position (0-200):");
        scanf("%d",&Position);
        printf("\n  Enter Speed (1-100):");
        scanf("%d",&Speed);
        printf("\n\tSending Command to Servo...\n");
        moveservo(&hCom,BoardNum,ServoNum,Position,Speed);
        printf("\n\tDone!\n");
    }
}
return (0);
}

void moveservo(HANDLE *hCom,int board,int servo,int position,int
speed)
{
    unsigned char buffer[6];          //create empty command packet
    unsigned long BytesWrittend;     //records # of bytes sent

    buffer[0]=board%16 + 0xf0;       //board id #
    buffer[1]=16;                    //move raw command
    buffer[2]=servo%16; //servo #--used to identify which servo
    buffer[3]=position%201;//position to which the servo will move
    buffer[4]=speed%101;             //speed at which servo will move
    buffer[5]='\0';                  //NULL character added to disable checksum
    WriteFile(*hCom,buffer,5,&BytesWrittend,NULL); //send packet
}

int InitPort(unsigned int PortNum, unsigned int BaudRate, HANDLE *hCom, DCB
*dcb)
{
    BOOL fSuccess;
    char pcCommPort[4]='\0';
    if(PortNum==1)
        sprintf(pcCommPort,"COM1");
    else if(PortNum==2)
        sprintf(pcCommPort,"COM2");
    else if(PortNum==3)
        sprintf(pcCommPort,"COM3");
    else if(PortNum==4)
        sprintf(pcCommPort,"COM4");
    else
        printf("\tPort Number not recognized\n");

    *hCom = CreateFile( pcCommPort,
        GENERIC_READ | GENERIC_WRITE,

```

```

        0, // must be opened with exclusive-access
        NULL, // no security attributes
        OPEN_EXISTING, // must use OPEN_EXISTING
        0, // not overlapped I/O
        NULL // hTemplate must be NULL for comm
    );

if (*hCom == INVALID_HANDLE_VALUE)
{
    // Handle the error.
    printf("\tCreateFile failed with error%d.\n",
        GetLastError());
    return (2);
}

// Build on the current configuration, and skip setting the size
// of the input and output buffers with SetupComm.

fSuccess = GetCommState(*hCom, dcb);

if (!fSuccess)
{
    // Handle the error.
    printf ("\tGetCommState failed with error %d.\n",
        GetLastError());
    return (3);
}

// Fill in DCB: Baudrate,8 data bits,no parity, 1 stop bit.
switch(BaudRate)
{
    case 9600: dcb->BaudRate = CBR_9600; break;
    case 14400: dcb->BaudRate = CBR_14400; break;
    case 19200: dcb->BaudRate = CBR_19200; break;
    case 38400: dcb->BaudRate = CBR_38400; break;
    default: printf("\tBaud Rate not recognized\n");
             printf("\tUsing default rate of 9600bps.\n");
             dcb->BaudRate = CBR_9600; break;
}

dcb->ByteSize = 8; // data size, xmit, and rcv
dcb->Parity = NOPARITY; // no parity bit
dcb->StopBits = ONESTOPBIT; // one stop bit

fSuccess = SetCommState(*hCom, dcb);

if (!fSuccess)
{
    // Handle the error.
    printf("\tSetCommState failed with error %d.\n",
        GetLastError());
    return (4);
}

```

```
    printf ("\tSerial port successfully reconfigured.\n");  
    return (0);  
}
```

3. Additional Information

This code is available in the `vc_direct.c` file in the Examples directory of the ServoCenter3.1 CD or online at www.YostEngineering.com/ServoCenter.