

B&B ELECTRONICS **TECHNICAL NOTES**



Vlinx™ Ethernet to Serial Servers

Resetting Your Vlinx Ethernet Serial Server Remotely

Overview

The question has come up a number of times recently... "Can I reset my Vlinx hardware through software?" The answer is yes. If you are writing your own application you can send out messages over the Ethernet to cause the device to perform a reset. This document will start out with a description of what goes into doing this and will close with some source code to help your project along.

Getting Started

The Vlinx Ethernet Serial Server hardware allows users to reset the device through the Ethernet side of the device. You can do this by sending a UDP message out over the network. A UDP message is a broadcast message. All Vlinx Ethernet Serial Servers on the network will receive this message.

So how does your serial server know who this message is being sent to? Each message **MUST** include the **Serial Number** and **MAC Address** of the intended recipient. The devices use this information to determine who the message is for. Lets take a look at the code that makes it happen.

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

RR-WDR

The Source Code

This application was developed in Microsoft Visual C++ 6.0. The code can be copied from this document and placed right into your application, however the device information is specific to the device that we used and will need to be modified for use with yours.

The code can be broken into 2 different sections. The first section will be the CPP file and the second is the header file. Both are very important. We'll start with the CPP file.

The CPP File

```
#include "stdafx.h"
#define PACKET_SIZE    142

//Command Struction Declaration
struct cmd_msg {
    char pni[24];
    char serialno[16];
    unsigned char mac_addr[6];
    char command[12];
};

//*****
//Main Function
//*****
int main(int argc, char* argv[])
{
    cmd_msg      Message;
    SOCKET       hSocketHandle;
    char         send_data[PACKET_SIZE];
    struct sockaddr_in servaddr;
    int          returnValue;
    const int    on = 1;
    char         *pData;
    int          sendLen, len, i;
    UCHAR        sum = 0;
    WORD         wVersionRequested;
    WSADATA      wsaData;

    //Format the whole structure
    memset(Message.pni, 0, 58);

    //Setup Commannd and Passport Networks String Information
    wsprintf(Message.command, "reset\0");
    wsprintf(Message.pni, "Pas sport Networks Inc.\0");

    //MAC Address Setup
    Message.mac_addr[0] = 0;
    Message.mac_addr[1] = 12;
    Message.mac_addr[2] = 21;
    Message.mac_addr[3] = 34;
    Message.mac_addr[4] = 43;
    Message.mac_addr[5] = 56;

    //Serial Number
    Message.serialno[0] = '0';
    Message.serialno[1] = '3';
    Message.serialno[2] = '0';
    Message.serialno[3] = '5';
    Message.serialno[4] = '0';
    Message.serialno[5] = '0';
    Message.serialno[6] = '1';
    Message.serialno[7] = '0';
    Message.serialno[8] = '3';
    Message.serialno[9] = '4';
    Message.serialno[10] = 0;
    Message.serialno[11] = 0;
```

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

```
Message.serialNo[12] = 0;
Message.serialNo[13] = 0;
Message.serialNo[14] = 0;
Message.serialNo[15] = 0;

//Start WSA
wVersionRequested = MAKEWORD( 2, 2 );
returnValue = WSASStartup( wVersionRequested, &wsaData );

//Create a socket
hSocketHandle = socket(PF_INET, SOCK_DGRAM, 0);
if(hSocketHandle == INVALID_SOCKET)
{
    returnValue = WSAGetLastError();
}

//Turn on broadcast mode for this socket.
returnValue = setsockopt(hSocketHandle, SOL_SOCKET, SO_BROADCAST, (const char *) &on, sizeof(on));
if (returnValue < 0)
{
    returnValue = WSAGetLastError();
    closesocket(hSocketHandle);
}

returnValue = GetLastError();

//Initialize servaddr to the address we're broadcasting to.
servaddr.sin_family = PF_INET; // set to TCP/UDP
servaddr.sin_port = htons(8888);
servaddr.sin_addr.s_addr = INADDR_BROADCAST; // defined in WINSOCK.h

returnValue = connect(hSocketHandle, (SOCKADDR*)&servaddr, sizeof(servaddr));

//Broadcast a packet to the server.
memset(send_data, 0, 59);
sendLen = 0;

pData = (char *)&Message;
len = sizeof(struct cmd_msg)-1;

for (i=0; i<len; i++)
{
    send_data[sendLen+i] = *(pData++);
}

sendLen = len + 1;

returnValue = GetLastError();
returnValue = sendto(hSocketHandle, send_data, sendLen + 1, 0, (struct sockaddr *) &servaddr, sizeof(servaddr));

if (returnValue < 0)
{
    returnValue = GetLastError();
    closesocket(hSocketHandle);
}

closesocket(hSocketHandle);
return 0;
}
```

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

The Header File

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#ifndef AFX_STDAFX_H_6677A2AB_BF2A_4646_8E79_DE5DCEFBF31E_INCLUDED_
#define AFX_STDAFX_H_6677A2AB_BF2A_4646_8E79_DE5DCEFBF31E_INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#define WIN32_LEAN_AND_MEAN           // Exclude rarely-used stuff from Windows headers

#include <stdlib.h>
#include <stdio.h>
#include <windows.h>
#include <winsock2.h>

// TODO: reference additional headers your program requires here

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_STDAFX_H_6677A2AB_BF2A_4646_8E79_DE5DCEFBF31E_INCLUDED_)
```



Linking Errors

Remember that you will need to add **ws2_32.lib** to your project before you can use the sockets functions. This can be done from the **Project Settings** dialog. Under the **Link tab** you will find an **Object/library modules**: field. Add the file to this list. Front or back of the list doesn't matter.

Right at the top of the CPP source code you can see that we are setting up the serial number and MAC address of the device (Message.mac_addr and Message.serialno). This portion is crucial to the success of the reset. This information **MUST** be changed to match the information for your device. This information can be located on the underside of the device. An example of this information would be:

Serial Number: 0305001034

MAC Address: 00:0B:11:04:0A

Note that the colons within the MAC address are ignored and that the number values are in hexadecimal. However in the sample code these values are placed in the byte array as decimal numbers.

Summary

Using this code you should be able to reset your ESP hardware. Questions welcome.

Jeremy Severson

B&B Electronics - 707 Dayton Road - Ottawa IL 61350

Contact me: JSeverson@bb-elec.com

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

RR-WDR

