DelcomX USB HID ActiveX Control Datasheet

890682 – DelcomX USB HID ActiveX Control

Table of Contents

1 Description

The DelcomX is an ActiveX control that can run in any ActiveX enabled container. Such as Internet Explore and other programs that support ActiveX (e.g. MSWord, Excel, PowerPoint). The DelcomX control enables simply communication between your program and the Delcom USB device. The control works with all Delcom G2 USB HID products (PartNumbers 9xxxxx).

2 Security

ActiveX objects can access your computer files; therefore there is a security risk in running ActiveX controls. Only install signed ActiveX controls from a source your can trust.

3 Requirements

To run the DelcomX object you will need a container (program) that supports the ActiveX interface, the DelcomX OCX or CAB file and one of the Delcom USB HID G2 Devices.

Most programming languages can communicate to an ActiveX object, but the most common are: VBScript, JavaScript and C++.

The DelcomX control can be downloaded from our website, see link below.
http://www.delcomproducts.com/downloads/DelcomX.cab

4 Class Identifier

Each ActiveX object has a unique identifier. This identifier is called the Class Identifier (CLSID) and is use to find and access the Active X control on your system. The class identifier for the DelcomX control is listed below.

```
CLASSID="CLSID:e68b4792-0345-4cab-a503-946a53e29d60"
```

5 Install

There are two ways to install the DelcomX control. The first is just to include a link in your source code to the DelcomX.cab file on our website. And the second is manually copy the OCX file in to your working directory and then call in local.

5.1 Automatic Install

The automatic install is the recommend install usage. Uses this method is simply and ensure you get the most up to date version of the control. To use this method simply reference the fully URL address. http://delcomproducts.com/downloads/delcomx.cab

The first time your code runs it will automatic go to the Delcom website and download the CAB file. Windows will popup up a warning about running the ActiveX control. Check accept to download the control. Once downloaded Windows will automatically load and register the control with your system. Once the DelcomX is loaded on to your system, subsequent calls the DelcomX control will run automatically from the registry copy on your system. Note the one time download may take longer if your system does not have the required Windows support files.

IE example of the automatic install link.

```
<OBJECT ID="myDelcomX"
CLASSID="CLSID:e68b4792-0345-4cab-a503-946a53e29d60"
CODEBASE="http://delcomproducts.com/downloads/delcomx.cab">
</OBJECT>
```

5.2 Manual Install

One can also copy the DelcomX.OCX file from the Delcom website and include it in the working directory of the source code there working on. Then in your source code just make the reference to the local DelcomX.OCX file. Note when run an ActiveX control in IE in local host mode, you must enable Local Host ActiveX option.

1 - Select Tools->Internet Options from the IE menu.

2 - Click the ‘Advanced’ tab.

3 - Check the 2nd option under ‘Security’ in the tree (Allow active content to run in files on my computer.)

IE example of the local link.

```
<OBJECT ID="myDelcomX"
CLASSID="CLSID:e68b4792-0345-4cab-a503-946a53e29d60"
CODEBASE="delcomx.ocx">
</OBJECT>
```

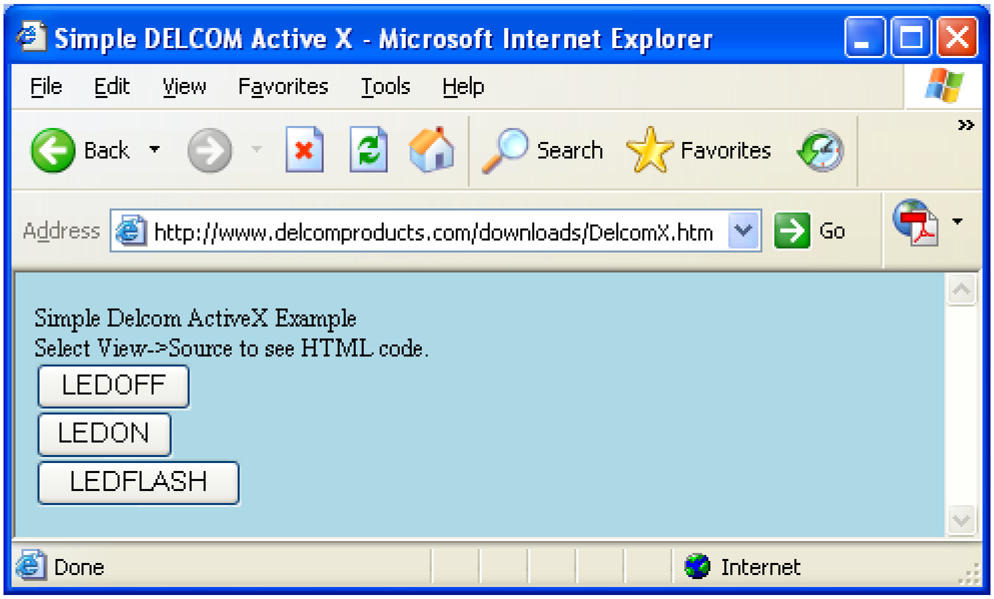
6 Quick Start

Below is a quick simply example on how to use the DelcomX ActiveX object in IE. The code displays three buttons that will change the state of the GPIO on the Delcom USB device. The GPIO pin that will change is Port1.0, if the device is a USB visual indicator then LED connected to this pin will either turn on, off or toggle every 0.5 seconds (flash the LED).

To start cut and paste the following code into a file and save it has 'test.htm' and then double click on the file. The file should open in Internet Explorer, if not right click on the file and select 'open with->Internet Explorer'. If the DelcomX object is not already loaded on your computer, IE will automatically start to download it. IE will popup a warning about the security risk, just click continue to load the DelcomX object. Once the DelcomX object is loaded you can click one of the buttons and if a Delcom USB device connected, then the P1.0 pin will toggle is state. If a Delcom USB Visual Indicator is attached then the LED will turn on, off, or flash.

```
<html>
<HEAD>
<TITLE>Simple DELCOM Active X</TITLE>

<OBJECT ID="myDelcomX"
CLASSID="CLSID:e68b4792-0345-4cab-a503-946a53e29d60"
CODEBASE="http://delcomproducts.com/downloads/delcomx.cab">
</OBJECT>

<script type="text/javascript" FOR="window">

function LED( Color, Mode)    //Color:0,1,2; Mode:0=Off, 1=On, 2=Flash
   {
   try
          {
          myDelcomX.LEDCtrl(0, 0, Color, Mode);
          }

   catch(err)
          {
          txt ="Error number: " + err.number + "\n";
          txt+="Error name: " + err.name + "\n\n";
          txt+="Error description: " + err.description + "\n\n";
          txt+="Click OK to continue.\n\n";
          alert(txt);
          }
   }

</script>
</HEAD>

<body bgcolor=lightblue LANGUAGE = JavaScript>
Simple Delcom ActiveX Example<br/>
<INPUT TYPE=BUTTON VALUE=LEDOFF onclick="LED(0,0);" ><br/>
<INPUT TYPE=BUTTON VALUE=LEDON onclick="LED(0,1);" ><br/>
<INPUT TYPE=BUTTON VALUE=LEDFLASH onclick="LED(0,3);" ><br/>
</body>
</html>
```

The IE screen capture.



http://www.delcomproducts.com/downloads/DelcomX.htm

Simple Delcom ActiveX Example
Select View->Source to see HTML code.

7 Conventions

Each program language has it's own data type names. So to simplify the data type in the calling methods we will use the following conventions.

U8	Unsigned 1 Byte Integer (Byte)
U16	Unsigned 2 Byte Integer (Short)
U32	Unsigned 4 Byte Integer (Long)
S8	Signed 1 Byte Integer
S16	Signed 2 Byte Integer
S32	Signed 4 Byte Integer
*	Data types that have an asterisk prefix are passed by reference (pointers).

8 Methods

The DelcomX control supports the following Methods. Methods are functions that the user can call to do a certain action. When a function fails an error is throw. Throw errors can be captured with the TRY and CATCH controls.

All methods have the same first two parameters. There are TID (Type Identifier) and SID (Serial Number Identifier). When the method is run the function will only send the command to a matching TID and/or SID device. In most cause you can set both these variables to zero. This will let the function to find any Delcom USB device. If the TID is non zero then it will only find a Delcom USB device with the matching Family Type. If SID is non zero the function will only find a Delcom USB device that has the matching Serial Number. Use TID or SID when you have more than one Delcom USB device connected to your system.

TID – Family Types:
0=Any device – Searches for all Delcom USB devices
1=USB HID Chips
2=USB Visual Indicators
3=USB Numeric Displays
4=USB Traffic Lights
5=USB Buzzer

8.1 Basic Methods

The Basic Methods control basic GPIO functions, such as setting a pin high or low. Basic functions are supported on all Delcom USB device.

IsDevicePresent(U16 TID, U16 SID)

This does nothing if the USB device is present. If the USB device is not present an error is thrown. Use the try and catch statements to detect if the device is present. See Error Handling section below.

Example

```
try     {
        myDelcomX.IsDevicePresent(0, 0);
        }
catch(err) {
        // Device not present
        }
```

WritePorts(U16 TID, U16 SID, U8 Port0, U8 Port1)

Writes Port0 and Port1 values to the USB chips ports.

Example

```
myDelcomX.WritePorts(0, 0, 255, 0);
```

WritePort0(U16 TID, U16 SID, U8 Port0)

Writes Port0 value to the USB chips port0.

WritePort1(U16 TID, U16 SID, U8 Port1)

Writes Port1 value to the USB chips port1.

U32 ReadPorts(U16 TID, U16 SID)

Reads the values on Ports P0 and P1 and returns them. Port0 value is returned in byte 0 and Port1 value is returned in byte 1. On error the returned value is -1.

Example

```
Result = myDelcomX.ReadsPorts(0, 0);
Alert(“Port0=” + Results & 0xFF);
Alert(“Port1=” + (Results & 0xFF00)>>8 );
```

U32 ReadPort0(U16 TID, U16 SID)

Reads the values on Ports P0 and returns them. Port0 value is returned in byte 0. On error the returned value is -1.

U32 ReadPort1(U16 TID, U16 SID)

Reads the values on Ports P1 and returns them. Port1 value is returned in byte 0. On error the returned value is -1.

8.2 Advance Methods

Advance methods allow the user to low level commands to the Delcom USB device. See the USBIOHID manual of commands descriptions.
http://delcomproducts.com/downloads/USBIOHID.pdf

WritePacket(U16 TID, U16 SID, U8 Major, U8 Minor, U8 DataLSB, U8 DataMSB, u32 DataHID)

Sends a USB HID data command to the USB device.

WritePacketExt(U16 TID, U16 SID, U8 Major, U8 Minor, U8 DataLSB, U8 DataMSB, u32 DataHID, u32DataExtHigh, u32 DataExtLow)

Sends a USB HID data command with data extension parameter to the USB device. See the USBIOHID manual of commands.

S32 ReadPacket4(U16 TID, U16 SID, U8 Command, *u32 DataHigh, *u32 DataLow)

Sends a USB HID read command and returns the first 4 bytes in the function return value.

Example – Reads ports using the advanced read functions

```
Result = myDelcomX.ReadsPorts(0, 0, 100);
Alert(“Port0=” + Results & 0xFF);
Alert(“Port1=” + (Results & 0xFF00)>>8 );
```

ReadPacket(U16 TID, U16 SID, U8 Command, *u32 DataHigh, *u32 DataLow)

Sends a USB HID read command and returns the read data in parameters DataHigh and DataLow. Note DataHigh and DataLow passed by reference.

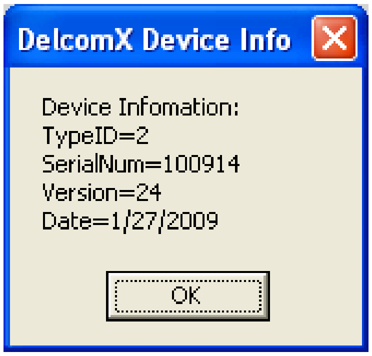
ReadPacketExt(U16 TID, U16 SID, U8 Command, *u32 DataHigh, *u32 DataLow, *u32 DataExtHigh, *u32 DataExtLow,)

Sends a USB HID read command and returns the read data in parameters DataHigh and DataLow. Note DataHigh and DataLow passed by reference.

8.3 Information Methods

DisplayDeviceInfo(U16 TID, U16 SID)

This method reads the USB device information and displays it in a popup dialog box. This method will display the, TID, SID and firmware version and date of the USB device.



Device Infomation:
TypeID=2
SerialNum=100914
Version=24
Date=1/27/2009

U32 GetTypeID(U16 TID, U16 SID)

This method reads the USB device TID (TypeID). See section 8.

U32 GetVersion(U16 TID, U16 SID)

This method reads the USB device firmware.

U32 GetSerialNum(U16 TID, U16 SID)

This method reads the USB device Serial Number.

U32 GetDay(U16 TID, U16 SID)

This method reads the USB device firmware day.

U32 GetMonth(U16 TID, U16 SID)

This method reads the USB device firmware month.

U32 GetYear(U16 TID, U16 SID)

This method reads the USB device firmware year.

8.4 Visual Indicator Methods

The following methods control the LEDs on the Delcom USB Visual Indicators. See the USB HID IO manual for more information on these commands.

LEDCtrl(U16 TID, U16 SID, U8 Color, U8 Mode)

This method turns the LED on, off or to flash mode. The Color parameter specifies the LED bank. The Color value can be 0, 1 or 2. The color depends on the light type, for example part number 904000 has the following color mapping; 0=green, 1=red and 2=blue. The Mode parameter controls the LED mode: 0=off, 1=on and 2=flash.

LEDPower(U16 TID, U16 SID, U8 Color, U8 Power)

This method set the LED power level. Valid range is from 0 to 100. Default value is 80.

LEDPreScalar(U16 TID, U16 SID, U8 PreScaler)

This method set the LED global pre-scalar value. The lower the value the faster the flash rate.

LEDDutyCylce(U16 TID, U16 SID, U8 Color, U8 DutyOn, U8 DutyOff)

This method set the LED duty cycle in flash mode. Valid parameter range is 0 to 255. default is 100 for both DutyOn and DutyOff.

LEDOffset(U16 TID, U16 SID, U8 Color, U8 Offset)

This method set the LED offset. Valid range is from 0-255. Default value is 0. Increasing this value will delay the start off the flash cycle when the LEDSync function is called.

LEDSync(U16 TID, U16 SID)

Synchronizes all the LEDs flash rates. Also see LEDOffset.

9 Events

Currently the DelcomX control does not support any events.

10 Error Handling

If a method fails an error will be thrown. To catch the error your code must use the TRY and CATCH statements. Below is a list of custom DelcomX errors. Errors below 10000 are system errors. Also see system error information.

Error Number	**Description**
10000	USB Device not found.
10001	Parameter invalid or out of range.

11 Software Release Notes

Version 1.0.0.1 - Dec 10 2009 - Initial Release

12 Examples

12.1 VBScript Example

The following simple example will write and then read the USB devices GPIO port pins each time the page is refreshed.

```
</BODY>
</HTML>
<html>
<HEAD>
<TITLE>Simple DELCOM Active X – VBSript read ports</TITLE>

<OBJECT ID="myDelcomX"
CLASSID="CLSID:e68b4792-0345-4cab-a503-946a53e29d60"
CODEBASE="http://delcomproducts.com/downloads/delcomx.cab">
</OBJECT>

<SCRIPT Language="VBScript">
<!--

Function vbsReadPorts()
   dim result, port0, port1
   result=myDelcomx.ReadPorts(0,0)
   port0 = result AND &HFF
   port1 = (result AND &HFF00) /&H100
   MsgBox "Port0="& port0 &” Port1-“& port0
   vbsReadPorts=result
End Function

Function vbsWritePorts(Port0, Port1)
   dim result
   result=myDelcomx.WritePorts(0,0, Port0, Port1)
   vbsWritePorts=result
End Function

-->
</SCRIPT>
</HEAD>

<body >
VBSrcipt Example<br/>

<SCRIPT Language="VBScript">
<!--

vbsWritePorts(255,0)
vbsReadPorts()

-->
</SCRIPT>


</body>
</html>
```

12.2 JavaScript Example

```
<html>
<HEAD>
<TITLE>Simple DELCOM Active X</TITLE>

<OBJECT ID="myDelcomX"
CLASSID="CLSID:e68b4792-0345-4cab-a503-946a53e29d60"
CODEBASE="http://delcomproducts.com/downloads/delcomx.cab">
</OBJECT>

<script type="text/javascript" FOR="window">

function LED( Color, Mode)    //Color:0,1,2; Mode:0=Off, 1=On, 2=Flash
   {
   try
          {
          myDelcomX.LEDCtrl(0, 0, Color, Mode);
          }

   catch(err)
          {
          txt ="Error number: " + err.number + "\n";
          txt+="Error name: " + err.name + "\n\n";
          txt+="Error description: " + err.description + "\n\n";
          txt+="Click OK to continue.\n\n";
          alert(txt);
          }
   }

</script>
</HEAD>

<body bgcolor=lightblue LANGUAGE = JavaScript>
Simple Delcom ActiveX Example<br/>
<INPUT TYPE=BUTTON VALUE=LEDOFF onclick="LED(0,0);" ><br/>
<INPUT TYPE=BUTTON VALUE=LEDON onclick="LED(0,1);" ><br/>
<INPUT TYPE=BUTTON VALUE=LEDFLASH onclick="LED(0,3);" ><br/>
</body>
</html>
```

13 References

13.1 Documentation

Delcom USB HID Documentation and Examples

http://www.delcomproducts.com/productdetails.asp?productnum=900000

Delcom USB HID Chips

http://www.delcomproducts.com/products_USBIO.asp

Appendix A. Document Revision History

Rev	Date	Author	Description
0	11/10/2009	DL	Initial Release

Appendix B. Notices

DELCOM PRODUCTS INC. takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on DELCOM PRODUCTS INC procedures with respect to rights in DELCOM PRODUCTS INC. specifications can be found at the DELCOM PRODUCTS INC. website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers' or users of this specification, can be obtained from the DELCOM PRODUCTS INC. Executive Director.

DELCOM PRODUCTS INC. invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the DELCOM PRODUCTS INC. Executive Director.

Mention of third party products, companies, websites and links by Delcom is for informational purposes only and constitutes neither an endorsement nor a recommendation. Delcom assumes no responsibility with regard to the selection, performance or use of these products or vendors. Delcom makes no representations regarding the quality, safety or suitability of any third party content. Delcom provides this only as a convenience for our users.

This document and the information contained herein is provided on an "AS IS" basis and DELCOM PRODUCTS INC. DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

LIFE SUPPORT POLICY - Delcom Products are not authorized for use in life support devices and/or systems without the express written approval of Delcom.