

```

/*
* 23-June-2008
* Version 2
*
* www.tctec.net
*
* Top16 C# example
*
* Console applicaion.
* Uses top16.dll and FTD2XX.dll, these dlls must be in the execution directory (release and debug).
* Alternatively, they can be located in the system32 folder
*
*
*/
using System;
using System.Collections.Generic;
using System.Collections;
using System.Text;
using System.Runtime.InteropServices;
// Import functions from Top16 DLL
//
class top16
{
    [DllImport("top16.dll")]
    public static unsafe extern Int32 dllversion();
    [DllImport("top16.dll")]
    public static unsafe extern Int32 ListTop16Boards(byte* /*[n][20]*/ boardNames);
    [DllImport("top16.dll")]
    public static unsafe extern Int32 SetOutputs(UInt32 Handle, Byte mask, Byte set);
    [DllImport("top16.dll")]
    public static unsafe extern Int32 GetInputs(UInt32 Handle);
    [DllImport("top16.dll")]
    public static unsafe extern Int32 readAnalogInput(UInt32 Handle, Int16 input, byte gain);
    [DllImport("top16.dll")]
    public static unsafe extern UInt32 OpenBoard(StringBuilder boardName);
    [DllImport("top16.dll")]
    public static unsafe extern Int32 CloseBoard(UInt32 Handle);
}
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            const int numberOfBoards = 124;
            const int nameLength = 20;

```

```

IntPtr memBlock;
String bigString;
byte[] byteArray = new byte[numberOfBoards * nameLength];
Int32 boardsFound = 0;
ArrayList boardStrings = new ArrayList();
// print the Dll version
//
String outstring = "DLL version = " + top16.dllversion() + "\n";
Console.WriteLine(outstring);

unsafe
{
    memBlock = Marshal.AllocCoTaskMem(numberOfBoards * nameLength); // get a block of memory

    // the following passes a memory block to the ListTop16Boards() function
    // which expects a 2 dimensional array of characters (][[20])
    // This block of bytes is then converted into strings of length 20
    //
    boardsFound = top16.ListTop16Boards((byte*)memBlock.ToPointer());
    System.Text.ASCIIEncoding enc = new System.Text.ASCIIEncoding();
    // copy memory block into a 1 dimensional array of bytes
    //
    Marshal.Copy(memBlock, byteArray, 0, byteArray.Length);

    // free fixed block
    //
    Marshal.FreeCoTaskMem(memBlock);

    // convert array of bytes into a single long String
    //
    bigString = enc.GetString(byteArray);
}
for (int i = 0; i < boardsFound; i++)
{
    // List the names of all Top16 boards found, and add them to an ArrayList
    // By chopping up the big string into small strings of length 20
    //
    boardStrings.Add(bigString.Substring(i * nameLength, nameLength));
    outstring = "Found : " + boardStrings[i] + "\n";
    Console.Write(outstring);
}
while (0 == 0) // loop forever
{
    // for each board, open it, set digital outputs, read some analog inputs
    // ( no exception handling, or return value checking (!))
    //
    for (int i = 0; i < boardStrings.Count; i++)

```

