

DLL for the Conversion of ADPCM data to PCM

The software DLL described herein is provided under non-disclosure agreement (NDA) between OKI and its Customers. This document consists of four (4) pages.

The MSM9841.DLL (hereafter referred to as DLL) is a 32-bit Direct Link Library for Windows 95 which converts a stream of raw 4-bit ADPCM data, such as produced by OKI speech LSIs, to 12-bit PCM data. The filename is "**msm9841.dll**". It is designed to be tied into common programming languages such as C++, Visual Basic or Delphi Pascal and can be used to generate for instance Windows WAV sound files.

An example of how-to-use is explained hereafter using the DLL with Visual Basic 4.0 while it is assumed that the user is familiar with this programming language.

To embed the DLL, copy it to Windows\System or to the application directory and enter in a VB module's general declarations section:

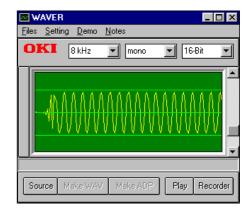
Declare Function msm9841 Lib "msm9841.dll" (ByVal code As Integer) As Integer

The DLL receives an Integer, which is an ADPCM nibble value($0\sim15$) and returns a 2's compliment PCM integer value (-2048 to +2047, full dynamic range) as conversion result. Before an ADPCM data stream can be converted, the algorithm contained in the DLL must be reset:

i = msm9841(-1)

Here is an example for a test program which plots a sine waveform from a few ADPCM sine data on a picture control using a custom scale. The example routine may serve as a simple DLL tester:

```
Private Sub Test_Click ()
  Dim adpcm As String
  Dim a As String
  Dim i As Integer
  Dim pcmdata As Integer
  adpcm = "00007F7F7F4C00409C81409C81409C81
           409C81409C81409C81409C81409C81
           409C81409C81409C81409C81409C81
           409C81409C81409C81409C81409C81"
  Picture1.Scale (0, 2048)-(Len(Adpcm), -2048)
  Picture1.Cls
  Picture1.PSet (0, 0)
  pcmdata = msm9841(-1)
  For i = 1 To Len(adpcm)
    a = "&H" + Mid$(adpcm, i, 1)
    pcmdata = msm9841(CInt(a))
    Picture1.Line -(i, pcmdata)
  Next
End Sub
```



No doubt, some little coding and form design is required to give this an appearance, however, at the end of the day, the plot result of this routine should look like the screen capture.

Since ADPCM data, for its waveform description, relies on estimated values derived from previously computed values, ADPCM nibbles do not convert as single values. Make sure to reset the DLL before you convert an ADPCM data stream.

For generation of a Windows WAV file, a Type variable is required in the same section to write the typical 44-Byte header:

```
Type WAVHeader
Riff As String * 4
Chunk As Long
WaveFmt As String * 8
FmtChunk As Long
Tag As Integer
NoCh As Integer
Samp As Long
AvgB As Long
nBlock As Integer
Bits As Integer
Data As String * 4
Size As Long
End Type
Public WAV As WAVHeader
```

This example shows how to implement conversion of an ADPCM data file to a Windows 16bit, mono, 8kHz WAV sound file. It provides that each ADPCM nibble is stored in a byte.

```
Private Sub Convert_Click (Source As String, Target As String)
  Dim i As Integer
  Dim pcmdata As Integer
  Dim a As Byte, b As String
  Open Source For Binary As 1
  Filesize = LOF(1)
  Open Target For Binary As 2
    WAV.Riff = "RIFF"
    WAV.Chunk = Filesize * 2 + 36
    WAV.WaveFmt = "WAVEfmt "
    WAV.FmtChunk = 16
    WAV.Tag = 1
    WAV.NoCh = 1
                                              'Mono (1) or stereo (2)
    WAV.Samp = 8000
                                              'Sampling frequency in Hz
    WAV.AvgB = 16000
                                              'Sampling frequency * 2
    WAV.nBlock = 2
    WAV.Bits = 16
                                              '16-Bit PCM data
    WAV.Data = "data"
    WAV.Size = Filesize * 2
    Put #2, , WAV
                                              'Write WAV header
    pcmdata = msm9841(-1)
                                              'Reset DLL
    For i = 1 To Filesize
      Get #1, , a
      pcmdata = (2048 + msm9841(CInt(a))) * 8 'Convert
      b = Chr$(pcmdata Mod 256) + Chr$(Int(pcmdata / 256))
      Put #2, , b
    Next
  Close
End Sub
```

```
Syntax:
          short __stdcall msm9841(short code)
Return:
          short type
          decoded value (-2048 to 2047)
Argument:
                        short type
          code
                        ADPCM code( 0 to 15)
                        but in case of -1, function is initialized
Example:
        #include <stdio.h>
        #include <stdlib.h>
        #include <string.h>
        FILE
               *fopen(),*fp1,*fp2;
        short ___stdcall msm9841();
        void main()
        {
                short synth,code;
                char adpcm;
                if( (fpl=fopen("adpcm_file","rb")) == NULL) {
                       printf("adpcm file open error\n");
                       exit(1);
                }
                if( (fp2=fopen("decoded_file","wb")) == NULL) {
                       printf("decoded file open error\n");
                        exit(1);
                }
                msm9841(-1);
                                                                //initialize
                while(fread(&adpcm,1,1,fp1)!=0){
                                                               // ADPCM data read
                       code = adpcm;
                                                               // ADPCM decode
                       synth = msm9841(code);
                                                               // file write
                       fwrite(&synth,4,1,fp2);
                }
        }
        fclose(fp1);
        fclose(fp2);
```

Next, please refer to another programming example, if you are using C/C++.

NOTICE: Software described herein is copyright 1997-98 Oki Electric Industry Co., Ltd. and distributed on an 'as-is' basis. The author specifically disclaims all warranties, expressed or implied, with respect to software and documentation defects, or suitability of the software for a particular purpose. It is prohibited to make and distribute exact duplicates to other individuals or organisations, and to decompile, alter its contents, remove or add data. This software is not an integral part of a sales contract. All brand, company and product names are the trademarks or registered trademarks of their respective owners.



OKI Electric Industry Co., Ltd., Device Buysiness Group, Marketing Communications

10-3 Shibaura, 4-chome, Minato-ku, Tokyo 108, Japan, Tel: +81-3-5445-6027, Fax: +81-3-5445-6058, e-mail: brenner595@oki.co.jp