



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 16-bit, 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
```

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

Syntax:

```
short __stdcall msm9841(short code)
```

Return:

```
short type  
decoded value (-2048 to 2047)
```

Argument:

```
code          short type  
              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( (fp1=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;
        synth = msm9841(code);                  // ADPCM decode
        fwrite(&synth, 4, 1, fp2);             // file write
    }

    fclose(fp1);
    fclose(fp2);
}
```

**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

**People to People Technology**

**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