VS1003B WMA REWIND PATCH

VSMPG "VLSI Solution Audio Decoder"

Project Code:
Project Name: VSMPG

Revision History			
Rev.	**Date**	**Author**	**Description**
1.0	2008-04-17	PO	Added automatic fix for Explicit Packet Length < Minimum Data Packet Length
0.9	2006-05-12	PO	Added support for other Media Object Number Length Types
0.8	2005-11-16	PO	Separate version for broadcast-only patch
0.7	2005-11-02	PO	Broadcast support
0.6	2005-04-22	PO	Still more robust resync
0.5	2005-04-18	PO	Initial version

1 VS1003B WMA Rewind

VS1003B supports WMA v2-v9: 5kbps - 320kbps files. Because of the file-format nature of WMA and data-stream nature of VS1003B, random-access for rewind and fast-format features must be done in the controller. The VS1003B firmware allows you to delete and insert ASF packets, but it is moderately hard to determine the ASF packet boundaries on a low-MHz CPU.

This WMA Rewind Patch takes over the decoding of WMA files and provides an easier resynchronization mechanism and slightly more robust decoding to allow easier implementation of rewind and fast-forward operations. The 0.7 version of the patch also supports decoding forever for internet-radio applications.

The 1.0 version also handles the cases when Explicit Packet Length < Minimum Data Packet Length by setting the padding variable automatically to the right value.

Later chapters also describe how to generate a WMA header that VS1003B understands. The required ASF objects and their required contents are listed.

The patch does not overlap with previous VS1003B patches so you can use both the “VS1003B Spectrum Analyzer” and “MPEG2.5 Layer-3 8kHz Stereo Patch” at the same time if you like.

Select the right C file for your setup. When you upload the code, it will be activated automatically. A hardware or software reset de-activates the patch. When the patch is active, SCI_HDAT1 contains ’Wm’ (0x576d) instead of ’WM’ (0x574d) when WMA file is being decoded.

Chip	**File**	**IRAM Location**	**Features**
VS1003B	wmarew2.c	0x2f0 .. 0x4ff	Rewind/resync and broadcast patch
VS1003B	wmarew3.c	0x300 .. 0x37f	Broadcast patch only

2 Usage

In addition to normal usage of SCI registers, two communication registers are used:

Register	Usage
SCI_AICTRL1 (0x0D)	Seek indication, negative for backwards
SCI_AICTRL2 (0x0E)	Synchronization

When you want to make a seek in the WMA file, use the following algorithm:

1. Write 0x1234 to SCI_AICTRL2.
2. Continue sending linear data until SCI_AICTRL2 becomes 0x2345.
3. Start sending data from a new location.
4. SCI_AICTRL2 becomes 0 when a frame has been correctly decoded.
5. When SCI_AICTRL1 is zero, write the seek size to SCI_AICTRL1 (i.e. the number of skipped bytes, negative for seeks backwards) to update file position inside VS1003B.
6. SCI_AICTRL1 becomes 0 when the seek has been handled.

You can take advantage of the WMA bytes per second information in SCI_HDAT0 to determine a good skip amount.

Note: if the seek size is for example 50000 bytes, first write 32767 to SCI_AICTRL1, continue sending data normally until SCI_AICTRL1 becomes 0, then write the remaining 17233 (50000-32767).

If you fail to provide the seek information, the decoding can end prematurely or continue unnecessarily long.

The mode register bit SCIMB_OUT_OF_WAV can be used to end decoding in the middle, just like without the patch.

If you have enabled the broadcast mode (see chapter 3.2), the file size information is ignored and you do not need to provide the seek information. In this case you have to end the decoding by using SCIMB_OUT_OF_WAV or software reset.

If you use software or hardware reset, remember to reload all patch codes.

Note: the resynchronization is not 100% reliable. It is very robust with high bitrates, but the reliability decreases with lower bitrates.

3 Broadcast, WMA Streaming

If you are streaming WMA, you need to generate an appropriate ASF/WMA header that VS1003b understands. The following shows the minimum requirements for a WMA header.

- ASFHeader
  - 16 bytes: ASFHeader GUID
  - 4 bytes: header size
  - 4 bytes of header size high part skipped
  - 2 bytes: object count
  - 2 bytes: object count hi word, must be 0
  - 2 bytes: reserved 2 field must be 0x02
  - ASFFileProperties:
    - 16 bytes: ASFFileProperties GUID
    - 4 bytes: object size
    - 4 bytes of object size high part skipped
    - 64 bytes skipped
    - 4 bytes = file properties, not used in VS1003b
    - 4 bytes = minimum data packet length, must be > 0
    - rest skipped, if any
  - ASFStreamProperties:
    - 16 bytes: ASFStreamProperties GUID
    - 4 bytes: object size
    - 4 bytes of object size high part skipped
    - 16 bytes: ASFAudioMedia GUID
    - 24 bytes skipped
    - 4 bytes: size of audio media (must be >= 24)
    - 4 bytes skipped
    - 2 bytes: stream number
    - 4 bytes skipped
    - 2 bytes: 0x61 0x01 WMA 7 codec
    - 2 bytes: channels
    - 4 bytes: frequency
    - 4 bytes: bytes per seconds
    - 2 bytes: block align
    - 4 bytes skipped
    - 2 bytes: encode options
    - rest skipped
  - rest skipped
- ASFDataObject:
  - 16 bytes: ASFDataObject GUID
  - 4 bytes: data size
  - 4 bytes of data size high part skipped
  - 26 bytes skipped
  - Data packets follow...

The WMA Broadcast flag is not yet supported in VS1003B firmware, thus the data size field should be correct or replaced with a high enough value, but small enough that the bitend variable (bit pointer to file end) does not overflow. A good value for the data object size in the broadcast case 0x1ff00000. If the file size (header size + data size indicated in the data object) is larger than 0x10000000, WMA rewind patch assumes a broadcast stream and decodes forever or until OUT_OF_WAV is set or resync fails.

3.1 Explicit Packet Length < Minimum Data Packet Length

In some files the explicit ASF packet length is smaller than the minimum data packet length defined in the ASF header (for example header minimum data packet length 1567, explicit packet length 1564) and unfortunately the ASF parser in VS1003b does not handle this case correctly.

This version of the patch handles this case by updating the padding size field automatically.

If you are not using the patch, you can set the minimum data packet length in the ASF header to the right value, i.e. the actual packet length (1564 in the example) and remove the padding bytes. Or you can encode the difference as padding (and send the padding too).

1. calculate the difference 1567-1564 = 3
2. replace the explicit packet length with header packet length 1564 → 1567
3. set "flags" field to include padding length type (if not already set to something)
4. add the padding field with the difference 3 (or add 3 to an existing padding value)

This fix is upwards-compatible, i.e. you don't need to remove the fix when you use a chip with a fully capable ASF parser.

3.2 Setting Broadcast Mode (`bitend`)

The "bitend" variable marks the end of the WMA file so that VS1003 knows when to end the decoding and return to try to find another audio file type. It is a 32-bit bit offset into the file, so generally it is eight times the file size in bytes.

If the value of `bitend` is larger or equal to 0x80000000UL, broadcast mode is active. You can do this by setting the data object size in the header to 0x1ff00000. But you can also enable broadcast mode after WMA decoding has started by writing 0x1953 to SCI_WRAMADDR, then writing 0x8000 to SCI_WRAM.

This is useful because in broadcast mode the file size is ignored and you do not need to update the file size when you perform jumps in the file.