

VS1011E WAV FIX

VSMPG “VLSI Solution Audio Decoder”

Project Code: VS1011e
Project Name: Support

All information in this document is provided as-is without warranty. Features are subject to change without notice.

Revision History			
Rev.	Date	Author	Description
1.0	2013-05-23	PO	Initial version

1 Description

The old WAV (RIFF) parser in vs1011e and vs1003b is not very intelligent, it expects a wav file to have the chunks in a specific order. Some files have extra chunks in them that the parser does not know how to skip, so such files do not play at all.

This patch allows the parser to skip unknown chunks in the file.

Chip	File	IRAM Location	Description
VS1011E	wav11e.plg	0x30 .. 0x4b	plugin format
VS1003B	wav03b.plg	0x30 .. 0x4b	plugin format
VS1011E	wav11e.c	0x30 .. 0x4b	loading table format
VS1003B	wav03b.c	0x30 .. 0x4b	loading table format
VS1011E	wav11e.cmd	0x30 .. 0x4b	command file format
VS1003B	wav03b.cmd	0x30 .. 0x4b	command file format

Hardware or software reset will deactivate the patch. You must reload the patch after each hardware and software reset.

This patch uses the application address to start automatically (the last entry in the patch tables writes to SCI_AIADDR), but does not use it afterwards. So, you must load any patch that actually uses the application address after this patch or it will be deactivated.

This patch is not compatible with the MPEG2.0 Layer 2 Patch nor with the MPEG2.5 Layer-3 8kHz Stereo Patch.

2 How to Load a Plugin

A plugin file (.plg) contains a data file that contains one unsigned 16-bit array called plugin. The file is in an interleaved and RLE compressed format. An example of a plugin array is:

```
const unsigned short plugin[10] = { /* Compressed plugin */
    0x0007, 0x0001, 0x8260,
    0x0006, 0x0002, 0x1234, 0x5678,
    0x0006, 0x8004, 0xabcd,
};
```

The vector is decoded as follows:

1. Read register address number `addr` and repeat number `n`.
2. If (`n & 0x8000U`), write the next word `n` times to register `addr`.
3. Else write next `n` words to register `addr`.
4. Continue until array has been exhausted.

The example array first tells to write 0x8260 to register 7. Then write 2 words, 0x1234 and 0x5678, to register 6. Finally, write 0xabcd 4 times to register 6.

Assuming the array is in `plugin[]`, a full decoder in C language is provided below:

```
void WriteVS10xxRegister(unsigned short addr, unsigned short value);

void LoadUserCode(void) {
    int i = 0;

    while (i < sizeof(plugin)/sizeof(plugin[0])) {
        unsigned short addr, n, val;
        addr = plugin[i++];
        n = plugin[i++];
        if (n & 0x8000U) { /* RLE run, replicate n samples */
            n &= 0x7FFF;
            val = plugin[i++];
            while (n--) {
                WriteVS10xxRegister(addr, val);
            }
        } else { /* Copy run, copy n samples */
            while (n--) {
                val = plugin[i++];
                WriteVS10xxRegister(addr, val);
            }
        }
        i++;
    }
}
```