Website: www.heber.co.uk
Sales: +44 (0) 1453 732302
sales@heber.co.uk
Support: +44 (0) 1453 732312
support@heber.co.uk

Pluto 6 Technical Bulletin

This Technical Bulletin applies to Pluto 6. For other information, refer to the **PLUTO 6 SOFTWARE DOCUMENTATION HELP FILE** (Heber part no: 80-17314).

Pluto 6 Streamed Sound Driver Recommended Usage

This technical bulletin corresponds to the version of the Streamed Sound Driver in Issue 6 of the Pluto 6 Interface layer software.

To achieve maximum reliability when using the Pluto 6 Streamed Sound Driver the following recommendations should be observed:

1. Sound Sample Preparation on Compact Flash

1.1. Always use a single file format for sound files within a game. It is recommended that 16Khz U-law compressed files be used for maximum reliability.

1.2. Always de-fragment the Compact Flash FAT32 file system before use on the Pluto 6. This can be done using the disk defragmenter program supplied with Microsoft® Windows®.

2. Using Sound Samples on Pluto 6

2.1. Initialise all sound file descriptors before playing any sound samples. If sound samples are playing they should be stopped before initialising any new samples.

2.2. It is recommended that the function SearchForWavFile() is used for initialising sound files rather than the InitWavFile() function. If the SearchForWavFile() function is used to initialise the sound sample, the PlaybackWavFile() function should be used to play the sound.

2.3. Do not attempt to access the Compact Flash card for other purposes, for example to load an image file, whilst a sound sample is playing.

2.4. Do not attempt to play stereo samples on mono channels.

2.5. Do not attempt to play mono samples on stereo channels.

Detailed Information

The above recommendations for usage are derived from the way in which the sound, ATA and FAT32 drivers on Pluto 6 have been implemented. This section provides a detailed description of how sound samples are played on Pluto 6.

1. Initialising the Sound Sample

Two functions are provided to initialise the sound sample to be played:

- void InitWavFile(BYTE *filename, WAVFILEINFO *wavFileInfo)

The InitWavFile() function calls the FileOpen() function in the FAT32 driver. A linked list of the disk clusters used by the file is saved in Pluto 6 memory, and a WAVFILEINFO structure is populated with information from the file header.

- BYTE SearchForWavFile(BYTE *filename, FILE_DESCRIPTOR *file)

The SearchForWavFile() function also calls the FileOpen() function in the FAT32 driver. A linked list of the disk clusters used by the file is saved in Pluto 6 memory, and a FILE_DESCRIPTOR pointer is saved to allow the file to be accessed.

The InitWavFile() function will typically store a large amount of data in the WAVFILEINFO structure for every file opened. Because of the amount of available memory for storing open file information it is not possible to have many files open at the same time.

The FILE_DESCRIPTOR used by the SearchForWavFile() function uses less memory that the WAVFILEINFO structure used in the InitWavFile() function. It is therefore recommended that the SearchForWavFile() is used in applications that require multiple files to be open.

It should also be noted that if the InitWavFile() function is used, and the associated WAVFILEINFO structure declared locally, this may result in stack overflow due to the amount of data contained in the structure. The smaller FILE_DESCRIPTOR used in the SearchForWavFile() function will take less stack space if declared locally.

Process access to the Compact Flash card is not scheduled or prioritised. If access requests to the Compact Flash card file system exceed the available processing time failures will occur. It is recommended that sound sample files are not initialised whilst open files are being played to avoid exceeding the available processing time. If available processing time is exceeded in this way, either files will not be opened correctly and will be inaccessible, or failure on the DMA channel used to send sound data to the audio amplifier will occur. Failures will also occur if image files are loaded whilst sound samples are being initialised or loaded.

If it is necessary to load new samples, to avoid exceeding available processing time, it is advisable to stop any sound samples that are playing using the StopAudioChannel() function. The SoundChannelBusy() function can be used to determine if a channel is in use and needs to be stopped as demonstrated in the code sample below:

```
while(SoundChannelBusy(channelnumber))
{
StopAudioChannel(channelnumber);
}
```

It should also be noted that a linked list of clusters used by the opened sound file is stored in memory on the Pluto 6 board. If the file that has been opened is not fragmented, and the clusters are contiguous, the linked list will only contain the start and end clusters. To ensure the minimum memory usage when opening files it is highly recommended that files/file systems on the Compact Flash card are de-fragmented.

2. WAV File Format Usage

The sound driver supports a variety of .wav file formats. U-Law compressed files offer 2:1 compression ratio. The decompression of the sound data is done in real time with a relatively low processing overhead. Using U-Law compressed files does mean that the amount of data transferred from the Compact Flash card is reduced. The reduced Compact Flash access reduces the overall processing overhead involved with playing sound samples.

3. Audio Channels

The driver will allow developers to specify playing a stereo file over a mono channel. When the audio data is processed in an Interrupt Service the driver will detect that a stereo sample is being played over a mono channel and will change the channel status to: CHANNEL_INVALIDCHANNEL.

If a mono sample is played over a stereo channel the sound will only be audible over the left channel, and the right channel will be silenced. Playing mono samples over stereo channels is not recommended because playback is restricted to the left channel only.