Tarsus Archive Data Format

Revision M 03/22/2018

Ulyssix Technologies, Inc
7470 New Technology Way, Suite B, Frederick, MD 21703-9461
Tel: 301-846-4800 ~ Fax: 301-846-0686 ~ www.ulyssix.com

Tarsus Archive Data Files Explained

Archive files recorded by the ALTAIR software are binary files containing data from either the Decom or Frame Sync circuitry of the ALTAIR hardware. The archive data files begin with a 328-byte File Header. After the File Header, the rest of the file is composed of minor frame blocks. Each minor frame block begins with a 12-byte Minor Frame Header, which contains the time stamp of the minor frame and lock indicator information. After the Minor Frame Header, the minor frame data follows. All archive data files end with the extension ".tad" (Tarsus Archive Data). This section of the manual explains the format of the archive data.

For a Decom archive file, the PCM data is received in the hardware, bit synchronized, frame synchronized, and then decommutated. The decommutated data is word aligned and stored into a large dual port memory device. This byte aligned data is stored as the minor frame data in the archive file.

For a Frame Sync archive file, the PCM data is received in the hardware, bit synchronized, and frame synchronized. The minor frame data is packed into a bit stream and stored into a large dual port memory device. This continuous bit stream data is stored as the minor frame data in the archive file.

1.1 Data Storage Format

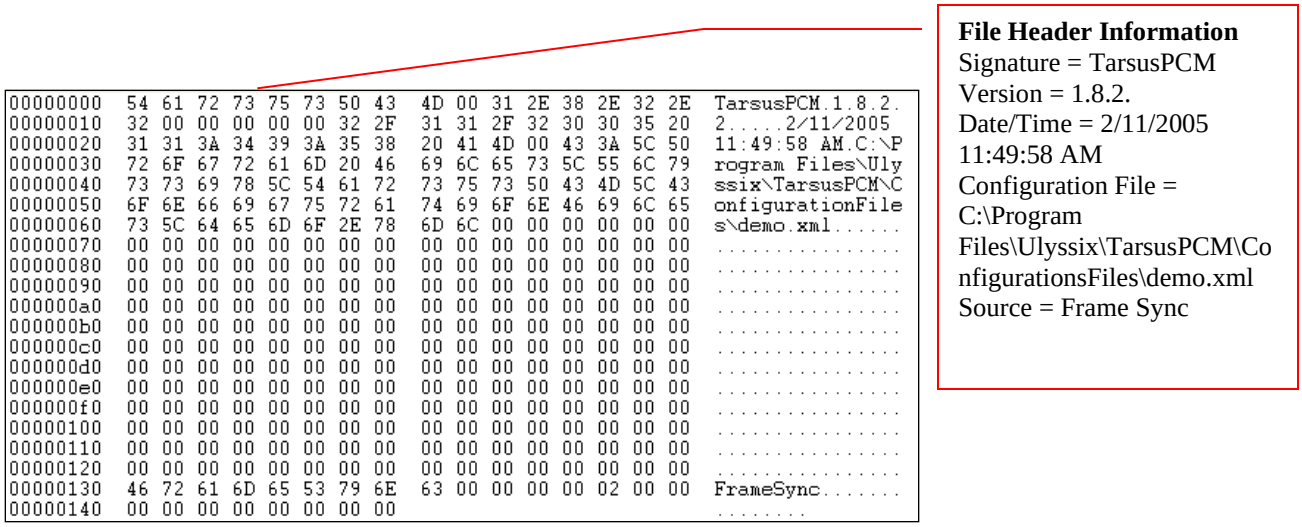
The TarsusHS hardware and ALTAIR software stores the archive data in "Little Endian" format. "Little Endian," derived from the phrase "Little End In," means the little end of the data is stored in memory first. For example, 0x12345678 would be stored in memory as (0x78 0x56 0x34 0x12).

1.2 File Header Definition

Each archived data file contains one file header structure. The file header is stored to indicate the date, time, and configuration file used during the archive sequence. All file header data is in ASCII characters to allow viewing with a standard text editor. The header consists of 328 bytes and is defined as follows:

10 bytes – Signature
12 bytes – Version
22 bytes – Date/Time **Note:** Time stamped when the file is created
260 bytes – Configuration file and path
12 bytes – Input Data Source ("Frame Sync" or "Decom")
1 Unsigned Integer (32 bits) - Bits/Minor Frame
1 Unsigned Integer (32 bits) - Spare
1 Unsigned Integer (32 bits) – Spare

1.2.1 File Header Example



File Header Information
Signature = TarsusPCM
Version = 1.8.2.
Date/Time = 2/11/2005 11:49:58 AM
Configuration File = C:\Program Files\Ulyssix\TarsusPCM\ConfigurationsFiles\demo.xml
Source = Frame Sync

Figure 1 – Archive File Header Example

1.3 Minor Frame Header Definition

As stated above, header data precedes every minor frame in the archive file. The header contains time in Binary Coded Decimal (BCD) along with various status indicators. The time stamp for each minor frame occurs at the first bit on the frame sync pattern. The header data is defined as three 32-bit data words with the following format.

32 Bit Words	Bits																													
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0000				100's days				10's days				1's days				10's hours				1's hours				10's minutes				1's minutes			
1	10's seconds				1's seconds				100's msec				10's msec				1's msec				100's usec				10's usec				1's usecs			
2	Minor Frame Count																TC R Bit	TC F Bit	I/E	Slip	# Sync Errors				B Lock	SF Lock		MF Lock		FFI		

Table 1 – Archive Data Header Definition

- Time Code Reader Status Bit
- Time Code Flywheel Status Bit
- Bit Sync Source (0 = Internal, 1=External)
- Bit Slip (1 = yes, 0 = no)
- Bit Sync Loop Lock
- Sub-Frame Lock (2 = Lock, 1 = Check, 0 = Search)
- Minor Frame Lock (2 = Lock, 1 = Check, 0 = Search)
- Current Format # Indicator

1.4 Data Description

The TarsusHS hardware aligns the archived Decom data on 16-bit boundaries regardless of the PCM word size. The 16-bit alignment provides an efficient way of extracting a single word of data out of the archive file. The PCM data will always be right justified in a 16-bit word. For example, a 12-bit PCM word with a value 0x444 hexadecimal followed by another word with a value of 0x555 hexadecimal will show up in the archive file as 0x04440555. This value can be broken into four hexadecimal bytes: 0x04, 0x44, 0x05, and 0x55.

The ALTAIR hardware bit packs the archive Frame Sync data. The bit packing provides a smaller data size; however, it makes extracting a single word of data out of the archive file more difficult. For example, a 12-bit PCM word with a value of 0x444 hexadecimal followed by another word with a value of 0x555 hexadecimal will show up in the archive file as0x 444555. This value can be broken into three hexadecimal bytes: 0x44, 0x45, and 0x55.

The following examples show the actual archived data using a hexadecimal viewer application:

1.4.1 Archive Data 32-Bit Sync 16-Bit Data

For this Frame Sync configuration, a Decom Archive File and a Frame Sync Archive file will have identical data. Since the Frame Sync and Word Size are both integer multiples of 8-bits, the byte aligned Decom archive data and the bit stream Frame Sync archive data are identical.

Sync Pattern Size: 32-bits
Sync Pattern: FE6B2840
Minor Frames: 1
Words per Frame: 16

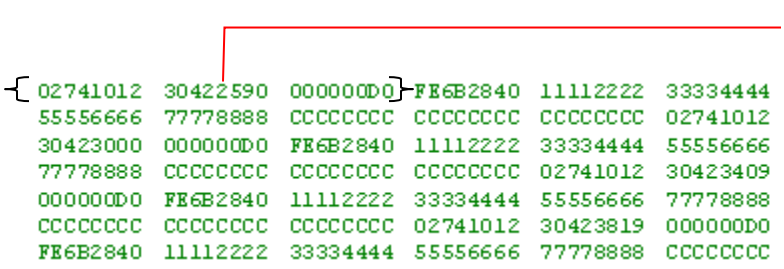


Figure 2 – Archive Data Header Example

Header Information
Time = 274:10:12:30.422.590 (Sampled 1st bit frame sync pattern)
Minor Frame Count = 0
Internal Bit Sync
No Bit Slips
Bit Sync Loop Lock
Sub-Frame = Lock
Minor Frame = Lock
Current Format = 0

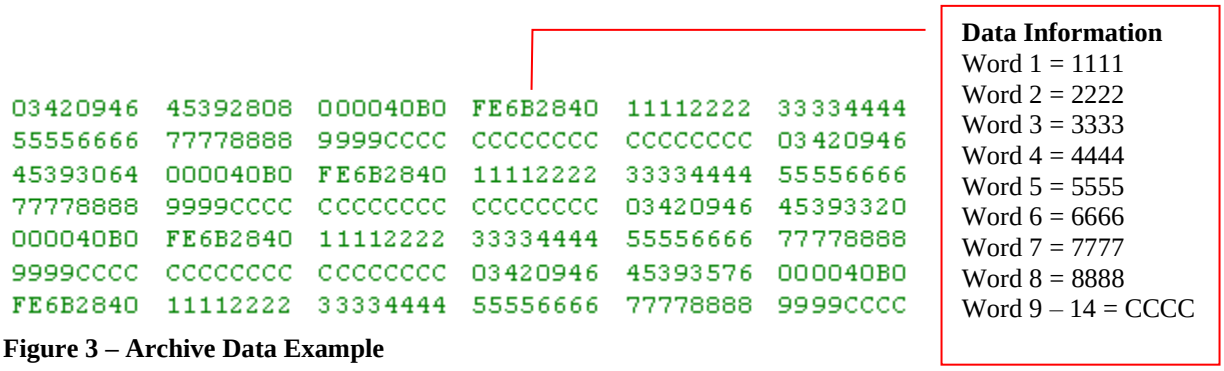


Figure 3 – Archive Data Example

1.4.2 Archive Data 24-Bit Sync 12-Bit Data

For this Frame Sync configuration, the Decom archive data will have leading zeros as the 12-bit decom word is packed into a 16-bit space. The Frame Sync archive will not have the leading zeros; therefore, the Frame Sync Archive minor frame will take up fewer bytes than the Decom Archive minor frame.

Sync Pattern Size: 32-bits
Sync Pattern: FE6B2840
Minor Frames: 1
Words per Frame: 8

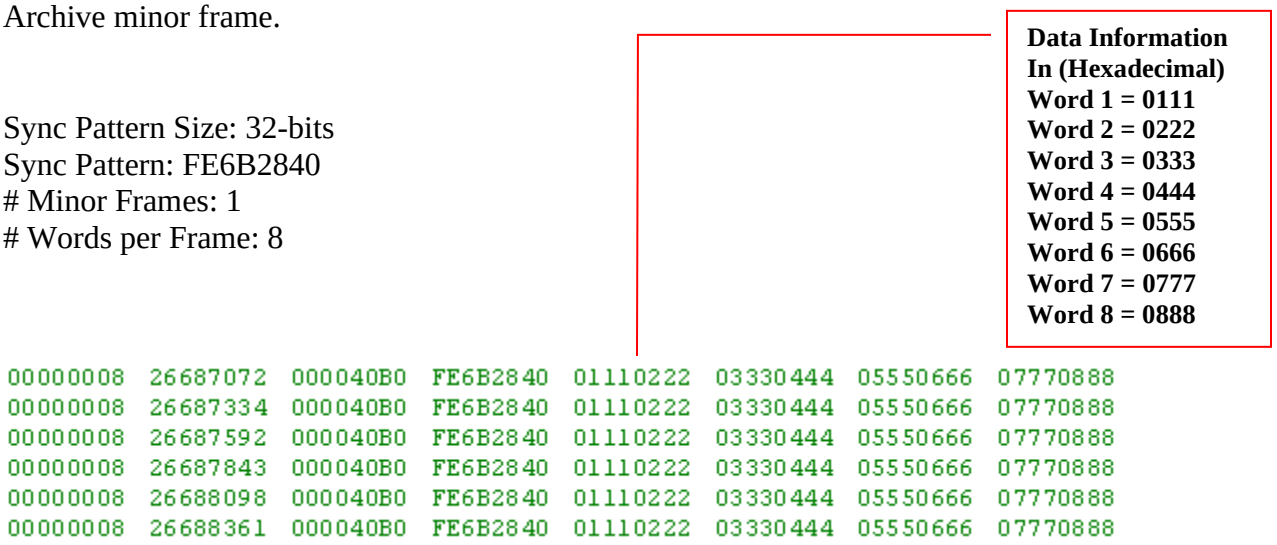


Figure 4 – 12-bit Decom Archive Example

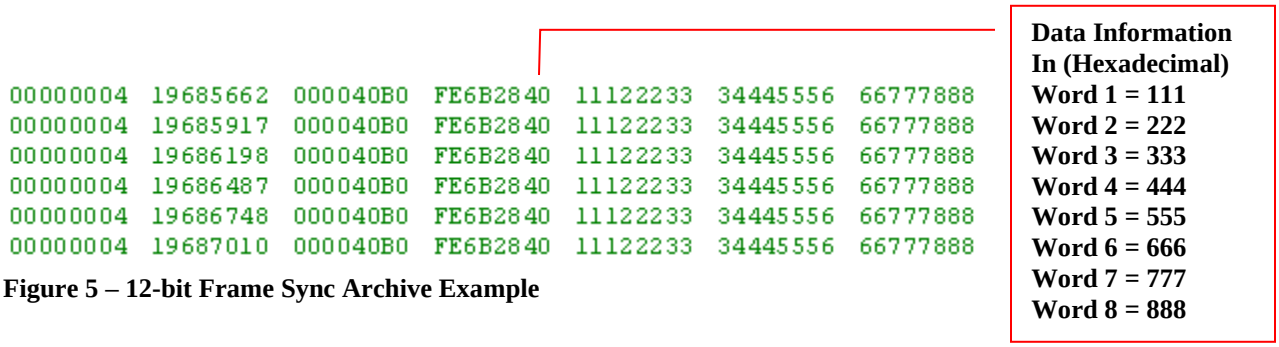


Figure 5 – 12-bit Frame Sync Archive Example