SOFTWARE CONVERSION OF LEGACY RECORDING FORMAT TO IRIG 106 CHAPTER 10 FILE

Richard A. Graham Jr.
US Navy, NSWC Corona
Corona, CA, 92878
richard.a.graham2@navy.mil

Steven G Shepherd
US Navy, NSWC Corona
Corona, CA, 92878
steven.g.shepherd@navy.mil

ABSTRACT

This paper examines how to convert files recorded on a legacy recorder to an IRIG 106 Chapter 10 file.

INTRODUCTION

NSWC Corona is the data repository for the US Navy. As such, we have a large number of recordings that go back a number of decades. The more recent recordings are all stored on optical discs (CD/DVD) or hard drives in IRIG 106 Chapter 10 format. Prior to these files NSWC Corona used primarily Sypris Model 80 recorders. These were not Chapter 10 recorders.

The Model 80 files used a somewhat proprietary file format. However, some of the structures in the files match up to the legacy IRIG106-09. This paper will be referencing IRIG 106 - Chapter 6 (DIGITAL CASSETTE HELICAL SCAN RECORDER/REPRODUCER, MULTIPLEXER/DEMULTIPLEXER, TAPE CASSETTE, AND RECORDER CONTROL AND COMMAND MNEMONICS STANDARDS) and IRIG 106 - Appendix L (ASYNCHRONOUS RECORDER MULTIPLEXER OUTPUT RE-CONSTRUCTOR (ARMOR)).

This paper will not provide every detail needed to make a program that converts legacy files to Chapter 10. Instead, this paper will list some of the details either that are less than clear in the IRIG documents, or where the files do not adhere to the standard.

FILE READ

We will start with Chapter 6, section 6.7. Originally, this section dealt with the S-VHS magnetic tape recording format. However, the Model 80 acts as a sort of simulated S-VHS recorder, but uses hard drives instead of tapes. In this section, we will see how to read the setup, and how the channels are recorded in the file.

Setup Block

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  53 4F 53 E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  SOSç=ç=ç=ç=ç=ç=ç
00000010  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
00000020  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
00000030  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
00000040  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
00000050  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
00000060  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
```

Figure 1

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

0001FFA0  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
0001FFB0  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
0001FFC0  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
0001FFD0  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
0001FFE0  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
0001FFF0  3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7 3D E7  =ç=ç=ç=ç=ç=ç=ç=ç
00020000  3D E7 3D 45 4F 53 07 6C 6C 07 20 20 20 20 20 20  =ç=EOS.ll.
00020010  36 2E 39 20 00 00 44 00 38 00 B4 C4 04 00 00 00  6.9 ..D.8.´Ä....
00020020  00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00  ................
00020030  00 0B 08 00 60 85 73 01 01 00 80 15 CE 05 3C 00  ....`…s...€.Î.<.
00020040  00 00 AC 70 2E 00 ED 07 00 00 14 00 0C 00 08 00  ..¬p..í.........
00020050  00 00 59 36 6A 02 00 4E 00 00 00 06 00 00 00 10  ..Y6j..N........
```

Figure 2

AT the beginning of each file will appear “SOS” (0x534F53). This stands for “Start Of Setup”. The files then repeats 0xE73D until “EOS” (0x454F53). This stands for “End Of Setup”. After that will appear the header section, the channel section, and the trailer section.

Header Section

The description of this section can be found in Appendix L, Table L-3.

*Note: The start of the header section will appear two bytes later. You will notice the 0x076C followed by 0x6C07 after the EOS. This is one of the places where the standard does not match the file. You can take the second 16-bit word and read it as least significant byte first. This may appear to be more convoluted than just taking the first 16-bit word as most significant byte first. However, the file is usually least significant byte first, so taking the second 16-bit occurrence is more consistent.

Channel Section

For the rest of the setup read through each channel. Note that each channel type is a different length. For example, PCM input and output setups are 51 bytes long, while analog input and output setups are 53 bytes long. These lengths are in Appendix L Table L-4.

*Note: The words are least significant byte first. For example, suppose the first two bytes of the first channel were 0x08 and 0x00. This is 0x0008, not 0x0800.

*Note: Table L-5 says type 1 is “8 BIT PCM INPUT”. It should be “8 MBIT PCM INPUT”

*Note: Table L-6 is titled “ANALOG INPUT AND OUTPUT CHANNELS”. It should be “PCM OUTPUT CHANNELS”.

Trailer Section

This section is described in Appendix L, section 2.3. The footer contains a description, a scanlist, and a checksum. The crucial part here is the scanlist. The scanlist describes the order of the channels as they are written to the file.

For those familiar with Chapter 10 files, you know that the packets are written in any order (except for things like TMATS and the first time packet, which are at the beginning of the file). In the legacy file system the data is written in a specific order, and not necessarily in the order the channels were listed in the setup.

Each channel of the scanlist uses three bytes. The first byte is the index (starting at one, not zero). The second and third byte is the length (least significant byte first).

The scanlist lists the channels as they appear in the data packets.

For example:

Scanlist Index	Channel Type	Description
1	15	Time 1
2	19	Time 2
3	20	Time 3
5	8	PCM 20 Mbps Input
4	8	PCM 20 Mbps Input
6	8	PCM 20 Mbps Input
7	8	PCM 20 Mbps Input
10	6	HF Analog Input
11	6	HF Analog Input
8	6	HF Analog Input
9	6	HF Analog Input

Table 1

In the above example, the packet would have the three time packets, then the PCM data, then the Analog data. However, the channels would not occur in order (4, 5, 6, 7, 8, 9, 10, 11). This is perfectly acceptable. However, if you assume that they occur in order you will not have the channels be assigned to the correct channels in the Chapter 10 file.

*Note: A Channel index of 0xFF is filler.

Reading Data

Step 1				2	3	4				5	6				7
1	2	3	4			1	2	…	m		1	2	…	m	
Sync				Filler		PCM Channel Blocks				Filler	PAR Channel Blocks				Filler

Figure 3: Chapter 6, Figure 6-13

Each packet contains a Sync, Filler, PCM, Filler, PAR Channel, and Filler.

The Sync word is 0xFE6B2840.

Time

The time words occur after the sync, but before the PCM. The time for each packet is split into three parts. The first two parts are each 24 bits long, and the third part is 16 bits long.

TABLE 6-14. TIME CODE WORD FORMAT

BIT	WORD 1	WORD 2	WORD 3
22	D8	S6	
21	D7	S5	
20	D6	S4	
19	D5	S3	
18	D4	S2	
17	D3	S1	
16	D2	S0	
15	D1	SE	0
14	D0	NT	0
13	0	0	HN13
12	H5	0	HN12
11	H4	MS11	HN11
10	H3	MS10	HN10
9	H2	MS9	HN9
8	H1	MS8	HN8
7	H0	MS7	HN7
6	M6	MS6	HN6
5	M5	MS5	HN5
4	M4	MS4	HN4
3	M3	MS3	HN3
2	M2	MS2	HN2
1	M1	MS1	HN1
0	M0	MS0	HN0

LEGEND

D = Day of year.
H = Hour of day.
M = Minutes past the hour.
S = Seconds past the minute.
MS = Milliseconds past the second.
HN = Hundreds of nanoseconds past the millisecond.
SE = Sync error (time code decoding error).
NT = No time code (input signal detect fail).
0 = Always 0.

Figure 4: Chapter 6, Table 6-14

You can see above that the decoding is straightforward. However, be aware that the time words are not necessarily adjacent as shown in the table. Many test files had different time layouts. Some would repeat the words more than once. This is where the scanlist comes in handy. For example, suppose that in one file each time word appeared three times each, and in another file they appeared twice. It would be problematic to find the pattern for every file. If you use the length of the time in each scanlist you will be able to deal with the pattern with no problem. If the length of the channels is a multiple of 24/16 then you can just read the first 24/16 bits and ignore the rest, since it is just a repetition.

PCM

Remember that the first 16-bit word of each PCM channel is the length of the data, in bits. The length in the setup is the number of bytes reserved for the channel. This allows the bit rate to vary by a couple percent and not overflow the space allocated for the channel. When moving to Chapter 10 it will be necessary to collect the data into bytes for Chapter 10/11 packetization.

CHAPTER 10 FILE WRITE

The details needed for writing a Chapter 10 file are far beyond the scope of this paper. However, I will note certain things that are specific to this conversion.

1) I found it important to pay attention to the sample rates of the analog channels. The Chapter 10 recorder I used for testing has specific analog sample rates. For example, if there were only a 2 MSa/sec and a 5 MSa/sec option the file would fail with a 2.5 MSa/sec channel. It was necessary to change/oversample certain channels. However, techniques for the sample rate changes go beyond the scope of this paper.
2) I found it relatively easy to set up TMATS code specific to this project. Fields like the Channel Data Link Name were easy to set up. Channel numbers were assigned sequentially. My entire TMATS code is around 200 lines.
3) Pay attention to details like the Chapter 10/11 packet length. In Chapter 11, section 11.2.1.1c it says, “The value shall be in bytes and is always a multiple of four”. Details like this may make the file fail on load/import.

OTHER ISSUES

One of the main problems with reading the file is that during recording many operators used the analog input to record time, but did not hook up IRIB time to the time card input. This means that a large percentage of the files that were tested showed January 1st for the date. It was necessary to either accept this and add a time offset during analysis, or to decode the IRIB time from an analog channel. However, the decoding portion goes beyond the scope of this paper.

CONCLUSIONS

It is possible to convert from legacy recording formats to Chapter 10. Many details cause the entire process to fail. It is necessary to have one or more developers that are familiar with telemetry recorders. It is recommended that the developer spend significant time with a hex viewer and to work through the packet details in the documentation.

REFERENCES

[1] IRIG 106-09 – Chapter 6, "DIGITAL CASSETTE HELICAL SCAN RECORDER/REPRODUCER, MULTIPLEXER/DEMULTIPLEXER, TAPE CASSETTE, AND RECORDER CONTROL AND COMMAND MNEMONICS STANDARDS", Apr. 2009.
[2] IRIG 106 – Appendix L, "ASYNCHRONOUS RECORDER MULTIPLEXER OUTPUT RE-CONSTRUCTOR (ARMOR)", Apr. 2009.
[3] IRIG 106 – Chapter 11, "Recorder Data Packet Format Standard", July. 2017.