SOFTWARE PAM

Richard A. Graham Jr.
US Navy – NSWC – Corona – Telemetry Systems
Corona, CA, 92878
RAGraham@ieee.org

ABSTRACT

Pulse Amplitude Modulation (PAM) is a legacy modulation technique still in use in older telemetry systems. Normally a telemetry system relies on hardware solutions to demodulate and decommutate PAM. This paper examines now a software solution can process from baseband.

INTRODUCTION

This paper is targeted for use by a programmer responsible for telemetry processing. It should be relatively easy to expand the base program presented to cover a range of PAM telemeters. The essential theory will be discussed, as well as a discussion of trade-offs made for speed of execution. Some recommendations will be suggested as to future expansion.

THE BODY

PAM Basics

As of IRIG 106-17 The PAM documentation appears in appendix K, Annex A-1 [1]. This paper will focus on NRZ-PAM. Specifically, the paper will make reference to one specific test file, produced by a PAM simulator and recorded on a modern telemetry recorder.

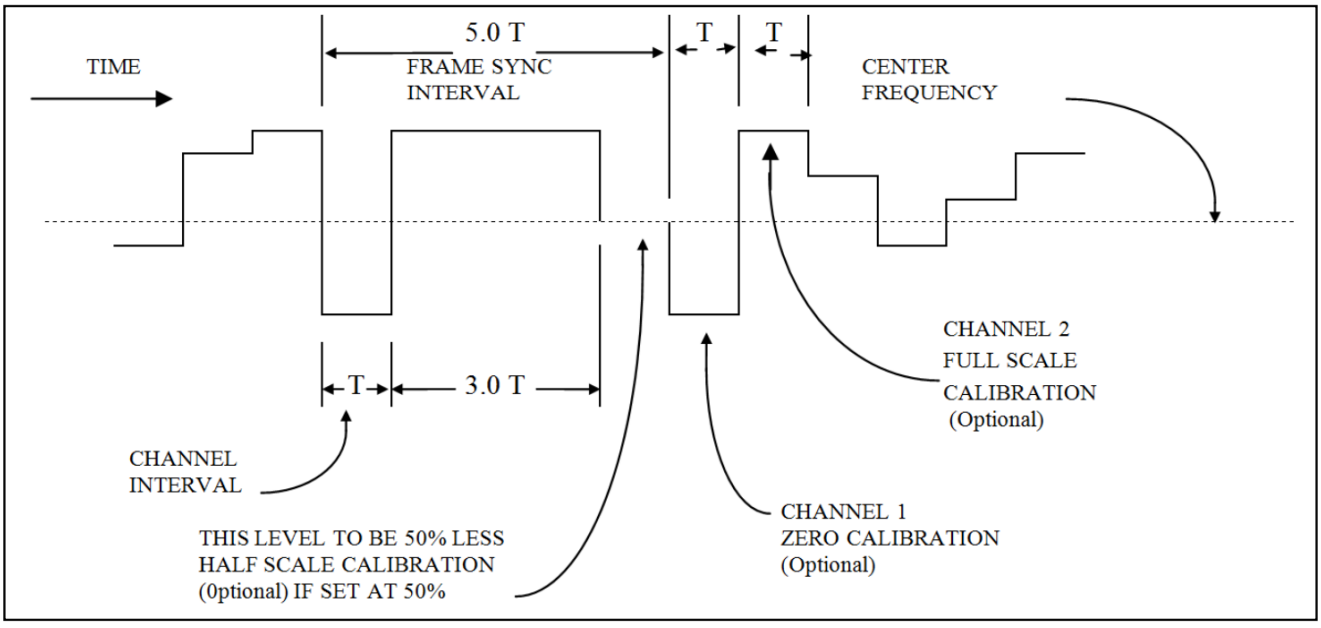

Figure 1: 100 percent duty cycle PAM with amplitude synchronization

The baseband PAM consists of several “channels”. Each of these channels represents something like a sensor voltage, or a discrete value, like one bit. These channels are then serialized in a specific order, and a synchronization pattern is inserted between each frame. This pattern is “zero, full, full, full, half”. So, for the width of one channel the signal will be near zero volts. Then for the width of three channels the voltage will be near 100% of the maximum voltage. Then for the width of one channel the voltage will be near 50% of the maximum voltage.

The test signal was set up to output a total of 64 channels, including synchronization pattern. The rate was set to 36k channels per second. A few other options were selected such that interesting data would be produced, instead of flat voltages.

Fast PAM Demodulator

It was decided that the PAM demodulator needed to process faster than “1x”. 1x would represent real-time. So 2x would be twice as fast, whereas 0.5x would be half speed. For example, if a one-minute file were to take 30 seconds to process then it would be at 2x. This speed requirement was important to allow the program to be used for range safety calls. Also, it would become unusable if the processing took hours.

Software PAM Steps

The major steps required for PAM Demod/Decom are

1) Auto scale
2) Synchronization Detect
3) Decommutation

A short explanation for each will follow:

1) Auto scale:

The signal arrives at a level that we will need to scale. It is necessary to scale the signal such that the voltages are presented from zero to 100%. For example, we may decide to auto scale the received signal for every 1000 sample points. It is necessary to include the sample rate in this calculation. For example, the test file samples at 2 million samples per second. So for the rate of 36000 channels per second we arrive at more than 55 samples per channel. For the chosen 1000 samples it would include only 18 channels. That is less than a frame. In this case it is probably best to choose a much higher number. In the software the author chose to scale one Chapter 10 packet at a time. This resulted in may thousands of samples all scaled at the same time. This appears to work well.

2) Synchronization Detect:

The next step is to step through the samples and look for the zero, full, full, full, half pattern. So the test program looks for the first channel to be less than 0.1, the next three channels to be greater than 0.9, and the fifth channel to be between 0.4 and 0.6. After this the program takes the first 64 channels after this pattern in the list and sends them for decommutation.

The recorder will have sampled the data at a higher rate than the channel rate. Again, at our 36k channel rate, with a 2M Sa/sec recorder rate we will have over 55 samples per channel.

It is necessary to determine the Channel Period. This will then be used to find the time offset of each channel. These offsets will be added to a current sample time as the program loops through the samples.

Example:

Current start sample: 137.000000 sec

Channel Rate: 36000 Channels / sec

Channel Period: 28 usec

T0: 137.000000 sec

T1: 137.000028 sec

T2: 137.000056 sec

T3: 137.000083 sec

T4: 137.000111 sec

We then look at the value at each time. The test program finds the first samples after each designated sample. These samples are then used to test for synchronization. If the test fails, the program iterates to the next sample and recalculates the time offsets. If the synchronization succeeds then the program calculates the time offsets for all the channels, finds the samples at these time offsets, and sends the samples to decommutation.

3) Decommutation:

At this point you should have 64 channels all lined up and ready to send to an output file or to a GUI. The demo program displays the channel values as text and as bars. Any further processing will require a data dictionary. In this case the test file just puts out a pattern, so the output is sent to the GUI directly.

CONCLUSIONS

The original test was to see if we could replace hardware PAM equipment with software. The answer appears to be a resounding yes. In the past it has been necessary to purchase and maintain various pieces' hardware for many years, including after most organizations have moved on to other modulation techniques. This requires people with the knowledge to operate the equipment. It also requires the space to store the equipment. This is time and money not spent on other things.

As long as organizations continue to use legacy telemeters it will be necessary for the telemetry organizations to assess the priority for support of these telemeters. It is hoped that this will be a better solution than what has been offered so far.

COMPUTER REQUIREMENTS

The software was tested on several computers. All had Intel Core i7 processers, with 16 gigabytes of RAM. The software does utilize multiple cores. Extensive testing was not performed in regards to minimum requirements. A modern computer should work fine. Further testing on other computers, including small single board computers, is still needed. The software requires .Net 4.5 or higher.

REFERENCES

[1] Range Commanders Council Telemetry Group, Range Commanders Council, White Sands Missile Range, New Mexico, *IRIG Standard 106-17: Telemetry Standards*, Annex A.1, 2017. (Available on-line at http://www.wsmr.army.mil/RCCsite/Documents/106-17_Telemetry_Standards/annexa-1.pdf).