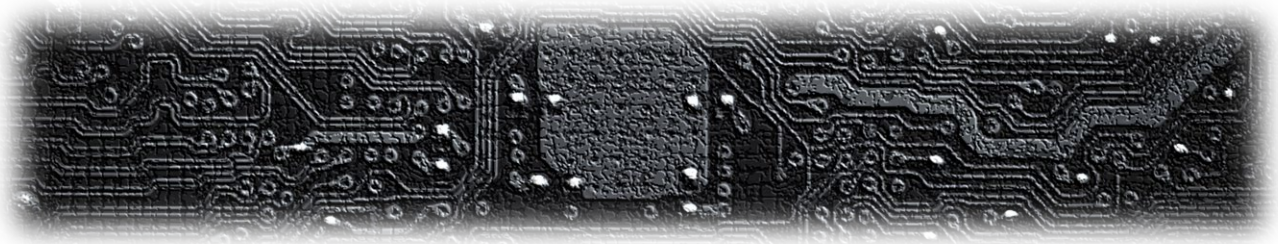




ERDOSMILLER



Advanced Mud Pulse Telemetry — M-ary Encoding for MWD tools (Chapter 2)

Protocol Definition Document

Revision 1.3

Authored By:

Kripa Nidhi

David Erdos

Table of Contents

Table of Contents	2
1 Introduction.....	3
2 Synch Block.....	3
3 Survey Sequence Number Block.....	4
4 Toolface/Logging Sequence Number Block.....	5
5 Data Block Format	6
5.1 Logical Data Block.....	6
5.2 Zero-Offset Scaled Data Block	8
5.2.1 Scaling.....	8
6 Breaking Binary Data Into 2 and 3-bit Packets.....	10
7 M-ary Pulse Encoding	10
7.1 Example.....	11
7.1.1 Scaling.....	11
7.1.2 3-Bit Packet	11
7.1.3 Pulse Encoding	11

1 Introduction

This document covers the encoding and higher-level design of M-Ary mud pulse telemetry, including how the data is actually encoded into the 2 and 3-bit packets. For an introduction to M-Ary telemetry, please see the MUD Pulse Telemetry M-ARY Encoding and Decoding white paper by Erdos Miller.

The mud-pulse telemetry transmitter sends out blocks of encoded pulses in the following order after every flow cycle:

1. Synch Block
2. Survey Sequence Number (1-4)
3. Survey Data
4. Toolface/Logging Sequence Number (1-4)
5. Toolface/Logging Data
6. Repeat steps 4 and 5.

For the purpose of this document, we will define the pulse width – which is fixed as 2 TU (Time Units) wide. Most Current Mud pulse telemetry systems in use, define the minimum pulse width value to be 250 milliseconds and a maximum value of 3 seconds.

Acceptable Mud pulse widths (in seconds) are: 0.250, 0.375, 0.500, 0.600, 0.800, 1.0, 1.2, 1.5, 2.0, 3.0

2 Synch Block

The most common type of synchronization is what is known as the 1111 synchronization sequence, this is what will be covered in this document. The Synchronization Block consists of 4 pulses (each 2 TU's wide) that are 4 TU's apart. Note that the distance between the last sync pulse and the first pulse of the next block (*Survey Sequence Number*) should be the sum of the final spacing of the *Sync* block and the spacing before the first pulse for the *Survey Sequence Number* block. For *Survey Sequence Number 1*, the spacing between the last sync pulse and the first *Survey Seq Number* pulse would be 15 (4+11) TU's.

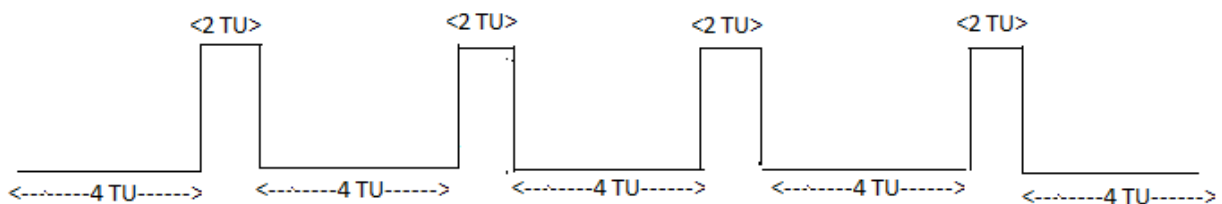


Figure 1: Example of the Synch Block consisting of 4 pulses.

3 Survey Sequence Number Block

Most M-Ary mud pulse telemetry controllers allow for Survey Sequence numbers 1-6; however, some older systems only support 1-4. For simplicity, this document covers only the pulse pattern for Survey Sequence Numbers 1-4.

The total spacing *before* the first pulse of the Survey Sequence Number would be the sum of the final spacing of the *Synch* block and the first pulse of the *Survey Seq Number* block. Similarly, the total spacing after the last pulse of the Survey Sequence block would be the sum of the final spacing of the Survey Sequence block and the spacing before the first pulse of the next block (*Survey Sequence Data* Block or the *Toolface/Logging Sequence Number* block). Examples of what the pulse trains for survey sequences 1-4 are shown below in Figure 2, Figure 3, Figure 4, and Figure 5.

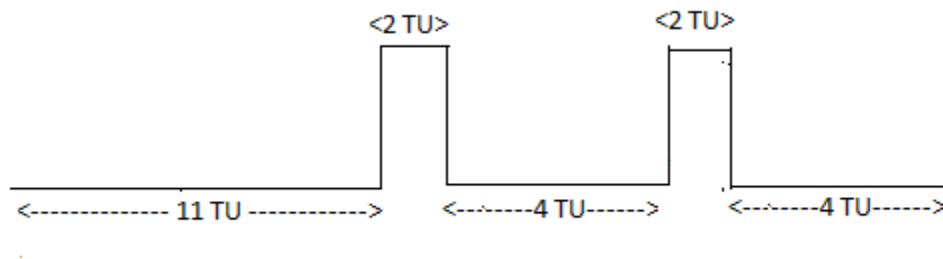


Figure 2: The pulse train for Survey Sequence number 1.

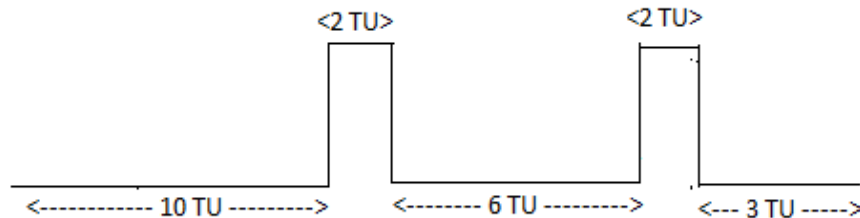


Figure 3: The pulse train for Survey Sequence number 2.

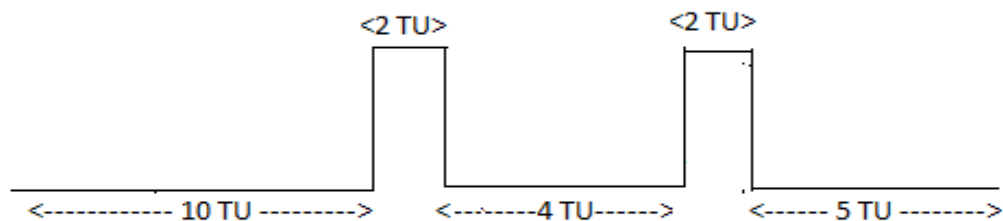


Figure 4: The pulse train for Survey Sequence number 3.

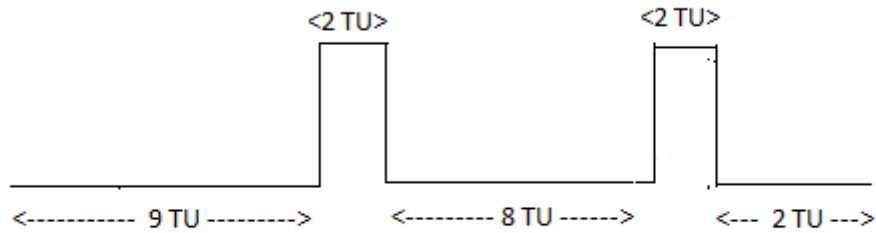


Figure 5: The pulse train for Survey Sequence number 4.

4 Toolface/Logging Sequence Number Block

Most M-Ary mud pulse telemetry controllers allow for Toolface/Logging Sequence numbers 1-6; however, some older systems only support 1-4. For simplicity, this document covers only the pulse pattern for Toolface/Logging Sequence numbers 1-4. Examples of what the pulse trains for toolface/logging sequences 1-4 are shown below in Figure 6, Figure 7, Figure 8, and Figure 9.

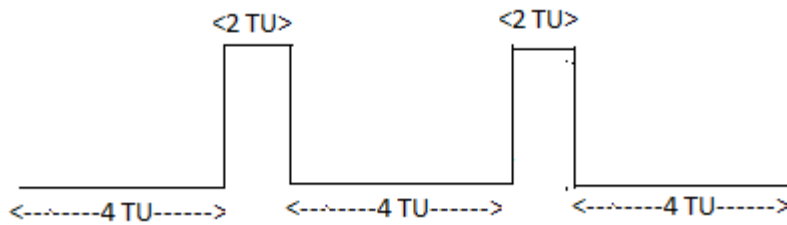


Figure 6: The pulse train for Toolface/Logging Sequence number 1.

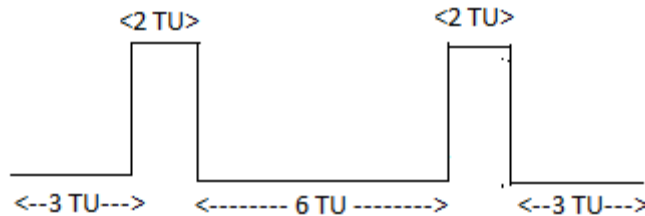


Figure 7: The pulse train for Toolface/Logging Sequence number 2.

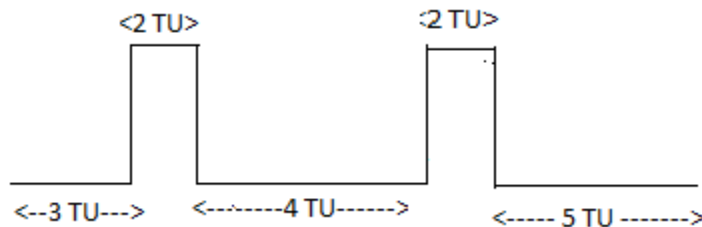


Figure 8: The pulse train for Toolface/Logging Sequence number 3.

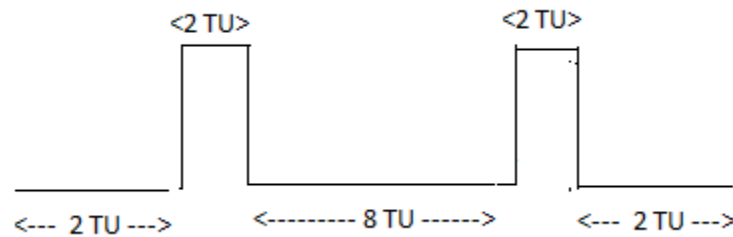


Figure 9: The pulse train for Toolface/Logging Sequence number 4.

5 Data Block Format

Both the Survey Sequence Data and the Toolface/Logging Sequence data have similar data block format specifications. This block encodes the data values of the various the tool measurements in the format specified in the Toolface/Logging or Survey Sequence. The data format defines how the various tool measurements are encoded in binary in the format specified. This document covers the following types of data.

1. Logical Data Block
2. Zero-Offset Scaled Data Block

5.1 Logical Data Block

The Logical data formats are used to encode/decode boolean Logical parameters (with values of either TRUE/FALSE) such as Bat2 and BattW values. The typical definition for Bat2 in the Sequence (Survey or Toolface) would be written as Bat2:P or Bat2. The presence of the “:P” indicates that parity info must be included in the data block.

A logical value is encoded within a total time period of 8 TUs. The pulse-coding of logical values are shown in Figure 10, Figure 11, and Figure 12.

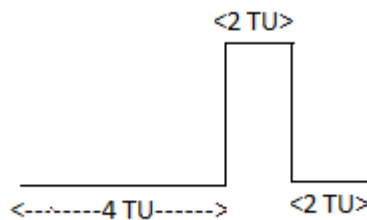


Figure 10: Value of FALSE (0) with or without parity.

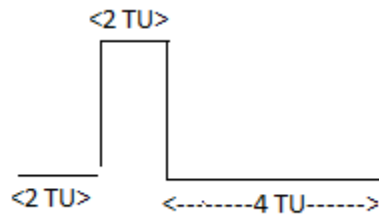


Figure 11: Value of TRUE (1) with parity.

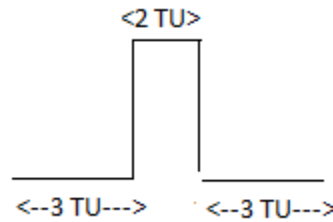


Figure 12: Value of TRUE (1) without parity.

5.2 Zero-Offset Scaled Data Block

Zero-offset format is the most common mode of transmitting downhole measurements. An example of a survey sequence with all zero-offset measurements is: Inc:11:P Azm:12:P Grav:10:P DipA:10:P MagF:12:P Temp:6:P BatV:8:P. To transmit a certain value, the pulse encoding for zero-offset data goes through a four-step process:

1. Scaling
2. Conversion to specified format
3. Breaking up the binary value into 2 or 3-bit packets
4. M-ary pulse encoding

5.2.1 Scaling

Given a value of X in engineering units, the scaling for different measurements is shown in the table below for Zero-Offset Scaling. *Note that at the receiver/decoder will have to apply the inverse operation to convert the decoded value to its engineering units.* For an n -bit wide integer in the table, let M be defined as:

$$M = 2^n - 1$$

In the special case of azimuth and toolface measurements, we define P as:

$$P = 2^n$$

The scale factors and offsets for common zero-offset measurements are presented in Table 1.

Measurement	Engineering Scaling (k) for Integer	Offset (b)	Conversion to Final Integer Value
Inclination	180	0	$((X+b)/k)*M$
Azimuth	360	0	$((X+b)/k)*P$
TFA (Mag & Grav & automatic)	360	0	$((X+b)/k)*P$
MagF	1	0	$((X+b)/k)*M$
Gravity	2	0	$((X+b)/k)*M$
DipAngle	180.00	90	$((X+b)/k)*M$
Raw Gamma	255	0	$((X+b)/k)*M$
Corrected Gamma	255	0	$((X+b)/k)*M$
Temperature	460	58	$((X+b)/k)*M$
Batt V	51	0	$((X+b)/k)*M$
AX, AY, AZ	2	1	$((X+b)/k)*M$
MX, MY, MZ	1.5	0.75	$((X+b)/k)*M$

Table 1: M-ary measurement conversions for zero-offset scaling.

6 Breaking Binary Data Into 2 and 3-bit Packets.

The converted data from section 4.2.2. must be broken into 2 and 3-bit packets. Given an n -bit binary value, it should be broken up into $n/3$ packets if n is divisible by 3 and into $(n/3 + 1)$ packet if the n is not divisible by 3, taking care to avoid a 1-bit packet. For n in the range 2 to 21, the data will be broken up into packets as shown in Table 2.

n	Packets and Packet size (in bits)
2	2
3	3
4	2+2
5	2+3
6	3+3
7	2+2+3
8	2+3+3
9	3+3+3
10	2+2+3+3
11	2+3+3+3
12	3+3+3+3
13	2+2+3+3+3
14	2+3+3+3+3
15	3+3+3+3+3
16	2+2+3+3+3+3
17	2+3+3+3+3+3
18	3+3+3+3+3+3
19	2+2+3+3+3+3+3
20	2+3+3+3+3+3+3
21	3+3+3+3+3+3+3

Table 2: Representation of 2 to 21-bits of binary data with 2 and 3-bit M-Ary packets.

7 M-ary Pulse Encoding

M-ary pulse encoding is a form of phase shift keying. The packets are encoded in 4 TU's (for 2-bit) or 8 TU's (for a 3-bit packet). The position of the rising edge of the pulse determines the packet value. For instance, in a 2-bit packet, the pulse would start at position 0 TU when the packet value is 3, at 1 TU for a value of 2, at 2 TU for a value of 1 and at 3 TU for a value of 0. Note that the pulse width itself is 2 TUs. Each packet is separated from the next packet by 4 TU's.

For a 3-bit packet, the pulse would start at position 0 TU when the packet value is 7, at 1 TU for a value of 6 and at 7 TU for a value of 0.

For example, a 12-bit value of 0xE75 (1110 0111 0101) would first be broken into 4 packets of 111, 001, 110 and 101. These packet values of 7,1,6 and 5 are coded over a space of 48 TU's (4 packets each 8 TU's long

with 4 TU's spacing after each packet). The pulses would occur at timeslot positions 0 (first packet, value of 7), 6 (second packet value of 1), 1 (third packet value of 6) and 2 (fourth packet value of 5) for each packet.

Over the entire length of 48 TU's, the 4 pulses would appear at 0th, 18th, 25th and 38th TU positions. See the MUD Pulse Telemetry M-ARY Encoding and Decoding white paper by Erdos Miller for additional details.

7.1 Example

This section describes an example of how a measurement value gets encoded as per the M-Ary Pulse rules. Let us try to encode an inclination value of 70° which has been specified in the sequence format as Inc : 12 (12-bit integer)

7.1.1 Scaling

Inclination value of 90° first gets scaled to $(70/180) * 2047 = 1593$

7.1.2 3-Bit Packet

A value of 1593 (0x639) is then broken up into four 3-bit packets.

0110 0011 1001 → 011 000 111 001.

The four packet values are now 3, 0, 7, and 1.

7.1.3 Pulse Encoding

After encoding, the pulse train generated will be the following:

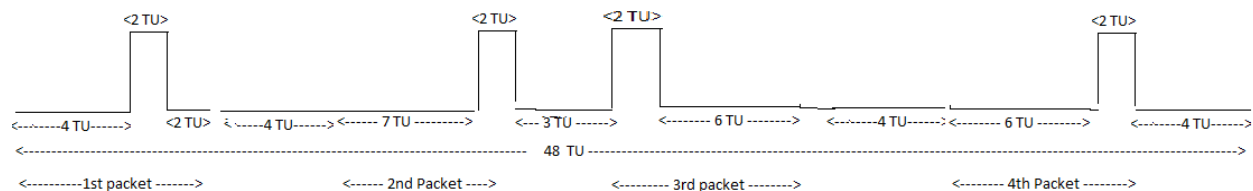


Figure 13: Example pulse train for the inclination encoding.