Appendix C

Example ADS-B Message Encoding

This page intentionally left blank.

C. Example ADS-B Message Encoding

C.1 Reed Solomon Encoding of Message Payload

The encoding is accomplished by means of a **systematic** RS 256-ary code with 8 bit code symbols (Bytes). The first parity symbol out of the FEC encoder is treated as the most significant symbol of the parity sequence. The parity check symbols are appended to the trailing end of the sequence of data symbols.

The ordering of the Bytes is from MSB to LSB, left to right, as they enter and exit the Reed Solomon encoder, as is the ordering of the bits within each Byte (MSB to LSB). When treated as polynomial coefficients this is equivalent to transmitting the high order coefficient(s) first. Since the systematic Reed Solomon Code is defined over the Galois Field GF(2^8), using the primitive polynomial [with binary coefficients, i.e., GF(2)] given as:

$$p(x) = x^8 + x^7 + x^2 + x + 1$$

all of the non-zero elements of the extension field GF(256) can be described as powers of a root of p(x); that is, for α such that $p(\alpha) = 0$, the nonzero elements of GF(256) are given as α^m, for m = 0, 1, 2, ..., 254 (where $\alpha^{255} \equiv \alpha^0 = 1$). For example,

$\alpha^8 \equiv \alpha^7 + \alpha^2 + \alpha + 1 =$ (1,0,0,0,0,1,1,1) as a binary 8-tuple (Byte).

To complete the description of the extension field elements, the "zero" element (additive identity) is denoted as $\underline{\mathbf{0}}$ = (0,0,0,0,0,0,0,0).

The generator polynomial for the Reed Solomon codes is given as:

$$G(x) = \prod_{i=120}^{P} (x - \alpha^i)$$

where P = 131 for RS (30,18) and P = 133 for RS (48,34) codes used for two ADS-B messages.

C.2 Reed Solomon Encoding of Basic Type 0 ADS-B Message Payload

The generator polynomial for the RS (30,18) Reed Solomon code is given as:

©2002, RTCA, Inc.

$$G(x) = x^{12} + \alpha^{76}x^{11} + \alpha^{66}x^{10} + \alpha^{157}x^{9} + \alpha^{28}x^{8} + \alpha^{92}x^{7} + \alpha^{220}x^{6} + \alpha^{88}x^{5}$$

$$+ \alpha^{20}x^{4} + \alpha^{145}x^{3} + \alpha^{50}x^{2} + \alpha^{56}x + \alpha^{231}$$

Table C-1 represents an example of a Basic ADS-B Message Payload with selected values for individual fields and the equivalent bit oriented representation. Each data field from Table C-1 is arranged in sequence as shown to depict the transmitted Basic ADS-B Message data sequence.

Transmitted Basic ADS-B Message Data Byte #

MSB 1	2	3	4	5	6	7
0000 0000	**1111 1010**	**1010 0001**	**0010 0011**	**0101 0101**	**0101 0101**	**0101 0101**
8	**9**	**10**	**11**	**12**	**13**	**14**
1100 0000	**0000 0000**	**0000 0000**	**0000 0011**	**0101 0100**	**0000 0110**	**0100 0100**
15	**16**	**17**	**18**			
0011 0010	**1100 0000**	**0010 1000**	**0000 0000**$_{LSB}$			

Table C-1: Example of Basic ADS-B Message Payloads

Data Field	Value	Bit-Oriented Equivalent
Payload Type Code	0	0 0000
Address Qualifier	0	000~~0~~
Aircraft Address	FAA123 (HEX)	1111 1010 1010 0001 0010 0011
Latitude (WGS-84)	60° N	010 1010 1010 1010 1010 1010
Longitude (WGS-84)	45º W	1110 0000 0000 0000 0000 0000
Altitude Type	0	0
Altitude	300 feet	0000 0011 0101
NIC	4	0100
Air/Ground State	0	00
[Reserved Bit]	0	0
North Velocity or Ground Speed	400 knots North	001 1001 0001
East Velocity or Heading	100 knots East	000 0110 0101
Vertical Velocity	+64 feet/minute (UP)	100 0000 0010
UTC coupled	1(YES)	1
[Reserved Bits]	0	000
[Reserved Byte]	0	0000 0000

Note: *The payload field definition of ADS-B Type 0 and Type 1 Messages in Table C-1 and Table C-2 respectively are current as of issue of this MOPS, version 1. These two messages are given as an example to describe how the field element α is defined within each symbol and how the data symbols sequence enter RS encoder and exit with FEC parity sequence at its trailing end.*

©2002, RTCA, Inc.

In generating the FEC, the ADS-B Message data bits are arranged into eight bit bytes assuming that the leftmost byte is the Most Significant Byte. The encoder accepts the 18 information symbols (Bytes) as:

$$\leftarrow [\underline{0}, \alpha^{165}, \alpha^{124}, \alpha^{232}, \alpha^{84}, \alpha^{84}, \alpha^{84}, \alpha^{105}, \underline{0}, \underline{0}, \alpha^{99}, \alpha^{214}, \alpha^{100}, \alpha^{143}, \alpha^{222}, \alpha^{105}, \alpha^{201}, \underline{0}]$$

and generates the 12 symbols parity sequence:

$$\leftarrow [\alpha^{60}, \alpha^{145}, \alpha^{41}, \alpha^{128}, \alpha^{120}, \alpha^{183}, \alpha^{138}, \alpha^{76}, \alpha^{220}, \alpha^{90}, \alpha^{175}, \alpha^{71}]$$

The parity sequence is then appended to the (right) end of the information sequence to complete the 30 symbols Codeword for transmission (left symbol first).

MSB **ADS-B Basic Message Payload Bits + FEC Parity Bits**

0000 0000 1111 1010 1010 0001 0010 0011 0101 0101 0101 0101 0101 0101 1100 0000 0000 0000
0000 0000 0000 0011 0101 0100 0000 0110 0100 0100 0011 0010 1100 0000 0010 1000 0000 0000
1111 1110 1001 0111 1100 0100 0011 0100 1110 0001 1111 1111 0101 0011 0110 0101 1100 1111
1000 1111 1010 1111 1110 0100$_{\mathbf{LSB}}$

C.3 Reed Solomon Encoding of Long Type 1 ADS-B Message Payload

The generator polynomial for the RS (48,34) Reed Solomon code is given as:

$$G(x) = x^{14} + \alpha^{82}x^{13} + \alpha^{49}x^{12} + \alpha^{21}x^{11} + \alpha^{70}x^{10} + \alpha^{26}x^{9} + \alpha^{140}x^{8} + \alpha^{135}x^{7} + \alpha^{138}\alpha^{6}$$

$$+ \alpha^{22}x^{5} + \alpha^{64}x^{4} + \alpha^{13}x^{3} + \alpha^{39}x^{2} + \alpha^{70}x + \alpha^{241}$$

Table C-2 represents an example of Long Type 1 ADS-B Message payload (Basic ADS-B State Vector plus MS [Mode Status] elements, and AUX State Vector report elements fields) and the equivalent bit oriented representation.

©2002, RTCA, Inc.

Table C-2: Example of Long Type 1 ADS-B Message Payloads

Data Field	Value	Bit-Oriented Equivalent
Payload Type Code	~~0~~ 1	0 0001
Address Qualifier	0	~~0~~000
Aircraft Address	FAA123 (HEX)	1111 1010 1010 0001 0010 0011
Latitude (WGS-84)	60° N	010 1010 1010 1010 1010 1010
Longitude (WGS-84)	45º W	1110 0000 0000 0000 0000 0000
Altitude Type	0	0
Altitude	+300 feet	0000 0011 0101
NIC	4	0100
Air/Ground State	0	00
[Reserved Bit]	0	0
North Velocity or Ground Speed	400 knots North	001 1001 0001
East Velocity or Heading	100 knots East	000 0110 0101
Vertical Velocity	+64 feet/minute(UP)	100 0000 0010
UTC coupled	1(YES)	1
[Reserved Bits]	0	000
~~Participant~~ Emitter Category Code and ~~Flight ID (~~Call Sign~~)~~ Characters#1 and #2	2 (Small) and "AB"	0000 1110 0001 1011
~~Flight ID (~~Call Sign~~)~~ Characters#3, #4, and #5	"CD1"	0100 1101 0000 1001
~~Flight ID (~~Call Sign~~)~~ Characters#6, #7, and #8	"234"	0000 1100 1111 1100
Emergency	0	000
MOPS Version	1	001
SIL	0	00
TMSO(6 LSBs of 12-bit MSO #)	1250 (only 6 LSB is transmitted)	10 0010
[Reserved Bits]	0	00
NAC_P	7	0111
NAC_V	2	010
NIC_{BARO}	0	0
Capability Class (CC) Codes	0	00
Operational Mode (OM) Codes	0	000
True Magnitude	0	0
[Reserved Bits]	0	00 0000 0000 0000 0000
Secondary Altitude	+300 feet	0000 0011 0101
[Reserved Bits]	0	0000 0000 0000 0000 0000 0000 0000

Note: *The payload field definition of ADS-B Type 0 and Type 1 Messages in Table C-1 and Table C-2 respectively are current as of issue of this MOPS, version 1. These two messages are given as an example to describe how the field element α is defined within each symbol and how the data symbols sequence enter RS encoder and exit with FEC parity sequence at its trailing end.*

Each data field from Table C-2 is arranged in sequence as shown to depict the transmitted Long Type 1 ADS-B Message data sequence.

©2002, RTCA, Inc.

Transmitted Long Type 1 ADS-B Message Data Bytes #

MSB----1	2	3	4	5	6	7
0001 0000	1111 1010	1010 0001	0010 0011	0101 0101	0101 0101	0101 0101
8	9	10	11	12	13	14
1100 0000	0000 0000	0000 0000	0000 0011	0101 0100	0000 0110	0100 0100
15	16	17	18	19	20	21
0011 0010	1100 0000	0010 1000	0000 1110	0001 1011	0100 1101	0000 1001
22	23	24	25	26	27	28
0000 1100	1111 1100	0000 0100	1000 1000	0111 0100	0000 0000	0000 0000
29	30	31	32	33	34	
0000 0000	0000 0011	0101 0000	0000 0000	0000 0000	0000 0000$_{LSB}$	

In generating the FEC, the ADS-B Message data bits are arranged into eight bit bytes assuming that the leftmost byte is the Most Significant Byte. The encoder accepts the 34 information symbols (Bytes) as:

$$\leftarrow [\alpha^{4},\alpha^{165},\alpha^{124},\alpha^{232},\alpha^{84},\alpha^{84},\alpha^{84},\alpha^{105},\underline{0},\underline{0},\alpha^{99},\alpha^{214},\alpha^{100},\alpha^{143},\alpha^{222},\alpha^{105},\alpha^{201},\alpha^{107},$$
$$\alpha^{49},\alpha^{117},\alpha^{205},\alpha^{101},\alpha^{58},\alpha^{2},\alpha^{144},\alpha^{34},\underline{0},\underline{0},\underline{0},\alpha^{99},\alpha^{202},\underline{0},\underline{0},\underline{0}]$$

and generates the 14 symbol parity sequence:

$$\leftarrow [\alpha^{195},\alpha^{208},\alpha^{63},\alpha^{181},\alpha^{92},\alpha^{9},\alpha^{124},\alpha^{222},\alpha^{236},\alpha^{6},\alpha^{92},\alpha^{235},\alpha^{3},\alpha^{153}]$$

The parity sequence is then appended to the (right) end of the information sequence to complete the 48 symbols Codeword for transmission (left symbol first).

MSB **Long Type 1 ADS-B Message Payload Bits + FEC Parity Bits**

~~0001~~ 0000 ~~0000~~ 1000 1111 1010 1010 0001 0010 0011 0101 0101 0101 0101 0101 0101 1100 0000
0000 0000 0000 0000 0000 0011 0101 0100 0000 0110 0100 0100 0011 0010 1100 0000 0010 1000
0000 1110 0001 1011 0100 1101 0000 1001 0000 1100 1111 1100 0000 0100 1000 1000 0111 0100
0000 0000 0000 0000 0000 0000 0000 0011 0101 0000 0000 0000 0000 0000 0000 0000 ~~1001~~ 0000
~~0010~~ 1110 ~~0100~~ 0001 ~~1000~~ 1001 ~~0110~~ 1000 ~~1011~~ 0101 ~~0101~~ 1111 ~~1110~~ 1111 ~~1011~~ 1001 ~~0101~~ 1100
~~1000~~ 0110 ~~1001~~ 1011 ~~1010~~ 0011 ~~0001~~ 0011 ~~0011~~ 0000 ~~0010~~ 0110 ~~1011~~ 0110 ~~1001~~ 1100 ~~0100~~ 1010
~~0000~~ 1001 ~~1011~~ 1111 ~~0101~~ 0100 ~~1001~~ 0011 ~~1111~~ 1101 ~~0000~~ 0001 ~~1000~~ 0010 ~~1011~~ 1110 ~~0010~~ 1000

LSB

©2002, RTCA, Inc.

Appendix C References

References for forward Error Coding and the Galois Field are listed below:

C-1. Peterson, W.W., and E.J. Weldon, Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, MA, 1972.

C-2. Michelson, A. M., and A. H. Levesque, Error-Control Techniques for Digital Communication, John Wiley & Sons, New York, NY 1985.

©2002, RTCA, Inc.