WX-250

Weather Radio Digital Interface Document

Computer Automation Technology, Inc.
Fort Lauderdale, Florida 33309
(954) 978-6171
www.catauto.com

6/22/2011 Rev A

Contents

Revision History 2

Introduction 3

1.1 Scope 3

1.2 Definitions 3

1.3 WX-250 User's Manual 4

1.4 Hardware specification 4

1.5 Interconnecting wire 4

1.6 Conductor definition 4

1.7 Electrical connections and shield requirements 4

1.8 Signal state definitions 4

1.9 Talker drive circuits 5

2.0 Listener receive circuits 5

2.1 Data transmission 5

2.2 Characters 5

2.3 Reserved characters 5

2.4 Valid characters 6

2.5 Fields 6

2.6 Identification field 6

2.7 Data fields 6

2.8 Variable length fields 6

2.9 Null fields 6

3.0 Checksum fields 7

3.1 Sentences 8

3.2 Description of sentences 8

3.3 Sentence structure description 9

3.4 Identification field definitions for Weather Receiver 10

4.0 $WXALT - Active alert just received (Free text field) 13

4.1 WXCFG - Receiver configuration settings 14

4.2 WXEVT - Receiver events selected 16
4.3 WXFIP - Receiver FIPS 17
4.4 WXFRQ - Receiver frequency setting 18
4.5 WXSWF - Receiver identification and software version 18
4.6 WXSIG - Receiver signal strength 19
4.7 WXACT - Receiver alert activity timer’s settings 20
4.8 WXTXT - Text message transmission (Free text field) 21

Revision History

Date	Rev By	Description
6/28/2011	acp	Document released

Introduction

This document was developed with input from manufacturers, private and government organizations and WX-250 users to document the RS-232 extended digital output. The information contained in this document is intended to meet the needs of users at the time of publication, but users must recognize that as applications and technology change, interface standards must change as well. Users of this document are advised to immediately inform Computer Automation Technology of any perceived inadequacies in this document.

1.1 Scope

This part contains the requirements for data communication between the WX-250 weather receiver and PC computer equipment when interconnected via an appropriate system. This document is intended to support one-way serial data transmission from a single talker (WX-250) to one or more listeners. This data is in printable ASCII encapsulated string format and may include information such as weather alerts received, signal strength, etc. Typical messages may be from about 10 to a maximum of 79 characters in length and generally require transmission no more rapidly than one message per second. The electrical definitions in this document are not intended to accommodate high-bandwidth applications or intensive database or file transfer applications. Since there is no provision for guaranteed delivery of messages and only limited error checking capability, this document should be used with caution in all safety applications.

1.2 Definitions

For the purposes of this document, the following definitions apply.

talker
A device which sends data to another device or devices.

listener
A device which receives data from another device

1.3 WX-250 User’s Manual

The WX-250 user’s manual contains the wiring diagrams and setup procedures necessary to enable the message strings sent by the WX-250 receiver including an electrical description or schematic of the listener/talker input/output circuits citing actual components and devices used, including connector type and part number.

1.4 Hardware specification

Multiple listeners may be connected in parallel over an interconnecting wire. The number of listeners depends on the output capability and input drive requirements of individual devices. Contact Computer Automation Technology for more information.

1.5 Interconnecting wire

Interconnection between devices may be by means of a two-conductor, shielded, twisted-pair wire.

1.6 Conductor definition

The conductors referred to in this document are the signal lines identified as signal A and B, and shield.

1.7 Electrical connections and shield requirements

All signal line “A” connections are connected in parallel with all devices “A” connections and all signal line “B” connections are connected in parallel with all devices “B” connections. The shields of all listener cables should be connected to the talker chassis only and should not be connected at each listener.

1.8 Signal state definitions

The idle, marking, logical 1, OFF or stop bit states are defined by a negative voltage on line A with respect to line B.

The active, spacing, logical 0, ON or start bit states are defined by a positive voltage on line A with respect to line B.

It should be noted that the above A with respect to B levels are inverted from the voltage input/output requirements of standard UARTs and that many line drivers and receivers provide a logic inversion.

1.9 Talker drive circuits

No provision is made for more than a single talker to be connected to the signal bus.

2.0 Listener receive circuits

Multiple listeners may be connected to a single talker. The listener receiver may need to consist of an opto-isolator and require protective circuits to limit current, reverse bias and power dissipation at the opto-diode.

2.1 Data transmission

Data is transmitted in serial asynchronous format. The first bit is a start bit and is followed by data bits, least-significant-bit first.

The following parameters are used:

– Baud rate = 9600
– Data bits = 8 (D7 = 0), parity none
– Stop bits = 1

2.2 Characters

All transmitted data shall be interpreted as ASCII characters. The most significant bit of the eight-bit character shall always be transmitted as zero (D7 = 0).

2.3 Reserved characters

The reserved character set consists of those ASCII characters shown in Table 1 below. These characters are used for specific formatting purposes, such as sentence and field delimiting, and except for code delimiting, are not used in data fields.

2.4 Valid characters

The valid character set consists of all printable ASCII characters (HEX 20 to HEX 7E) except those defined as reserved characters. See Table 2 below.

2.5 Fields

A field consists of a string of valid characters, or no characters (null field), located between two appropriate delimiter characters.

2.6 Identification field

An identification field is the first field in a sentence and always begins with the "$" delimiter; it serves to define the sentence start. Characters within the identification field are always upper case letters. The identification field shall not be a null field.

2.7 Data fields

Data fields in a approved sentence will follow with a "," delimiter and contain valid characters in accordance with this document. Data will only be located within a sentence that uses field delimiters (","). Therefore, it is essential for the listener to locate fields by counting delimiters rather than counting the total number of characters received from the start of the sentence.

2.8 Variable length fields

Although some data fields are defined to have fixed length, many are of variable length in order to allow devices to convey information and to provide data with more or less precision, according to the capability or requirements of a particular message. Variable length fields may be alphanumeric or numeric fields. Variable numeric fields may contain a decimal point and may contain leading or trailing zeros.

2.9 Null fields

A null field is a field of length zero, i.e. no characters are transmitted in the field. Null fields shall be used when the value is unreliable or not available. For example, if heading information were not available, sending data of "000" is misleading because a user cannot distinguish between "000" meaning no data and a legitimate value of "000". However, a null field, with no characters at all, clearly indicates that no data is being transmitted.

Null fields with their delimiters can have the following appearance depending on where they are located in the sentence:

",," ",*"

The ASCII NULL character (HEX 00) is not used as the null field.

3.0 Checksum fields

A checksum field is transmitted in all sentences. The checksum field is the last field in a sentence and follows the checksum delimiter character "*". The checksum is the eight-bit exclusive OR (no start or stop bits) of all characters in the sentence, including "," delimiters, between but not including the "$" and the "*" delimiters.

The hexadecimal value of the most significant and least significant four bits of the result is converted to two ASCII characters (0-9, A-F) for transmission. The most significant character is transmitted first.

Examples of the checksum field are:

$WXFRQ_0_162.400*55	Checksum = 55 = 01010101
$WXFIP_0_012099,012011,012086,*41	Checksum = 41 = 01000001
$WXEVT_0_SVA,SVR,TOA,TOR,RWT,,,*05	Checksum = 05 = 00000101

3.1 Sentences

This section describes the general structure of sentences. Details of specific sentence formats are found in later pages. Some sentences may specify restrictions beyond the general limitations given in this part of this document. Such restrictions may include defining some fields as fixed length, numeric or text only, required to be non-null, transmitted with a certain frequency, etc.

The maximum number of characters in a sentence shall be up to 82, consisting of a maximum of 79 characters between the starting delimiter "$" and the terminating delimiter <CR><LF>. The minimum number of fields in a sentence is one (1). The first field shall be an address field containing the identity of the talker and the sentence formatter which specifies the number of data fields in the sentence, the type of data they contain and the order in which the data fields are transmitted. The remaining portion of the sentence may contain zero or multiple data fields. The maximum number of fields allowed in a single sentence is limited only by the maximum sentence length of 82 characters. Null fields may be present in the sentence and shall always be used if data for that field is unavailable. All sentences begin with the sentence-starting delimiter character "$" and end with the sentence-terminating delimiter <CR><LF>.

3.2 Description of sentences

Data sentences are those designed for general use and detailed in this document.

All approved sentence contains, in the order shown, the following elements:

ASCII	HEX	Description
$	24	Start of sentence
<identification field>		Talker sentence identifier
[“_” <sequence number> “_”]		Sequence number 0..9
[<data field>]		First data fields
["," <data field>]		Data field or fields
["*" <checksum field>]		Checksum field
<CR><LF>	0D 0A	End of sentence

3.3 Sentence structure description

The following provides a summary explanation of the approved sentence structure:

Example: $WXxxx_n_ c---c*hh<CR><LF>

ASCII	HEX	Description
$	24	Start of sentence: starting delimiter
WXxxx		Identification field: alphanumeric characters identifying type of talker, and sentence formatter. The first two characters identify the talker. The last three are the sentence formatter mnemonic code identifying the data type and the string format of the successive fields. Mnemonics will be used as far as possible to facilitate read-outs by users.
“_”	5F	Header delimiter
n		Sequence number starting with 0
“_”	5F	Header delimiter
f1,f2,f3		Data sentence block: follows header and sequence field and is a series of data fields containing all of the data to be transmitted. Data field sequence is fixed and identified by the third and subsequent characters of the address field (the sentence formatter). Inside data fields contain comma separators (“,”).
“*”	2A	Checksum delimiter: follows last data field of the sentence. It indicates that the following two alpha-numeric characters show the HEX value of the checksum.
hh		Checksum field: the absolute value calculated by exclusive OR'ing the eight data bits (no start bits or stop bits) of each character in the sentence between, but excluding, "$" and "*". The hexadecimal value of the most significant and least significant four bits of the result are converted to two ASCII characters (0-9, A-F) for transmission. The most significant character is transmitted first. The checksum field is required in all cases.
<CR><LF>	0D 0A	End of sentence: sentence terminating delimiter.

3.4 Identification field definitions for Weather Receiver

Talker description	Identifier
Active alert just received	$WXALT-
Receiver configuration settings	$WXCFG_
Receiver events selected	$WXEVT_
Receiver FIPS	$WXFIP_
Receiver frequency setting	$WXFRQ_
Receiver signal strength	$WXSIG_
Receiver software version	$WXSWF_
Alert timers settings	$WXACT_
Text message transmission	$WXTXT_

Table 1 – Reserved characters

ASCII	HEX	DEC	Description
<CR>	0D	13	Carriage return
<LF>	0A	10	Line feed – End of sentence delimiter
$	24	36	Start of sentence delimiter
*	2A	42	Checksum field delimiter
,	2C	44	Field delimiter
_	21	33	Header separator

Table 2 – Valid characters

ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC
Space	20	32	@	40	64	`	60	96
Reserved	21	33	A	41	65	a	61	97
“	22	34	B	42	66	b	62	98
#	23	35	C	43	67	c	63	99
Reserved	24	36	D	44	68	d	64	100
%	25	37	E	45	69	e	65	101
&	26	38	F	46	70	f	66	102
‘	27	39	G	47	71	g	67	103
(	28	40	H	48	72	h	68	104
)	29	41	I	49	73	i	69	105
Reserved	2A	42	J	4A	74	j	6A	106
+	2B	43	K	4B	75	k	6B	107
Reserved	2C	44	L	4C	76	l	6C	108
-	2D	45	M	4D	77	m	6D	109
.	2E	46	N	4E	78	n	6E	110
/	2F	47	O	4F	79	o	6F	111
0	30	48	P	50	80	p	70	112
1	31	49	Q	51	81	q	71	113
2	32	50	R	52	82	r	72	114
3	33	51	S	53	83	s	73	115
4	34	52	T	54	84	t	74	116
5	35	53	U	55	85	u	75	117
6	36	54	V	56	86	v	76	118
7	37	55	W	57	87	w	77	119
8	38	56	X	58	88	x	78	120

9	39	57	Y	59	89	y	79	121
:	3A	58	Z	5A	90	z	7A	122
;	3B	59	[	5B	91	{	7B	123
<	3C	60	Reserved	5C	92	|	7C	124
=	3D	61	]	5D	93	}	7D	125
>	3E	62	Reserved	5E	94	Reserved	7E	126
?	3F	63	-	5F	95	Reserved	7F	127

Explanation of Weather Radio sentences

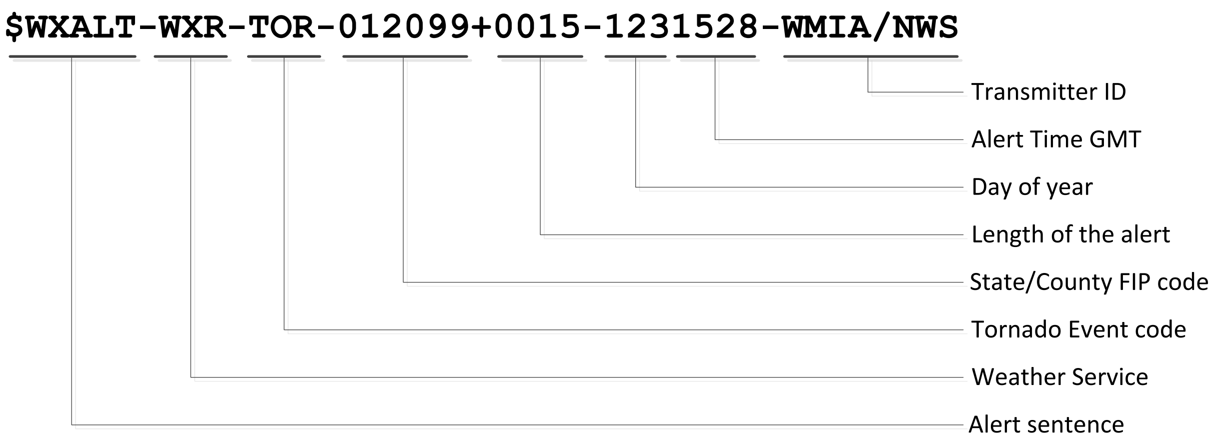
4.0 $WXALT - Active alert just received (Free text field)

This sentence identifier, along with some simple text messages are sent upon a received alert from the NOAA transmitting facility and has been decoded as a valid alert.

Examples of a tornado alert (TOR) as transmitted by the West Palm Beach, Florida NOAA transmitter.

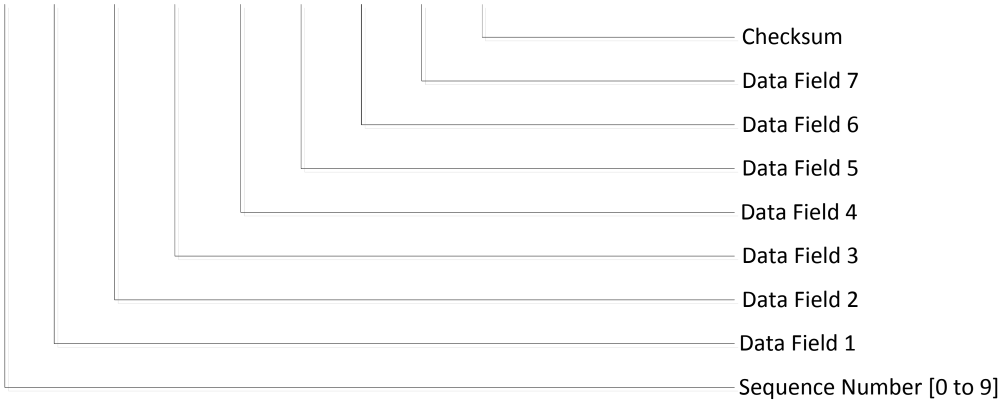
```
$WXTXT,SAME interrupt detected.*2E
$WXTXT,Preamble detected.*53
$WXTXT,StoredOrgEvent OK*72
$WXTXT,Match FIPS OK*20
$WXTXT,Match Event OK*60
$WXALT-WXR-TOR-012099+0015-1231545-KMIA-NWS
```

$WXTXT	SAME interrupt detected	This message is sent whenever the receiver DSP has received a SAME digital message.
$WXTXT	Preamble detected.	The preamble or start of message has been received and confirm as valid.
$WXTXT	StoredOrgEvent OK	The organization ID of WXR has been confirmed as valid.
$WXTXT	Match FIPS OK	The FIP codes stored within the weather radio has matched with one or more FIP codes sent by the NOAA transmitter.
$WXTXT	Match Event OK	The Event codes stored within the weather radio has matched the Event code sent by the NOAA transmitter.



4.1 WXCFG - Receiver configuration settings

```
$WXCFG_x_f1,f2,f3,f4,f5,f6,f7*hh<CR><LF>
```



Sequence Number = 0		
Data Field		Description
f1	Yes or No	Ignore Termination Packet.
f2	Yes or No	Accept All Event Codes.
f3	Yes or No	Accept All FIP Codes.
f4	Yes or No	Limit Monthly/Weekly Test to 15 Minutes Max.
f5	Yes or No	Permit Voice Interval Announcements.
f6	Yes or No	Limit Announcements to Warnings Only.
f7	Yes or No	Limit Announcements to Watch/Warnings Only.

Sequence Number = 1		
Data Field		Description
f1	Number	Line Timer Value in Minutes
f2	Number	Relay Activation Delay in Seconds
f3	Number	Pre-Voice Delay in Seconds
f4	Number	Voice Interval Timer in Minutes

f1 = 1, 2, 3, 4, 5, 6 Minutes.

f2 = 0, 2, 4, 6, 8 Seconds.

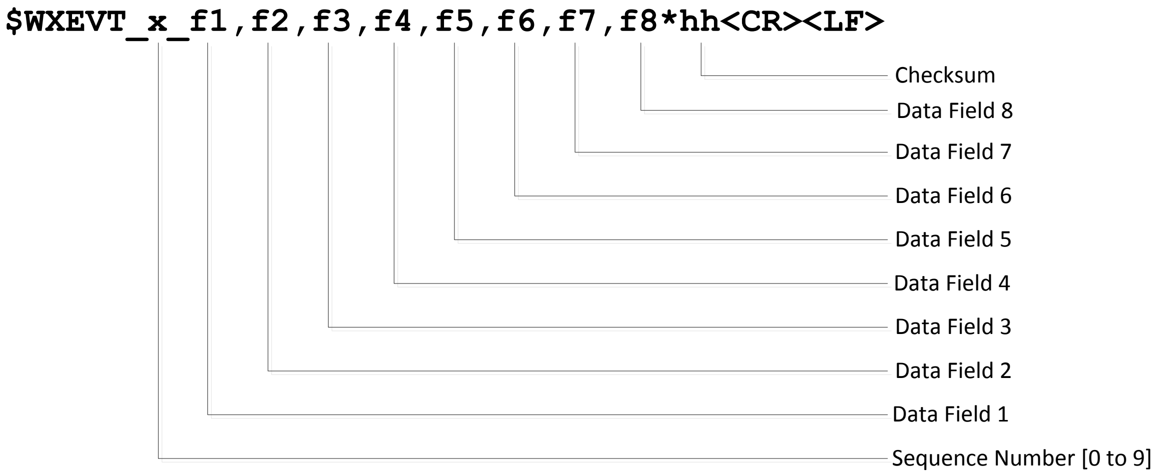
f3 = 0.0, 0.5, 1.0, 1.5, 2.0 Seconds.

f4 = 5, 10, 15, 20, 25, 30 Minutes.

Sequence Number = 2		
Data Field		Description
f1	Yes or No	Append Station Identification after an Alert.
f2	Yes or No	Append “This-is” before ID.
f3	Yes or No	Append “Repeater” after ID.
f4	Yes or No	Append “Transmitter” after ID.
f5	Letters	Call letters (ID) of Transmitter.

Note: Call letters field (f5) may be empty.

4.2 WXEVT - Receiver events selected



```
$WXEVT_x_f1,f2,f3,f4,f5,f6,f7,f8*hh<CR><LF>
```

Checksum

Data Field 8

Data Field 7

Data Field 6

Data Field 5

Data Field 4

Data Field 3

Data Field 2

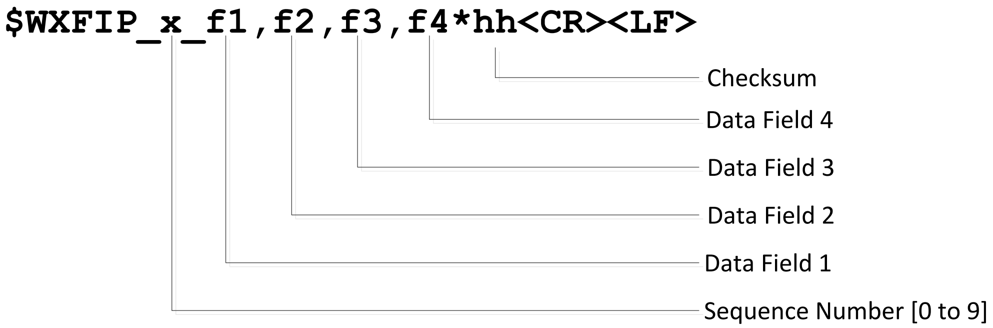
Data Field 1

Sequence Number [0 to 9]

Sequence Number = 0, 1, 2		
Data Field	Event Code	Description
f1	abc	Selected NOAA Event
f2	abc	Selected NOAA Event
f3	abc	Selected NOAA Event
f4	abc	Selected NOAA Event
f5	abc	Selected NOAA Event
f6	abc	Selected NOAA Event
f7	abc	Selected NOAA Event

abc = Three letter event code. Example “RWT” for Require Weekly Test.

4.3 WXFIP - Receiver FIPS

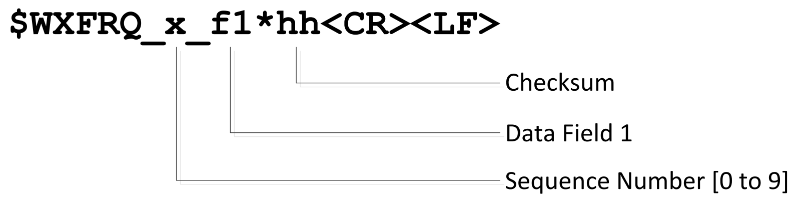


```
$WXFIP_x_f1,f2,f3,f4*hh<CR><LF>
```

Sequence Number = 0, 1, 2, 3, 4, 5, 6, 7		
Data Field	Event Code	Description
f1	abcdef	Selected NOAA FIP location code
f2	abcdef	Selected NOAA FIP location code
f3	abcdef	Selected NOAA FIP location code
f4	abcdef	Selected NOAA FIP location code

abcdef = Six numeric FIP code. Example “012099” = West Palm Beach, Florida.

4.4 WXFRQ - Receiver frequency setting



```
$WXFRQ_x_f1*hh<CR><LF>
```

Checksum

Data Field 1

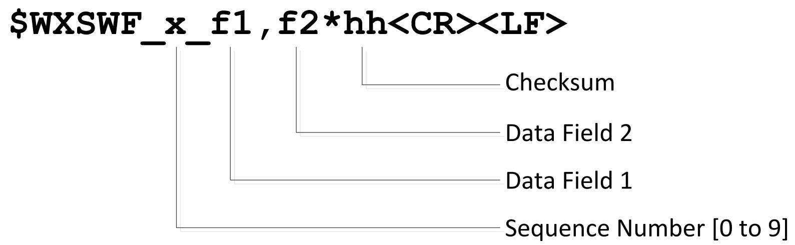
Sequence Number [0 to 9]

Data field f1 contains the frequency the radio is tuned to.

Example

f1 = 162.400

4.5 WXSWF - Receiver identification and software version



```
$WXSWF_x_f1,f2*hh<CR><LF>
```

Checksum

Data Field 2

Data Field 1

Sequence Number [0 to 9]

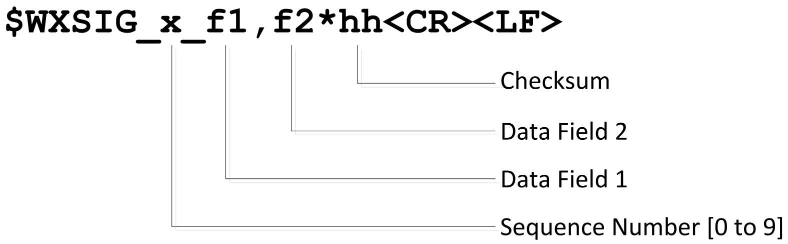
Sequence Number = 0		
Data Field		Description
f1		Radio Identification
f2		Radio Software version

Example

f1 = WX-250

f2 = 1.04

4.6 WXSIG - Receiver signal strength



```
$WXSIG_x_f1,f2*hh<CR><LF>
```

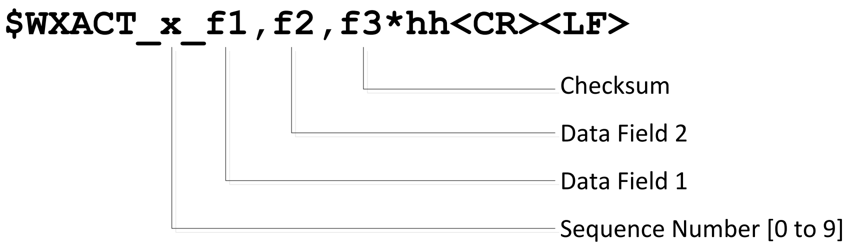
Sequence Number = 0		
Data Field		Description
f1		Radio RSSI in dBuV
f2		Radio S/N ratio in dB

Example

f1 = 45 dBuV

f2 = 53 dB

4.7 WXACT - Receiver alert activity timer's settings



```
$WXACT_x_f1,f2,f3*hh<CR><LF>
```

Sequence Number = 0		
Data Field		Description
f1	Sec Hex Value	During an Event, Line Timer value
f2	Sec Hex Value	During an Event, Voice Activity value
f3	Min Hex Value	During an Event, Event Timeout value

Example

f1 = 00A2 = 162 seconds remain before a Line Timeout timer expires.

f2 = 0113 = 275 seconds before next Voice Announcement.

f3 = 0000000F = 15 Minutes before active event expires.

4.8 WXTXT - Text message transmission (Free text field)

Text Identification field during certain radio activity.

$WXTXT_SAME interrupt detected.*5D
$WXTXT_Preamble detected.*20
$WXTXT_StoredOrgEvent OK*01
$WXTXT_Match FIPS OK*53
$WXTXT_Match Event OK*13

All of the text message are in free text ASCII characters and are self-explanatory. Below is a list of Text message contain programmed within the Weather Radio at the time of this document. The header and checksum have been removed to provide improved readability.

Alert received but system is disabled.
SAME interrupt detected.
Preamble detected.
End message packet detected.
StoredOrgEvent OK
Match FIPS OK
Match Event OK
Force monitor entered.
Force monitor exit.
External force monitor entered.
External force monitor exit.
Voice FIP codes entered.
Voice FIP codes exit.
Test alert entered.
Test alert exit.
Voice alert entered.
Voice alert exit.
Line timer expired.
Alert cancelled by user.
Alert timer has expired.
Voice identification entered.
Voice identification exit.
Found previous alert.